Universidad de Granada

Departamento de Ciencias de la Computación
e Inteligencia Artificial

Closed-Domain Natural Language Approaches: Methods and Applications

Tesis Doctoral

Alejandro Moreo Fernández

Granada, Julio de 2013

Editor: Editorial de la Universidad de Granada
Autor: Alejandro Moreo Fernández
D.L.: GR 525-2014
ISBN: 978-84-9028-834-4

Universidad de Granada

Closed-Domain Natural Language Approaches: Methods and Applications

MEMORIA QUE PRESENTA

Alejandro Moreo Fernández

PARA OPTAR AL GRADO DE DOCTOR EN INFORMÁTICA

Julio de 2013

DIRECTORES

Dr. Juan Luis Castro Peña
Dr. Jose Manuel Zurita López

Departamento de Ciencias de la Computación
e Inteligencia Artificial

La memoria titulada “*Closed-Domain Natural Language Approaches: Methods and Applications*”, que presenta D. Alejandro David Moreo Fernández para optar al grado de doctor, ha sido realizada dentro del Máster Oficial de Doctorado “*Soft Computing y Sistemas Inteligentes*” del programa de Doctorado Oficial “*Ciencias de la Computación y Tecnología Informática (P36.56.1)*” del Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada bajo la dirección de los doctores D. Juan Luis Castro Peña y D. Jose Manuel Zurita López.

El doctorando Alejandro David Moreo Fernández y los directores de la tesis Juan Luis Castro Peña y Jose Manuel Zurita López garantizamos, al firmar esta tesis doctoral, que el trabajo ha sido realizado por el doctorando bajo la dirección de los directores de la tesis y hasta donde nuestro conocimiento alcanza, en la realización del trabajo, se han respetado los derechos de otros autores a ser citados, cuando se han utilizado sus resultados o publicaciones.

El doctorando Alejandro David Moreo Fernández ha realizado una estancia en el *Istituto di Scienza e Tecnologie dell'Informazione “A. Faedo”*, un instituto de investigación de Pisa perteneciente al *National Research Council* de Italia. La estancia, de tres meses de duración, fue supervisada por los investigadores Fabrizio Sebastiani y Andrea Esuli en el campo de *Selección de Características en Classificación de Textos.* Aunque este proyecto sigue aún en desarrollo, se han reflejado algunos experimentos previos en este texto.

De acuerdo a la normativa vigente para el reconocimiento de *Mención Internacional*, tanto el planteamiento de la investigación como las conclusiones se presentan en dos idiomas, español e inglés en nuestro caso. El desarrollo de la investigación se presenta únicamente en inglés, el idioma más extendido en la literatura científica.

Granada, Julio de 2013

El Doctorando

Fdo: Alejandro David Moreo Fernández

Los Directores

Fdo: Juan Luis Castro Peña

Fdo: Jose Manuel Zurita López

Esta Tesis doctoral ha sido financiada por medio de la beca FPU (Formación de Profesorado Universitario) referencia AP2008-02464 del Ministerio de Educación y Ciencia.

Agradecimientos

Quiero dedicar esta memoria en primer lugar a mis padres por haberme animado para que llegara hasta aquí y haberme enseñado el valor del esfuerzo y la dedicación, y a mis directores de Tesis Juan Luis y Jose Manuel porque sin ellos este trabajo no hubiera sido posible y por haberme enseñado a valorar la investigación durante estos años. Espero seguir colaborando y aprendiendo con ellos. También quiero agradecer a mis hermanos Pedro y Migue por haberme apoyado siempre y particularmente durante la carrera, en la que hemos compartido muchos punteros a *null*, y a Angy por haber aguantado estoicamente todo el proceso de locura que ha acarreado esta Tesis, y haberme ayudado tanto en la preparación de ponencias y revisión de artículos. He aprendido mucho de todos vosotros.

Gracias a mis amigos Vene, Edu, Víctor, Sabi, Jesús, Michela, Christoph y Carlos con quienes he ahogado algunas penas y muchas alegrías. A Paco y Viqui, con quienes comparto tantas aficiones y tanto me han apoyado y ayudado con la burocracia. Gracias también a mis amigos y compañeros de trabajo y de carrera Manu, Álvaro y Leo, hemos compartido muchos ratos de estudio, bebidas energéticas y comida basura.

También quiero citar a mis compañeros de trabajo más allegados, María, con la que he compartido un sinfín de situaciones patéticas, y Víctor, que tanto me ayudó en mis primeros tropiezos. También me gustaría agradecer a la gente de Virtual Solutions, y en especial Edu y Javi, con quienes he colaborado codo con codo, por su trato tan cercano. Gracias también a Fran, mi compañero de despacho, que me ha ayudado a dar forma a este trabajo. Literalmente. Me prestó su plantilla de LaTeX. Y a Chopin y AC/DC, que han compuesto la banda sonora de estos años de trabajo.

No puedo dejar atrás a mis compañeros en Italia, que tanto me ayudaron durante mi periodo de estancia, y en especial cuando *il ragazzo spagnolo non capiva niente*, Giacomo, Cristina, Tiziano y Diego, que se esforzaron en darme una calurosa acogida, Fabrizio y Andrea de quienes tanto he aprendido durante estos meses, y Fabrizio y Flavia, que me hicieron sentir tan cómodo durante mi estancia. Espero no perder el contacto con vosotros.

GRACIAS A TODOS

Resumen

Este proyecto de Tesis tiene por objetivo aportar soluciones novedosas en forma de aplicaciones informáticas a problemas abiertos con margen de mejora y de actual demanda por la sociedad de la información. En este trabajo proponemos un análisis de las tecnologías de dominio cerrado basadas en lenguaje natural en función de los diferentes niveles de información disponibles: la estructura del conocimiento, y el modelado conceptual o metaconocimiento. Nuestra hipótesis de partida es que tanto el rendimiento de la aplicación como la reducción de los costes asociados podrían beneficiarse de técnicas específicas en base a estos niveles de conocimiento. Concretamente, se abordan los siguientes temas: (i) qué características confieren relevancia a un término frente a un dominio, (ii) cómo delimitar el contexto de discurso para mejorar el análisis del lenguaje, (iii) cómo diseñar sistemas colaborativos escalables y refinar su contenido y (iv) cómo simular un proceso de aprendizaje basado en razonamiento por analogía del lenguaje. Hemos reflejado su estudio mediante los siguientes problemas: (i) Selección de Características en Clasificación de Textos, (ii) Análisis de Opiniones en comentarios a noticias, (iii) recuperación de FAQs y sistemas basados en plantillas y (iv) Interfaces en Lenguaje Natural. Este trabajo es fruto de proceso de ingeniería de carácter científico. Desde el punto de vista científico, hemos logrado interesantes resultados y contribuciones al estado del arte avaladas por diversas publicaciones científicas. Desde el punto de vista técnico, hemos desarrollado diferentes aplicaciones informáticas de interés comercial que están siendo usadas en la actualidad.

Summary

The main goal of this Thesis is to offer meaningful solutions through concrete applications to real open-problems coming from the information society. We propose a categorization of closed-domain natural language approaches in basis of the different knowledge-levels available, including the structure of the data and the meta-knowledge. Our hypothesis is that both the system performance and the reduction in the associated costs could be benefited from specific techniques being aware of these knowledge-levels. More specifically, following issues are here tackled: (i) which are the main characteristics that make a term become relevant to a given domain, (ii) how the discourse-context could be delimited in advance in order to improve the analysis of texts, (iii) how to design truly collaborative systems through scalable methods that could in addition be self-aware of its own knowledge weaknesses, and (iv) how the language capabilities of the system could be improved by simulating a learning process through reasoning by analogy. These studies have been respectively approached from the following problems: (i) Feature Selection for Text Classification, (ii) Feature-based Sentiment Analysis on News items, (iii) Template-based FAQ retrieval approaches, and (iv) Natural Language Interfaces. This work is the result of an engineering process of scientific nature. From the scientific point of view, several valuable results and contributions to the state-of-the-art have been published in different scientific media. From a technical point of view, different computer applications of commercial interest have been developed.

Contents

List of Figures

Chapter I

Planteamiento de la Investigación y Revisión del Estado del Arte

1 Introducción

El desarrollo de Internet y las nuevas tecnologías han revolucionado la forma en que tendemos nuestra sociedad, dirigiéndola hacia lo que ya se ha dado a conocer como *la Era de la información y las telecomunicaciones*. En este contexto, la facilidad de acceso y recuperación de la información se han convertido en un tema de innegable interés. Dado que probablemente el Lenguaje Natural (en adelante LN) representa el mecanismo de comunicación más eficaz y conveniente para los humanos, cómo simular un proceso de comprensión automática del LN ha despertado el interés de la comunidad investigadora. Puesto que el lenguaje verbal tal y como lo conocemos es fruto de un largo proceso evolutivo y cultural, el LN se presenta como un complejo sistema de comunicación que ha sido y es estudiado desde diferentes disciplinas como la lingüística, la psicología, o la neurociencia, en base a modelos tan diversos como los distintos niveles del lenguaje, los procesos cognitivos, o la estimulación nerviosa de regiones cerebrales, respectivamente. Sea como fuere, su estudio supone un camino apasionante para llegar a entendernos y conocernos mejor a nosotros mismos.

Desde su aparición en la famosa Conferencia de Dartmouth de 1956, la *Inteligencia Artificial*, o IA, tiene como fin aproximar procesos inteligentes mediante mecanismos automáticos. Una de las diferentes ramas de la IA, la *lingüística computacional*, centra sus esfuerzos en el estudio del LN desde un punto de vista computacional, con la salvedad en este caso de que por primera vez los modelos propuestos pueden, además, ser implementados y simulados.

Aún así, el LN sigue siendo un sistema demasiado complejo como para ser abordado directamente. Por ello, muchos de los esfuerzos dedicados a su estudio se han centrado en los llamados *sistemas de dominio cerrado*. A diferencia de los sistemas de dominio abierto o general, los sistemas de dominio cerrado se caracterizan por gestionar una cantidad de información pertinente a un ámbito concreto, cuyo contenido puede ser gestionado de forma privada. De esta manera, los mecanismos de recuperación y comprensión asociados pueden beneficiarse de esta premisa para ofrecer un mejor servicio. Este tipo de sistemas es de particular interés para compañías y organizaciones, que encuentran en ellos una potencial solución al problema del acceso y gestión de su propio conocimiento.

Generalmente en IA, y en particular en los *Sistemas Expertos*, se mantiene una separación clara entre el proceso automático (o motor de inferencia) y el conocimiento. Nuestra propuesta de Tesis se centra principalmente en esta diferenciación. A menudo, el conocimiento del dominio se presenta como un recurso predefinido, y son los mecanismos automáticos los que suelen adaptarse a este. Siendo conscientes de ello, los procesos de IA pueden aplicar estrategias particulares para sacar el máximo rendimiento posible del conocimiento en favor de la aplicación. Ocasionalmente, se dispone del tiempo y los medios suficientes para crear recursos adicionales de conocimiento con los que explotar aún mejor el dominio. Este proceso, sin embargo, requiere de cierto esfuerzo y abordarlo mediante estrategias eficaces puede suponer una mejora sustancial tanto de cara al rendimiento de la aplicación como de cara a reducir su coste de creación. Nos proponemos, por tanto, enfocar este trabajo a las distintas estrategias automáticas que pueden llevarse a cabo para el desempeño de tecnologías del LN en función del conocimiento disponible.

Internet supone una fuente de conocimiento inagotable y un medio de comunicación integrado ya en nuestra sociedad actual. Las tecnologías basadas en LN se presentan como un medio para almacenar, acceder y consultar la información de forma interpretable e intuitiva. Existen actualmente multitud de tecnologías basadas en LN, algunas muy avanzadas, que influyen activamente en nuestra rutina diaria, de entre las que podemos destacar a los buscadores Web. Sin embargo, las exigencias de información de nuestra sociedad van más allá, y la demanda de aplicaciones más sofisticadas y nuevas soluciones es una constante. Tomemos por ejemplo la toma de decisiones de un potencial comprador interesado en un producto. Sería de gran interés una aplicación capaz de generar un resumen interpretable de las valoraciones de otros compradores. Imaginemos, por otro lado, el ahorro de tiempo que supondría contar con un ayudante virtual en una web, capaz de interpretar y responder a nuestras preguntas en LN. Esto nos ahorraría esfuerzo y tiempo evitando las búsquedas manuales por dominios web, a veces complejos y enrevesados. Supongamos por último lo ventajoso que resultaría acceder en LN a una Base de Datos sin necesidad de conocer su estructura ni el lenguaje formal de consulta subyacente. Estos ejemplos, y tantos otros que abordaremos en este texto, representan solo algunos casos de problemas abiertos de gran interés social y para los que, aunque ya hay algunas propuestas sólidas en la literatura, aún queda un margen de mejora considerable. Y son estos, precisamente, los problemas que motivan esta Tesis y para los que pensamos que el diseño de técnicas específicas en función del análisis del conocimiento disponible puede aportar un valor tangible.

Esta Tesis doctoral tiene por finalidad identificar y analizar algunas de las principales dificultades de los sistemas basados en LN de dominio cerrado, aportando avances científicos al estado del arte en forma de soluciones tecnológicas a problemas abiertos y de interés social. El número de problemas abiertos y demandados en el ámbito de las tecnologías de LN es muy elevado. Sin embargo, la perspectiva de análisis que proponemos en función del los niveles de conocimiento nos permite establecer una clasificación de problemas y métodos que nos será muy útil para identificar dificultades comunes y estrategias particulares. Nuestra *metodología* consistirá en seleccionar y abordar un problema representativo y de justificado interés de cada clase de problemas. Nuestro *objetivo* es doble: (i) proponer soluciones y técnicas basadas en el aprovechamiento del conocimiento para mejorar el estado actual del problema escogido, y (ii) abstraer conclusiones y extender la estrategia para hacerla potencialmente útil al resto de problemas del mismo grupo.

A diferencia de los sistemas de dominio abierto, la delimitación del dominio impone de forma implícita una serie de restricciones a la aplicación en cuestión, pero también es una fuente de conocimiento *a priori* que puede ser explotada en beneficio del sistema. Podemos distinguir dos recursos bien diferenciados dentro de los sistemas de dominio cerrado: el *conocimiento*, y el *metaconocimiento* del dominio. El conocimiento (o base de conocimiento), hace referencia a la información recogida explícitamente en el dominio. Es por tanto el principal recurso que justifica y da interés a la herramienta en sí, y puede presentarse de forma *estructurada* o *desestructurada*.

Ejemplos de representaciones de conocimiento pueden ser una Base de Datos o un conjunto de documentos. Por otro lado, el metaconocimiento hace referencia a información adicional que explica al propio conocimiento. Este recurso, de existir, puede ser explotado para realizar inferencias de índole semántica y razonamiento automático. Las ontologías, los lexicones de terminología propia del dominio o los conjuntos de *keywords* destacando los conceptos más relevantes del dominio, pueden servir como ejemplos de metainformación.

Comenzaremos con un estudio preliminar desde los niveles de conocimiento más bajos disponibles (documentos no estructurados y ausencia de metaconocimiento). En este primer caso se enmarcan todas aquellas aplicaciones que se enfrentan a grandes volúmenes de documentos sin ningún tipo de información adicional. Intentaremos dar respuesta aquí a algunas cuestiones importantes como *¿qué* características confieren relevancia a un término en un dominio concreto?, o *¿cuáles* son los términos más importantes del dominio? El estudio de este problema tiene un interés justificado por la gran cantidad de aplicaciones que se valen del concepto de *importancia* de una palabra. La clasificación de documentos, el resumen automático de textos o la recuperación automática de documentos, son solo algunos ejemplos de aplicaciones concretas en este ámbito. Enfocaremos nuestro estudio siguiendo la tendencia actual que plantea el problema como una Selección de Características. En concreto, nos valdremos de las llamadas *funciones de filtrado* y *políticas de selección* para evaluar la eficacia de la correlación positiva como criterio de selección. Este estudio nos proporcionará una base sólida sobre la que desarrollar métodos sofisticados para explotar los niveles de conocimiento superiores.

Continuaremos nuestro análisis abordando dificultades particulares en base a niveles intermedios de conocimiento, poniendo especial énfasis en el problema del modelado y refinado del conocimiento. Uno de los principales problemas de análisis a los que se debe enfrentar un sistema de LN consiste en la identificación de entidades en un documento escrito, esto es, *¿de qué* y *de quién* se habla en un texto? Para esto, explotaremos un recurso de gran utilidad en sistemas de dominio cerrado: los lexicones jerárquicos (diccionarios estructurados de términos y conceptos relacionados). En concreto, abordaremos el resumen de opiniones, de sumo interés de cara a estrategias empresariales, campañas políticas, y toma de decisiones de consumidores potenciales. Por medio de técnicas específicas de análisis, exploraremos soluciones a varios problemas específicos del LN, como la resolución de anáfora, elipsis, y desambiguación, que nos permitirá identificar los focos de análisis de forma más fiable. Nos plantearemos cómo mejorar el análisis automático de opiniones en noticias por medio de una delimitación previa del contexto.

Seguiremos nuestro estudio con los sistemas colaborativos, donde generalmente el conocimiento se estructura en Unidades de Información. Este tipo de sistemas cobra un especial interés en el marco de la Web 2.0 en general y el *e-learning* en particular. En este tipo de sistemas, el usuario deja de ser un mero consumidor de información para tomar parte activa en la generación de nuevo conocimiento. Investigaremos en este caso cómo crear sistemas escalables cuyo conocimiento crece de forma incremental y de acuerdo a las necesidades de información de sus usuarios. Más concretamente, aportaremos técnicas para reducir eficientemente los costes asociados al mantenimiento de esta clase de sistemas. Para ello, investigaremos cómo dotar de interpretabilidad a mecanismos formales ampliamente utilizados en tecnologías de LN, como son las expresiones regulares.

Concluiremos el estudio analizando aquellos sistemas basados en LN que parten de unos niveles de conocimiento más avanzados (conocimiento estructurado y metaconocimiento disponible), considerando en este caso el fenómeno del aprendizaje del lenguaje por medio del razonamiento por analogía. Lograr un nivel de interpretación avanzada del LN es un paso crucial para muchas tecnologías de acceso a la información, como los Asistentes Virtuales, los Pacientes Simulados o las Interfaces en Lenguaje Natural (de los que hablaremos más adelante). La finalidad última en estos sistemas es permitir a los usuarios interactuar con el sistema en LN para diferentes objetivos. A este respecto, las gramáticas formales han demostrado ser un mecanismo eficaz para alcanzar interpretaciones avanzadas del LN. Propondremos aquí un método nuevo de inferencia gramatical, basado en el razonamiento por analogía de casos, que permitirá simular un proceso de deducciones, y lo más importante, de *hipótesis* frente a sentencias nunca vistas. Por medio de una aplicación concreta, exploraremos nuevas soluciones al problema del acceso a la información, al tiempo que abordaremos problemas de índole lingüística pertinentes al aprendizaje del lenguaje.

No es objetivo de esta Tesis realizar un exhaustivo análisis de todas las tecnologías y métodos disponibles desde un punto de vista teórico, sino proponer soluciones concretas en forma de aplicaciones informáticas a problemas abiertos y demandados por la sociedad de la información. Este trabajo es, por tanto, el fruto de un proceso de ingeniería de índole científica. Con esto, pretendemos enfatizar el carácter de aplicabilidad y transferencia tecnológica del estudio. Esto es, gran parte de los resultados obtenidos han sido finalmente desarrollados como aplicaciones comerciales. Por otro lado, los desarrollos teóricos y métodos propuestos en este trabajo han sido, en su mayoría, publicados en diferentes revistas científicas y expuestos en congresos.

Este proyecto de Tesis ha sido realizado dentro del Máster oficial de Doctorado *Soft Computing y Sistemas Inteligentes* del programa de Doctorado oficial titulado “Ciencias de la Computación y Tecnología Informática” del Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada, bajo la dirección de los doctores D. Juan Luis Castro Peña y D. Jose Manuel Zurita López. El proyecto, de cuatro años de duración, ha sido financiado mediante la beca FPU (Formación de Profesorado Universitario) del Ministerio de Educación y Ciencia, y ha sido desarrollado en el Centro de Investigación en Tecnologías de la Información y las Comunicaciones (CITIC-UGR).

Dedicaremos este capítulo a revisar el Estado del Arte en lo referente a los sistemas basados en LN de dominio cerrado, y a exponer los objetivos de Tesis y la metodología seguida. El resto de la memoria de Tesis se estructura de la siguiente manera. En el capítulo II presentaremos un estudio preliminar desde los niveles mínimos de conocimiento disponibles. El capítulo III se dedica a aquellos sistemas en los que existe metaconocimiento, pero trabajan con documentos no estructurados. En el capítulo IV nos centraremos en los sistemas colaborativos e incrementales. El capítulo V está dedicado al aprendizaje y razonamiento por analogía partiendo del mayor nivel de conocimiento. Finalmente, presentaremos las conclusiones y trabajos futuros en el capítulo VI .

2 Estado del Arte

En esta sección presentamos una revisión del estado del arte de los sistemas de LN en dominio cerrado en función de los niveles de conocimiento. Plantearemos aquí una visión global, y se ofrecerá una visión más detallada y específica en los capítulos correspondientes.

2.1 Desde el Nivel de Conocimiento Más Bajo

La mayor parte de la información disponible en Internet no presenta una estructura conocida de antemano y no hay ningún nivel de metaconocimiento disponible para ella. Por esto, los sistemas de LN con conocimiento no estructurado y sin modelado de conocimiento representan una de las configuraciones de conocimiento con más aplicabilidad actualmente. Aparte de esto, la principal ventaja que ofrecen es que no requieren de ningún prerrequisito sobre los datos, esto es, no hay que generar ningún recurso adicional. Como contrapartida, la mayor parte de estos métodos suele requerir un corpus etiquetado de documentos de ejemplo del cual deducir un criterio de clasificación o comportamiento, dependiendo del caso. Este tipo de aprendizaje suele llamarse *supervisado*. Si bien es cierto que el etiquetado de documentos por parte de un experto es una tarea mucho menos restrictiva desde un punto de vista técnico, lo cierto es que la generación de estos corpus conlleva un proceso tedioso. Además, pese a que los algoritmos sean independientes del conocimiento, la disponibilidad del corpus de entrenamiento impone de forma implícita una dependencia con el dominio y restringe su uso a dominios de conocimiento estables.

Debido a que no hay información *a priori* sobre el conocimiento, los algoritmos se diseñan de forma independiente de los datos. Gracias a esto, existen una amplia variedad de algoritmos que son fácilmente adaptables a multitud de problemas. La mayor parte de los métodos que trabajan con LN bajo esta configuración aplican algoritmos de Aprendizaje Automático, más conocido como *Machine Learning* (ML) en la literatura científica [Mit97]. Entre las principales técnicas de ML cabe destacar las Redes Neuronales, Máquinas de Vectores Soporte, Computación Evolutiva, algoritmos de Agrupamiento, Árboles de Decisión y las Redes Bayesianas. Estas técnicas han sido aplicadas a diferentes problemas de LN, entre los que destacan:

Clasificación de Textos: es el problema de asignar cero, una, o varias etiquetas preestablecidas a un documento. La etiquetas representan distintas clases o categorías de documentos relacionadas semánticamente [Seb02, Seb05].

Agrupamiento de Textos: se define como la tarea de identificación de agrupamientos naturales de documentos, esto es, subconjuntos de documentos que tratan materias comunes. [LLCM03, AZ12].

Resumen de Textos: consiste en aplicar un procedimiento automático para reducir un texto original creando un nuevo texto en el que se reflejan únicamente los aspectos más importantes del texto original [NM01, DM07].

Recuperación de Textos: es una rama del campo de Recuperación de Información que consiste en recuperar los documentos, o fragmentos de documentos, más relacionados con una consulta formulada en LN por un usuario [SB88, Kor97].

Clasificación Ordinal de Textos: consiste en determinar de forma automática la posición de un documento en una escala de valores o etiquetas ordenadas [MWL$^+$07, BES10].

Análisis de Opiniones: estudia el análisis automático de opiniones y emociones expresadas en textos desde un punto de vista computacional. Generalmente, se atiende en el análisis a la polaridad (positivo o negativo), subjetividad e intensidad de las opiniones [PL08, CZ10]. Dedicaremos parte de este estudio al problema del análisis de opiniones en el capítulo III, sin embargo lo abordaremos desde un nivel de conocimiento diferente, por medio de las llamadas técnicas basadas en lexicon.

Clasificación de Preguntas: consiste en determinar automáticamente la clase más probable a una pregunta en función del tipo de respuesta esperado [Her01, HW03].

Generalmente, los documentos suelen representarse en un espacio de características o espacio vectorial. Las características se generan a partir de los términos que aparecen en el documento. Este proceso se decide en la etapa de *Extracción de Características* pero, de una u otra manera y debido a la enorme variedad léxica del lenguaje, lo más común es que este espacio vectorial sea de una dimensión muy elevada, del orden de millares o incluso decenas de millares. A pesar de que los algoritmos y tecnologías disponibles en la actualidad nos permitirían lanzar procesos de inferencia aún en estas circunstancias [Joa02, Joa06], se ha observado que es posible mejorar el rendimiento del método partiendo de un espacio reducido de características [YP97]. Dedicaremos parte de este estudio a esta subtarea conocida como Selección de Características (capítulo II).

2.2 Incorporando Información Lingüística

Podemos considerar que las técnicas de ML trabajan "a ciegas" en el sentido de que no disponen de ningún tipo de ayuda para enfrentarse al LN. Sin embargo, pese a que no existe ninguna representación o conocimiento previo del dominio salvo un conjunto de datos de entrenamiento etiquetados, hay técnicas que incorporan conocimiento lingüístico en el proceso de creación de las características. Por ejemplo, en [ZL03] se propone un algoritmo de clasificación de preguntas que incorpora información sintáctica, y en [BM07, LR06] se incorpora, además, información semántica de las palabras. También en clasificación de textos, [ÖG10] considera patrones de dependencia léxica en el espacio de características.

Aparte del análisis sintáctico y del análisis morfológico (asociado este último a las llamadas técnicas de PoST, del inglés *Part of Speech Tagging*) la principal vía para incorporar información lingüística e información del dominio al problema consiste en considerar diccionarios, tesauros o lexicones de dominio. Un diccionario es un listado plano de términos y definiciones junto, posiblemente, con listados de sinónimos, antónimos, etc. Un tesauro es un conjunto de términos empleados para representar conceptos junto con una serie de relaciones jerárquicas y semánticas entre ellas (como WordNet [MBF$^+$90] o Wikipedia[1]). Finalmente, un lexicón es un conjunto de términos junto con sus entradas léxicas y reglas morfológicas [Lie80].

Es obvio que este tipo de recursos lingüísticos y semánticos suponen una ayuda para cualquier tratamiento automático de documentos [MSC11]. Aunque existen lexicones y diccionarios genéricos y de acceso público [SKA68, HL04a, ES06b, WWH05], el significado y uso de las palabras puede variar entre dominios. Por esto, hay aplicaciones que requieren de un recurso específico y adaptado para el dominio. A este respecto, se han propuesto numerosas técnicas abordando el problema de la identificación de nuevos conceptos y expresiones [LMS06, NPI11, BV04], la adaptación del significado entre dominios [TW11a], o la adaptación de lexicones preexistentes [QLBC09, ALSZ06, ALM$^+$03]. Según [Liu12], las principales técnicas para crear estos lexicones pueden clasificarse como:

[1] http://en.wikipedia.org/

Métodos Manuales: a pesar de que suponen un esfuerzo considerable, permiten monitorizar exhaustivamente el funcionamiento del sistema. Es común que estas técnicas se combinen con métodos semiautomáticos destinados a expandir un vocabulario inicial creado a mano.

Métodos basados en Diccionarios: estas técnicas se valen de diccionarios genéricos para generar listas de sinónimos y antónimos pertinentes a un dominio de conocimiento concreto. Generalmente, se define un conjunto de "palabras semilla" a mano que es posteriormente expandido explorando las relaciones semánticas en WordNet. Algunas técnicas relevantes siguiendo este método pueden consultarse en [Tur02a, ES06a, KH06, VBGHM10].

Métodos basados en Corpus: la principal motivación de estas técnicas consiste en adaptar recursos genéricos a un dominio de conocimiento particular, teniendo en cuenta que el significado de ciertas palabras puede variar entre dominios. Algunos ejemplos de estas técnicas pueden encontrarse en [HM97, WWH05, KN06].

2.3 Incorporando Información Semántica

Diferentes autores han ido reciclando el concepto de lexicón de dominio, adaptándolo según las necesidades a problemas más sofisticados. Un ejemplo de este fenómeno podemos encontrarlo en el Análisis de Opiniones sensible a Características, más conocido en la literatura científica como *Feature-based Sentiment Analysis* [PL08, Liu12]. Este problema derivado del Análisis de Opiniones pretende realizar un análisis de subjetividad desglosado para los principales tópicos tratados en los documentos de opinión. Generalmente se ha considerado en el contexto de evaluación [HL04a, GSS07] o comparación de productos [PE05]. Por ejemplo, frente al posible comentario "La cámara es genial, pero su batería no acaba de convencerme", sería de esperar un resumen reflejando ⟨Cámara,Positivo⟩ y ⟨Cámara.Batería, Negativo⟩. Para esto, se han propuesto estructuras jerárquicas de conceptos por medio de relaciones PART-OF (relaciones objeto-componente [JL06]) e IS-A (relaciones clase-subclase [MRCZ12b]) que incluso pueden tener propiedades [KH04]. Trataremos en más detalle este tipo de métodos en el capítulo III.

A medida que los lexicones estructurados se valen de mecanismos más sofisticados para representar la semántica de relaciones entre las entidades del dominio, se aproximan conceptualmente a las *ontologías de dominio*. Las ontologías surgen como la evolución natural de diferentes mecanismos precursores de representación, entre los que cabe citar las redes semánticas, los marcos, los diagramas Entidad-Relación o los modelos Orientados a Objetos. Las ontologías surgen como un medio para representar los diferentes recursos digitales disponibles en la *Web Semántica* y suelen definirse como una *conceptualización* del conocimiento de una *parte del mundo* que recoge, mediante un vocabulario de dominio, los objetos, propiedades, relaciones y entidades de un dominio concreto. Su especificación permite, además, la inferencia de conocimiento por razonamiento automático. Se tratará en mayor detalle el concepto de ontología en el capítulo V sección 1.3.

La utilización de ontologías de dominio como soporte para el procesamiento del lenguaje se ha extendido a lo largo de numerosas disciplinas y métodos dentro del marco de las tecnologías del LN. Tanto es así que resultaría pretencioso intentar abordar en este texto una cantidad representativa de los aportes científicos derivados del uso de ontologías en las tecnologías de LN. Comentaremos en su lugar solo aquellas técnicas más representativas y que son más afines al trabajo desarrollado en esta Tesis y redirigiremos al lector interesado a [Bat97]. Concretamente, nos centraremos aquí en su utilización para la interpretación de preguntas en LN.

De acuerdo con [AS05], uno de los métodos principales para interpretar preguntas en LN bajo dominios cerrados puede clasificarse dentro de las llamadas estrategias de Procesamiento del Lenguaje

Natural (PLN) [ID10]. Los métodos PLN tratan de alcanzar una representación formal del LN a fin de lograr una interpretación completa del lenguaje y, a menudo, se valen de ontologías de dominio para ello. Entre los principales retos a que se enfrenta el PLN podemos destacar la traducción automática, la resolución de correferencias, el reconocimiento de entidades nombradas, la generación automática de lenguaje o el análisis gramatical. A pesar de que este tipo de técnicas ha sido aplicado a diferentes problemas como la recuperación de FAQs [WTRM08, MHG$^+$02, WMC09, TR08], del que hablaremos extensamente en el capítulo IV, lo cierto es que las tecnologías NLP junto con las ontologías cobran un especial interés en los Asistentes Virtuales, Pacientes Simulados Virtuales e Interfaces en Lenguaje Natural.

Asistentes Virtuales: son agentes conversacionales dedicados a responder preguntas sobre un dominio web [ELC12]. La principal ventaja de este tipo de sistemas consiste en liberar al usuario de realizar búsquedas manuales y, por tanto, de conocer la manera en la que la información se estructura en la web. Además, este tipo de agentes puede incorporar sistemas de recomendación, y generalmente pretenden simular comportamientos y reacciones humanas [ELC09].

Pacientes Simulados Virtuales: son agentes computacionales utilizados para el entrenamiento de diagnóstico en cursos médicos [LEC08, SPG09]. Estos sistemas intentan simular la manera en que un paciente humano respondería frente a preguntas del médico, algunas potencialmente embarazosas para las cuales el paciente podría no responder con sinceridad.

Interfaces en Lenguaje Natural: son sistemas para facilitar el acceso a la información almacenada en bases de conocimiento estructuradas a usuarios no expertos. En lugar de utilizar complejos mecanismos de consulta como SQL, SPARQL o XPath, los usuarios pueden formular sus preguntas en LN o utilizar algún mecanismo de entrada derivado del LN para ello [ART95, PRGBAL$^+$13].

Al igual que en el caso de los lexicones de dominio, la creación de ontologías requiere de un análisis apropiado del problema y es por tanto un proceso tedioso que suele quedar restringido a Ingenieros del Conocimiento. No obstante, se han propuesto varias técnicas, como [SB04], para crear automáticamente ontologías a partir de textos. Muchas de estas técnicas se tratan en detalle en [BCM05, MS01]. La creación o adaptación de una ontología suele ser una de las subtareas a seguir en la etapa de *configuración* de un sistema de LN frente a un dominio, que suele incorporar otras tareas como la definición de una lexicalización [MSC11, BCM$^+$11], el mapeado entre elementos del dominio con construcciones lingüísticas [GAMP87] o la mejora de la *habitabilidad* (capacidad del sistema para interpretar la variabilidad lingüística que los usuarios utilizan para interactuar con él) del sistema [Wat68, OB96, OMBC06].

Resulta complejo, incluso para un experto en tecnologías de LN, predecir la variedad de casos a los que el sistema podría enfrentarse. Este problema, que presenta al experto como un mediador entre las expectativas del usuario y la extensión de la base de conocimiento, suele denominarse en inglés *the bridge the gap problem*, que podríamos traducir aquí como “rellenar el hueco” [Hal06, WXZY07]. Una estrategia para evitarlo consiste en limitar las posibles construcciones lingüísticas que el usuario puede introducir, mediante mecanismos de entrada basados en menús o por bloques [BKK05, TPT05]. Otra estrategia para abordarlo consiste en diseñar técnicas de configuración interactivas, que permitan al experto depurar el sistema por medio del uso. Este tipo de sistemas puede utilizar diálogo para demandar información al experto [GAMP87, CHH07] o clarificar ambigüedades [DAC10, DAC12]. Puede encontrarse un tratado sobre los sistemas interactivos hombre-máquina en [HLP97]. Dedicaremos el capítulo V de este trabajo de Tesis a investigar nuevas técnicas de configuración del conocimiento basadas en la interacción y el aprendizaje.

2.4 Incorporando Información Sintáctica

Desde un punto de vista analítico, podemos considerar que tanto los lexicones como las ontologías de dominio permiten incorporar información de tipo morfológico y semántico. Concluiremos este recorrido por los distintos niveles de conocimiento de los sistemas de LN en dominios cerrados centrándonos en una forma diferente de agregar información al problema pertinente al nivel sintáctico (del que ya hicimos algunos apuntes referentes a los métodos de ML).

Podemos considerar que existen dos formas principales de incorporar información sintáctica en el dominio: la implícita y la explícita.

2.4.1 Sintaxis Implícita: Estructura

La *implícita* se refleja mediante la organización propia del conocimiento. No necesariamente nos referimos en este caso a sintaxis desde el punto de vista lingüístico sino, más bien, a la sintaxis interna en que se almacenan los datos, esto es, su estructura. Este enfoque no es excluyente con respecto a los métodos de modelado antes mencionados, y de hecho encontramos ejemplos combinando todos ellos en el problema de las Interfaces de Lenguaje Natural [WXZY07, CHH07] y *Question Answering* [Yan09b], entre otros. Sin embargo, también existen ámbitos en los que prescindir de cualquier otro modelado de conocimiento permite diseñar aplicaciones incrementales, y es este posiblemente el enfoque más eficaz para el diseño de sistemas colaborativos.

Los sistemas colaborativos evolucionan a partir del desarrollo de la Web 2.0 como un mecanismo que permite a los usuarios no solo ser consumidores de la información, sino también productores de la misma. De esta forma, la información añadida por un usuario es potencialmente útil para otros. Este tipo de sistemas es de particular interés en campos como el *e-learning* [MRPVR07] en el que los llamados *Virtual Learning Environments* (VLE) o *Learning Management Systems* (LMS) permiten a profesores y alumnos mejorar la experiencia de enseñanza y aprendizaje [AHBLGS$^+$12, DSS$^+$02] mediante aplicaciones como *WebCT*[2], *Blackboard*[3] o *Moodle*[4].

Una forma intuitiva de acceso a este contenido es mediante la consulta en LN. Algunos ejemplos de este tipo de sistemas son *Stackoverflow*[5] o *Yahoo! Answers*[6]. Este tipo de sistemas se vale de la premisa de que su conocimiento se almacena estructurado en listados de preguntas y respuestas posiblemente agrupados en categorías. Este problema se conoce generalmente como recuperación de FAQs y dedicaremos el capítulo IV a proponer métodos escalables de recuperación basados en LN y técnicas de mejora del conocimiento. Frente a otros métodos que incorporan modelado de conocimiento como [LFQL11, LLL10, HSCD09], nosotros abordaremos el problema desde un punto de vista meramente estructural, en la línea de otros sistemas como [BCC$^+$00, KS08a, KS06]. La principal ventaja de este enfoque radica precisamente en que al no requerir de un nivel de metaconocimiento, el sistema evoluciona con la simple adición de preguntas y respuestas que se deriva de su uso.

[2] www.WebCT.com

[3] www.blackboard.com

[4] www.moodle.org

[5] http://stackoverflow.com/

[6] http://answers.yahoo.com/

2.4.2 Sintaxis Explícita: Patrones

Por último, el mecanismo *explícito* de descripción sintáctica implica la especificación de patrones lingüísticos. Estos patrones, generalmente conocidos como *plantillas*, pueden utilizarse tanto para interpretar frases como para extraer respuestas. Las plantillas generalmente se definen por medio de *expresiones regulares*, un mecanismo formal propuesto en 1950 por Stephen Cole Kleene y de extensiva aplicabilidad en el ámbito de las tecnologías del LN.

A pesar de que muchos de estos métodos se utilizan comúnmente en problemas de dominio abierto, como la extracción de respuestas en *Question Answering* [RH02, ZL02, SGS06, CKC07] o la clasificación del tipo de pregunta en base al tipo de la respuesta esperada [Her01, HW03], también juegan un papel importante en los sistemas de dominio cerrado. Los sistemas de plantillas propuestos por Sneiders son, probablemente, los ejemplos más representativos. Entre ellos encontramos aplicaciones al problema de la clasificación automática de emails [Sne10], consulta a bases de datos [Sne02] y recuperación de FAQs [Sne09], que abordaremos en mayor detalle en el capítulo IV.

Los mecanismos basados en plantillas han probado ser muy robustos frente a diferentes tareas [DSS11, Sou01] puesto que encapsulan de forma eficaz el conocimiento del experto que las diseña. Sin embargo, su construcción es una labor tediosa que suele realizarse a mano. Los principales mecanismos de generación automática de plantillas, o inferencia de expresiones regulares, se han propuesto bien desde campos alejados del tratamiento del lenguaje, como la inferencia de descripciones de datos DTD o descripciones de esquemas XML [Fer09, BNST06, BGNV10], o bien para extraer la respuesta en dominios abiertos (*surface text patterns*, [RH02, ZL02]), por lo que ninguno explota directamente características propias de las preguntas en LN. Propondremos en en el capítulo IV métodos específicos para la generación y refinado de expresiones regulares para el reconocimiento de preguntas en LN.

3 Objetivos

El objetivo de esta Tesis es aportar soluciones específicas a diversos problemas abiertos y de interés en el marco de los sistemas de dominio cerrado basados en lenguaje natural, desde un punto de vista puramente práctico. Con esto, pretendemos enfatizar el carácter de *aplicabilidad* de este trabajo. Nuestro segundo objetivo consiste en extender los resultados particulares a una clase de problemas más general. Para conseguir estos fines, planteamos los siguientes objetivos transversales:

- Estudiar los sistemas de lenguaje natural en dominios cerrados atendiendo a los niveles de conocimiento y metainformación disponibles, a fin de identificar problemas reales característicos y proponer soluciones concretas en forma de aplicaciones informáticas.
- Investigar técnicas para explotar eficazmente los distintos grados de conocimiento disponibles en el dominio en cada caso: estructura y metaconocimiento.
- A menudo, adaptar un sistema a un dominio conlleva un gran esfuerzo y mucho tiempo. Por ello, las estrategias deben ser, en la medida de lo posible, independientes del dominio y capaces de explotar información reutilizable entre dominios.
- Aportar valor al estado del arte en cada caso por medio de metodologías novedosas. Desarrollar aplicaciones completas a partir de la implementación de estos métodos y validar las propuestas en entornos reales. Proponer contribuciones a la literatura científica.

De forma más específica, pretendemos abordar los siguientes objetivos parciales:

- Identificar los términos más relevantes de un dominio suele ser una subtarea de la que dependen muchas aplicaciones de LN. Con respecto a esto, nos planteamos investigar la influencia de la correlación positiva como componente destacada para la identificación de los términos más importantes de un dominio. Esto nos facilitará abordar los siguientes de niveles de conocimiento.
- Identificar los focos del discurso es también una tarea importante de cara a muchos procesos de análisis del NL. Por ello, proponemos investigar el impacto que se deriva de un análisis previo del contexto. Explorar nuevos mecanismos de resolución de anáfora, elipsis y ambigüedad para mejorar la identificación de focos.
- Investigar técnicas para reducir los costes de creación, mantenimiento y refinamiento de sistemas colaborativos. Concretamente, nos marcamos como objetivo diseñar métodos escalares de recuperación de información y mecanismos de adaptación frente a potenciales cambios en el dominio de conocimiento.
- El conocimiento de los sistemas colaborativos debe crecer de acuerdo a las demandas de información de sus usuarios. Para ello nos proponemos diseñar métodos de minería de uso que permitan la monitorización del conocimiento actual en sistemas colaborativos a fin de proveer al responsable del sistema de indicaciones útiles para mejorar el sistema.
- Dotar a los sistemas de la capacidad de monitorización experta para aquellos casos en los que garantizar un mejor funcionamiento justifique ciertos costes adicionales. Esto es, los mecanismos deben ser interpretables y revisables por un humano.
- Establecer mecanismos robustos y protocolos de interacción para reducir el coste de configuración de sistemas de LN y para permitir la portabilidad entre diferentes dominios de conocimiento.

4 Metodología

Esta sección se dedicada a presentar la metodología de análisis y desarrollo que se ha seguido a lo largo de esta Tesis.

Basaremos nuestro estudio en el análisis de los procesos automáticos de LN en función de los niveles de información: conocimiento, y metaconocimiento. El conocimiento representa la información propia del dominio, y puede presentarse de forma estructurada (una base de datos, o un fichero XML por ejemplo) o desestructurada (un corpus de documentos, o un conjunto de comentarios). El metaconocimiento establece un nivel superior de información sobre la información. Por ejemplo, una ontología definiendo las principales entidades y relaciones en un dominio, o un lexicón recopilando la terminología propia de un ámbito. Utilizaremos indistintamente los términos *metaconocimiento*, *metainformación* o *modelado de conocimiento* para referirnos aquí a este recurso.

Vamos a centrar nuestro análisis en estas dos características. Por un lado, atenderemos a la ausencia o presencia de estructura en su conocimiento. Por otro lado, atenderemos a la disponibilidad de metaconocimiento de dominio.

Esta metodología nos permite establecer una clasificación clara entre las diferentes técnicas de LN. Realizaremos un estudio centrado en identificar las principales características y dificultades presentes en cada configuración de conocimiento, por medio de un problema concreto. Este problema se seleccionará atendiendo a dos características principales: (i) es un buen representante del grupo, y (ii) es un problema abierto de actual interés y demanda. Este problema nos servirá para desempeñar un estudio pormenorizado tanto de las dificultades presentes como de las posibilidades que se derivan del tratamiento del conocimiento disponible. En cada caso, propondremos: (i) contribuciones particulares al problema, y (ii) conclusiones de carácter general y extensibles al resto de problemas enmarcados dentro de la misma configuración de conocimiento.

De acuerdo con esto, seleccionamos los siguientes problemas representativos de cada grupo:

- La *Clasificación de Documentos* constituye un problema donde el conocimiento no presenta una estructura interna conocida, y no existe metaconocimiento más allá de simples etiquetas posibles para los documentos. A modo de simplificación, este problema consiste en clasificar un documento dado en una, varias, o ninguna categoría preexistente, a partir únicamente de la observación de ciertos documentos de ejemplo preclasificados por un experto. Centraremos nuestros esfuerzos en la selección de características, una etapa fundamental y previa al proceso de clasificación, dedicada a identificar y seleccionar los términos más relevantes del dominio de cara a la clasificación. El concepto de importancia de una palabra en un dominio es de importancia generalizada en el ámbito de tecnologías de LN.

- El *Análisis Automático de Opiniones* tiene por finalidad crear resúmenes útiles de opiniones de usuarios. Generalmente, las opiniones se presentan como documentos de texto libre (sin estructura conocida de antemano), y podría por tanto catalogarse este problema como un caso particular de *Clasificación de Documentos* en el que las clases representan posibles valoraciones (positivo o negativo, o una escala ordenada). Dado el interés que suscita este problema, se han propuesto multitud de variantes. Una de las cuales, propone identificar, además de la polaridad de la opinión, el objeto u objetos de debate. Para este fin, suelen utilizarse modelos de conocimiento como los lexicones de dominio. Un lexicón jerárquico es una taxonomía de entidades y propiedades junto con terminología propia del dominio. Seleccionaremos aquí este tipo de técnicas como representativas de sistemas sin estructura en el conocimiento y con metaconocimiento disponible.

- Por otro lado, existen sistemas en los que no existe metaconocimiento, pero el conocimiento sí guarda una estructura interna. Este tipo de sistemas tiene cierta presencia en entornos colaborativos, como el *e-Learning*. Un ejemplo representativo de este grupo de problemas es el llamado *FAQ retrieval*. Un FAQ (de sus siglas en inglés *Frequently Asked Questions*) es un listado de parejas pregunta/respuesta sobre un tema particular que pueden estar, además, organizadas en categorías. El problema consiste en recuperar las entradas del FAQ más relevantes con respecto a una pregunta realizada en lenguaje natural. Puesto que estos métodos deben ser escalables e incrementales, la ausencia de metaconocimiento facilita la adición de información de forma colaborativa por usuarios no expertos.

- La última combinación que consideramos viene dada cuando el conocimiento está estructurado, y hay metaconocimiento disponible. Posiblemente, uno de los problemas más representativos de esta configuración sean las Interfaces en Lenguaje Natural. Generalmente, estos sistemas intentan interpretar una pregunta en LN para generar una consulta formal frente a una base de datos (el conocimiento estructurado), ayudados de un esquema de relaciones semánticas, plasmado generalmente en una ontología de dominio (el metaconocimiento). Ya que estas interfaces facilitan un acceso intuitivo y rompen las barreras tecnológicas para usuarios inexpertos, el interés que suscita este tipo de sistemas es indiscutible.

Finalmente, proponemos la siguiente clasificación y problemas atendiendo a las características presentes en los diferentes niveles de conocimiento (Figura i.1)

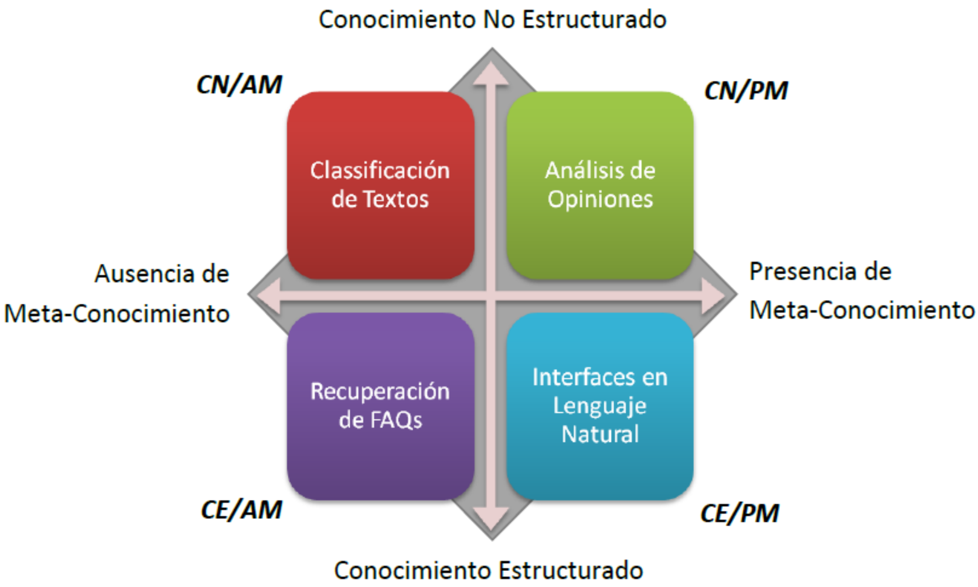

Conocimiento No Estructurado
CN/AM
CN/PM
Classificación de Textos
Análisis de Opiniones
Ausencia de Meta-Conocimiento
Presencia de Meta-Conocimiento
Recuperación de FAQs
Interfaces en Lenguaje Natural
CE/AM
CE/PM
Conocimiento Estructurado

Figure i.1: Selección de problemas en base a los niveles de conocimiento.

Entre nuestros objetivos principales se encontraba el de explotar los distintos niveles de conocimiento del dominio en beneficio de la aplicación. A este respecto, cabe resaltar que la combinación con Conocimiento No Estructurado y Ausencia de Metaconocimiento (CN/AM) ofrece el menor nivel de información disponible. Es por esto que hemos elegido esta combinación a modo de estudio preliminar más que como una posible aplicación en sí misma. Este estudio supondrá, sin embargo, el punto de partida hacia la exploración de los distintos niveles de conocimiento. Como cabe esperar y por coherencia, terminaremos el estudio con la combinación Conocimiento Estructurado y Presencia de Metaconocimiento (CE/PM), esto es, aquella que dispone del mayor nivel de información sobre el dominio.

La metodología general a seguir a lo largo de este trabajo seguirá las pautas siguientes de forma estricta y para cada uno de los diferentes casos de estudio:

- Discusión y particularidades del caso de estudio en base a los niveles de información del dominio. Ejemplos de aplicaciones.
- Justificación del problema seleccionado, interés en la actual sociedad de la información y definición formal.
- Exposición de los mecanismos formales que se utilizarán para abordarlo. Notación y descripción.
- Revisión del estado del arte.
- Propuesta del método. Implementación y evaluación experimental de la propuesta en contraste con métodos representativos del estado del arte.
- Extracción de conclusiones y propuesta de trabajos futuros.

Particularmente, proponemos la siguiente metodología a seguir en función de cada problema seleccionado:

Selección de Características: a fin de identificar un criterio de relevancia de palabras con respecto a un dominio, pondremos a prueba diferentes funciones de filtrado, políticas de selección, clasificadores, métricas de evaluación y bancos de datos para investigar la influencia de la correlación positiva en la selección de términos de un dominio.

Análisis Automático de Opiniones: realizaremos una delimitación previa del contexto de un grupo de comentarios sobre una noticia, con objeto de identificar los principales focos de opinión. Para ello, propondremos soluciones a los problemas de resolución de anáfora, elipsis y ambigüedad explotando la información semántica y sintáctica de un lexicon jerárquico del dominio.

Recuperación de FAQs: investigaremos los criterios de diferenciabilidad y minimalidad para diseñar un algoritmo de inferencia de expresiones regulares para la recuperación de las unidades de información. Diseñaremos luego técnicas de minería de uso para valorar la calidad del FAQ en función de las acciones y reacciones de los usuarios al interactuar con el sistema. Propondremos finalmente un método para generar plantillas interpretables a partir de expresiones regulares elaboradas, que permita a un experto revisarlas y modificarlas invirtiendo un mínimo esfuerzo.

Interfaces en Lenguaje Natural: nos valdremos de gramáticas libres de contexto para realizar las interpretaciones de las consultas en LN de los usuarios. Diseñaremos un método de inferencia gramatical basado en razonamiento por analogía para mejorar la capacidad de reconocimiento del sistema. Exploraremos técnicas basadas en el planteamiento de hipótesis y deducciones para hacer frente a términos, expresiones y construcciones lingüísticas nunca vistas por el sistema.

Chapter I

PhD Dissertation and Review of the State of the Art

1 Introduction

The development of the Internet and the continuous improvements in hardware technologies have brought a socio-cultural revolution, that is rather known as *The age of computers and communications*. In this context, how to facilitate access to information became an interesting challenge. Since Natural Language (NL) represents arguably the most convenient communication mechanism for humans, how to bring computers the ability to automatically understand NL represents a problem of the utmost importance in the scientific community. Given that current verbal language results from a large evolution, NL represents a complex communication system that has been and will be studied from a long variety of disciplines including linguistic, psychology, or neuroscience, from different perspectives such as the different levels of linguistic analysis, the cognitives processes, or the nerve stimulation, respectively. Be that as it may, the truth is that the study of NL represents an exciting challenge to better understand and comprehend ourselves.

Since the concept of *Artificial Intelligence* (AI) appeared in the famous Dartmouth Conference in 1956, AI has been devoted to approximate any sort of intelligent process through automatic mechanisms. Among the different branches in AI, *computational linguistics* concerns with the study of NL from a computational point of view. The main difference with respect to previous linguistic models is that these computational models could, for the first time, be implemented and tested.

In any case, NL still represents a too complex communication mechanism so as to be directly attempted. For this reason, many of the efforts dedicated to its study were made in the field of *closed-domains*. In contrast to open-domain systems, the information handled by closed-domain approaches concerns with a delimited field of knowledge and could, in addition, be privately managed. In this way, both retrieval and interpreting mechanisms could take advantage of these *a priori* conditions to offer a better service. This kind of systems is of particular interest for companies and organization as a potential solution to the access and management of their own knowledge.

Usually in AI, and more specifically in the so-called *Expert Systems*, there is a clear separation between the automatic processes and the knowledge. Our hypothesis focuses in this separation. The domain knowledge is often a predefined resource, and the automatic processes should be tuned accordingly. Being aware of this fact, AI processes could apply more sophisticated strategies to better approach the problem. Occasionally, one counts with the necessary time and means so as to create additional knowledge resources to better exploit the domain information. This is however a costly task that could be alleviated by means of certain methodologies that could, in addition, improve the final performance. It is therefore our aim to focus on the different automatic processes that could be carried out to develop rich NL applications in basis of the various knowledge levels.

The Internet represents an endless source of knowledge and a means of communication that has been strongly integrated as part of our current society. NL technologies could be regarded as a means for managing information in an interpretable and understandable manner. There is already a large quantity of NL technologies that influence actively our daily lives. However, the needs for information of the current society go further, and new and better solutions are constantly demanded. Let us take as an example the decision making for a potential purchaser interested in a particular product. An application capable for automatically summarizing other purchasers' opinions on this product would be of the utmost interest. As another example, a virtual web assistant could help users to resolve their queries by means of NL, avoiding the waste of time that navigating through a possibly complex web-domain may entail. It would also be desirable counting with a Database Management System capable for interpreting NL questions, freeing users for being aware of the particular Database structure or the underlying formal query language. These examples, along with so many others that will be later addressed in this dissertation, represent just some cases of open-problems of social interest, to which there is still a considerable room for improvement. These problems are precisely the motivation of this Thesis. We believe the design of specific techniques being aware of the different knowledge resources could add some value to the state of art.

This PhD dissertation is aimed for identifying and analysing main difficulties that arise on closed-domain NL approaches in order to propose valuable scientific contributions to the state of the art in form of technological solutions. The extent of interesting open-problems in the field of NL technologies is huge. Notwithstanding, the analytic perspective we propose in basis of the knowledge levels allow these methods to be naturally grouped. Our *methodology* consists of approaching each group from a representative problem of general interest. Our *goal* is two-fold: (i) to propose techniques and solutions exploiting the knowledge available in order to better approach the selected problem, and (ii) to abstract conclusions to make our methods become potentially useful and extensible to the rest of problems on its group.

In contrast to open-domain systems, the delimitation of the domain imposes an implicit restriction to the system at hand, but is also a controlled knowledge resource that could be efficiently exploited in favour of performance. We could differentiate two kinds of resources in a closed-domain: the *knowledge*, and the *meta-knowledge*. On the one side, the knowledge refers to all data explicitly stored in the domain. It is therefore the main source of information that justifies and brings interest to the application. In this regard, knowledge could be *structured*, e.g. a Database, or *unstructured*, e.g. a free-text document. On the other side, *meta-knowledge* refers to additional information explaining the data. This resource, if present, could be used to perform semantic inferences through automatic reasoning. Ontologies, lexicons collecting the domain terminology, or keywords sets highlighting main terms in a domain, could be some examples of meta-knowledge.

We start with a preliminary study on closed-domain NL approaches from the lowest knowledge resources available: unstructured knowledge with no meta-knowledge. This starting point groups together all those systems dealing with large volumes of document without any sort of additional information. In this study, we will investigate *what* makes a term become relevant to a given domain, and *which* are the main terms in a document. The interest of these open-questions lies on the huge quantity of applications that rely on the concept of term-relevancy. Text Classification, Automatic Text Summarization, or Text Retrieval, are just some examples in this regard. We will here adopt the current tendency whereby the problem is usually stated as a Feature Selection problem. Specifically, we will rely on the so-called *filtering methods* and *feature selection policies* to test the effectiveness of positive correlation as a predominant criterion for term-relevance. This study will bring us the necessary background upon which our subsequent analyses will be conducted.

We will continue our analysis on intermediate levels of knowledge, paying special attention to the modelling and refinement tasks. In this regard, one of the main concerns a NL process should deal with consists of the identification of the main topics in the discourse, that is, *what* is the text about, and *which* are the main entities involved? To this end, we will exploit a helpful resource for closed-domains: hierarchical lexicons —structured dictionaries collecting terminology and related concepts. Concretely, we will address a highly demanded problem: the automatic analysis and summary of opinions expressed in texts (aka Sentiment analysis or Opinion Mining), an issue of the utmost interest for companies strategies, political campaigns, and decision making for potential users. By means of specific analysis techniques, we will try to offer solutions to linguistic issues such as ambiguity, ellipsis, and anaphora resolution problems. We will investigate how to improve the automatic analysis of opinions on news items by previously delimiting the context.

Our study continues with collaborative systems, where knowledge is usually structured in separate Information Units. These systems result from the development of the so-called Web 2.0 and represent an interesting field to emerging paradigms such as the *e-learning*. Users are no longer mere consumers of information, but could also take an active part as producers of new knowledge. In this case, we will investigate how to create scalable collaborative systems in which the shared knowledge grows attending to actual users' information needs. More specifically, techniques for alleviating the costs associated to the creation and maintenance of collaborative systems will be offered. In pursuing this goal, we investigate how to bring interpretability to regular expressions, a broadly used formal mechanism in the context of NL technologies.

We end our study offering an analysis of NL approaches from the highest knowledge levels: structured knowledge with a suited meta-knowledge model available. In this case, we will focus on the learning phenomenon from the point of view of reasoning by analogy. Reaching an advanced interpretation of NL is a fundamental step for many NL technologies such as Virtual Assistants, Virtual Simulated Patients, or Natural Language Interfaces. The ultimate goal in this regard is to allow users interact with the system by means of NL in order to facilitate access to information. In this respect, formal grammars represented an effective tool to reach NL parsings. We will present a new grammar inference method based on reasoning by analogy to allow the system conjecture about the language while facing unseen expressions or terms. Through a concrete application, we will investigate new solutions to the information access problem paying attention to some relevant linguistically-related problems.

Providing an exhaustive analysis of all NL technologies and approaches from a theoretical point of view falls beyond the scope of this Thesis. We are rather interested in proposing concrete solutions to real problems in form of computer applications. Therefore, this PhD Thesis results from an engineering process of scientific nature. With this, we intend to reinforce the applicability nuance of this work. On the one side, most of the results obtained through this research have finally been developed as commercial applications. On the other side, most methods and theoretical advances presented in this dissertation have been published in different scientific journals or conferences.

This Thesis project has been developed as part of the official PhD Master degree on *Soft Computing and Intelligent Systems* under the PhD official program *Ciencias de la Computación y Tecnología Informática* of the Department of Computer Sciences and Artificial Intelligence of the University of Granada, under the tutelage of Professor Juan Luis Castro Peña and Jose Manuel Zurita López. This four-years project has been funded by the FPU (*Formación de Profesorado Universitario*) scholarship of the *Ministerio de Educación y Ciencia*, and has been developed in the CITIC-UGR (*Centro de Investigación en Tecnologías de la Información y las Comunicaciones*).

We will devote the rest of this chapter to review the state of the art of closed-domain NL approaches, and to expose our objectives and methodology. The rest of this dissertations is organized as follows. Chapter II offers a preliminary study from the lowest level of knowledge available. Chapter III is dedicated to those systems that operate with unstructured knowledge but have meta-knowledge at their disposal. Collaborative and incremental systems will be discussed in chapter IV. Chapter V focuses on learning by analogy from the highest knowledge levels. Finally, main conclusions and future works will be presented in chapter VI.

2 State of the Art

This section is to offer a review of main close-domain NL approaches in basis of the knowledge levels. We present here a general overview —a more detailed and specific review will be offered in subsequent chapters.

2.1 From the Lowest Knowledge Levels

Most of the current information scattered through the Internet do not present any known structure and there is not any sort of meta-information describing it. For this reason, unstructured knowledge-based NL approaches without any sort of modelling represent a challenging objective. The main advantage is that these kind of systems do not impose any specific restriction on the data, nor require any additional resource either. As a counterpart, some methods require a suited corpus of labelled documents to be used for inferring certain criteria, depending on the case. This inference is usually called *supervised learning.* Although labelling documents is undeniably less costly than implementing an specific system, the truth is that this is a tedious task that, to cap it all, restricts the applicability of the method only to stable domains.

Since usually there is no *a priori* information about the knowledge at hand, this kind of algorithms are independent of the data. As a result, there is a large quantity of domain-independent algorithms that are easily adaptable to different problems. Most of these methods, including Artificial Neural Networks, Support Vector Machines, Evolutionary computation, Clustering methods, Decision trees, or Bayesian Networks, come from the Machine Learning (ML) community [Mit97] and have been broadly applied to different NL-related problems, such as:

Text Classification: concerns with how to assign zero, one, or more labels to a given document. Potential labels are known in advance and represent categories of semantically related documents [Seb02, Seb05].

Text Clustering: is aimed for discovering natural groupings of documents relating to a coherent subject matter [LLCM03, AZ12].

Text Summarization: is the problem of reducing a document through an automatic process aimed for creating a summary or abstract gathering most important aspects in the text [NM01, DM07].

Text Tetrieval: a branch of Information Retrieval (IR) that consists of retrieving the text documents or text fragments that best match with a given user query [SB88, Kor97].

Ordinal Text Classification: consists of automatically determining the implied rating of a document on a given ordered sequence of ranks or labels [MWL^{+}07, BES10].

Sentiment Analysis: is the study of opinions, sentiments, and emotions expressed in texts, from a computational point of view. This usually entails the analysis of polarity, subjectivity, and strengths of opinions. We will focus on this problem in chapter III, albeit in a different manner. As will be seen, we will approach the problem from a lexicon-based perspective [PL08, CZ10].

Question Classification: concerns with automatically determining the most probable category for a given question with respect to its expected answer type [Her01]

Representing text documents through a feature vectorial space is a common practice in ML methods. These features are usually extracted from terms appearing in the documents in the so-called Feature Extraction subtask. In any case, the dimension of the resulting feature vector space is likely to be of the order of tons, or even tens of tons. Although current computers along with actual learners devices do actually allow us operate under this conditions [Joa02, Joa06], it was observed that reducing the feature space leads to better performance [YP97]. We will dedicate part of this dissertation to this task, known as Feature Selection (chapter II).

2.2 Adding Linguistic Information

ML approaches could be considered to be "blind" in the sense they operate without any sort of NL-specific support. Although there is no further knowledge resource apart from a labelled corpus of documents, there are techniques that also incorporate linguistic information in the feature space. For example [ÖG10] is a text classification approach that also consider lexical dependency patterns in the feature space. Also, in the field of question classification it was proposed to add syntactic information [ZL03], and semantic information [BM07, LR06] to the feature space.

Apart from syntactic parsings and morphological information provided by the Part of Speech Tagging (PoST) methods, the principal strategy to add domain-specific linguistic information to a problem consists of using dictionaries, thesauri, or domain lexicons. A dictionary is a plain list of terms and definitions along with a list of synonyms and antonyms. Thesauri collect terms defining domain-concepts along with the hierarchical and semantic relations among them, such as WordNet [MBF$^+$90] or Wikipedia[1]). Finally, a domain lexicon consists of a set of terms with their lexical entries and morphological patterns [Lie80].

These kind of resources are obviously a valuable aid for any NL approach [MSC11]. Even if there are public general-purpose lexicons and dictionaries [SKA68, HL04a, ES06b, WWH05], meaning of words may vary depending on the domain. For this reason, there is usually the case that a suited linguistic resource, specifically tuned for the domain at hand is required. In this regard, several approaches aimed for identifying new concepts and expressions [LMS06, NPI11, BV04], adapting the term representations to new scenarios [TW11a], or tuning previously existing domains to new ones [QLBC09, ALSZ06, ALM$^+$03] have been proposed. According to [Liu12], main approaches to create lexicons could be classified as:

Manual approaches: entail a sizeable amount of work, but allow the system to be exhaustively monitored. These techniques are usually combined with automatic or semi-automatic approaches to expand a initial controlled vocabulary.

Dictionary-based approaches: these techniques take advantage of common features of standard dictionaries, such as antonymous or synonymous lists. Usually, a few set of seed words is firstly collected and later expanded exploiting semantic relations through WordNet [MBF$^+$90]. Most relevant dictionary-based methods include [Tur02a, ES06a, KH06, VBGHM10].

Corpus-based approaches: the main motivation of these techniques is to adapt a general-purpose lexicon to a particular domain, bearing in mind that the sentiment of a given word may vary across domains. Some examples of corpus-based approaches could be found in [HM97, WWH05, KN06]

[1] http://en.wikipedia.org/

2.3 Adding Semantic Information

Different authors are constantly updating the concept of domain lexicon to face more sophisticated problems. This phenomenon is clearly reflected in the Feature-based Sentiment Analysis problem [PL08, Liu12]. This branch of Sentiment Analysis aims to separately analyse the opinions on the main topics involved in a text document. This has been broadly addressed for product review [HL04a, GSS07] and product comparison [PE05]. For example, given the user comment "This camera is great, but I don't really like its battery performance" one could expect a broke down analysis like ⟨Camera,Positive⟩ y ⟨Camera.Battery, Negative⟩. To this goal, different hierarchical structures, including PART-OF [JL06] or IS-A [MRCZ12b] relations with features [KH04], have been proposed. These methods will be later discussed in detail in chapter III.

As structured lexicons involve more sophisticated mechanisms to represent its concepts and relations, they get conceptually closer to *domain ontologies*. Ontologies emerge as the natural evolution of different representation mechanism such as semantic networks, frames, Entity-Relationship diagrams, or Object-Oriented models. Ontologies are useful to represent the variety of digital resources available in the so-called Semantic Web. It is usually defined as "the conceptualization of the knowledge related to a *part* of the world" that collects the objects, features, relations, and entities of a given domain of knowledge through a shared vocabulary. We will go back to this concept in chapter V, section 1.3.

The use of ontologies to support the NL processing has been broadly applied from a long variety of disciplines and problems in the field of NL technologies. Such was the case that trying to fairly reflect here a representative number of related contributions could be misleading. We will rather discuss most related approaches to our aim and redirect the interested reader to [Bat97]. More specifically, we focus here in the use of ontologies to interpret NL questions.

According to [AS05] one of the main methods to interpret NL in closed domains could be classified as Natural Language Processing (NLP) approaches [ID10]. NLP approaches try to reach a formal representation of NL in order to fully interpret language. In doing this, NLP techniques do usually take advantage of domain ontologies. Main challenges NLP deals with include automatic translation, co-reference resolution, named entities recognition, automatic language generation, or grammatical parsing. Although NLP techniques have been applied to a large variety of problems such as FAQ retrieval [WTRM08, MHG+02, WMC09, TR08] —discussed below in chapter IV—, maybe the most representative example of the use of ontologies in NLP could be found in Virtual Assistants, Virtual Simulated Patients, and Natural Language Interfaces.

Virtual Assistants: A Virtual Assistant (VA) is a web-based agent that answers NL queries on a web-domain [ELC12]. Thus, users are released from manually searching the requested information, and being aware of how information is organized in the web. Additionally, VAs could incorporate *information recommendation systems*, and usually attempt to simulate human behaviours and reactions [ELC09].

Virtual Simulated Patients: are computational agents used in health care education for diagnostic training [LEC08, SPG09]. These systems try to simulate how a human patient would react to medical questions, potentially controversial.

Natural Language Interfaces: (NLIs) [ART95, PRGBAL+13] allow non-technical people to access information stored in knowledge-bases in a natural manner. NLI technologies are thus meant to override the complexities formal query languages such as SQL, SPARQL or XPath impose to information access.

As in the case of domain lexicons, the creation of a suited ontology is a tedious process that requires a deep understanding on the domain elements and remains usually restricted to Knowledge Engineers. Notwithstanding, some techniques to automatically obtain domain ontologies from texts have been proposed [SB04]. To access a detailed review of main related approaches [BCM05, MS01] could be consulted. The creation or adaptation of the ontology is just one of the various subtasks the expert should undertake to customize the system. Other tasks in this regard include the definition of a suited lexicalization [MSC11, BCM$^+$11], the mapping between domain elements and linguistic expressions [GAMP87], or the improvement of the *habitability* [Wat68, OB96, OMBC06] of the system —how well the system interprets the linguistic variety users could employ while interacting with it.

It is difficult, even for an expert in NL technologies, to predict the variety of linguistic cases the system will deal with during its life time. This problem, whereby the expert is in the middle of user's expectations and the domain extension, is usually known as "the bridge the gap" problem [Hal06, WXZY07]. An strategy to avoid it consists of delimiting the possible linguistic constructions an user could pose to the system through structured or menu-based input methods [BKK05, TPT05]. An strategy to deal with it consists of designing interaction protocols to allow the expert in charge test and debug the system. Some related approaches engages the expert with dialogue to request information [GAMP87, CHH07] or clarify ambiguities [DAC10, DAC12]. A more extensive study on human-computer interaction could be found in [HLP97]. We will devote chapter V to investigate new customization techniques based on interaction and learning.

2.4 Adding Syntactic Information

From an analytical point of view, we could say that both lexicons and ontologies are meant to add morphological and semantic information to the problem at hand. We will conclude our route through the different knowledge levels discussing a different manner to incorporate syntactical information to the problem.

We could consider that there are two main ways to incorporate syntactical information in the domain: the *implicit* and the *explicit* manners.

2.4.1 Implicit Syntax: Structure

The *implicit* manner concerns with the structure of the data stored. In this case, the structure is not necessarily related to the syntactic analysis from a linguistic point of view, but rather to the inner data structure. This approach is not incompatible with above mentioned methods —indeed, we could find examples combining them all in Natural Language Interfaces [WXZY07, CHH07] or Question Answering [Yan09b]. However, it is often the case that operating without any further knowledge model allow to design truly incremental systems, and this is arguably the most promising way to design collaborative systems.

Collaborative systems evolved from the concept of Web 2.0 as a tool where users are no longer mere consumers of information, but also potential producers of information. In this way, information generated by a given user could be accessed by others. These kind of systems is of particular interest in fields like e-learning [MRPVR07], where *Virtual Learning Environments* (VLE) or *Learning Management Systems* (LMS) allow both teachers and students improve their learning experience

[AHBLGS$^+$12, DSS$^+$02] by means of applications such as *WebCT*[2], *Blackboard*[3] or *Moodle*[4].

NL represents an intuitive and efficient manner to access information stored in collaborative systems such as *Stackoverflow*[5] or *Yahoo! Answers*[6]. These systems are aware of the structure of their knowledge, that is, information is stored in question/answers pairs, possibly grouped in categories. We will deal with this problem, usually known as FAQ retrieval, in chapter IV offering new scalable retrieval algorithms and methods to improve the domain knowledge. In contrast to other methods exploiting the meta-knowledge level [LFQL11, LLL10, HSCD09], we will face the problem from a mere structural point of view along the lines of [BCC$^+$00, KS08a, KS06]. Because this approach does not count with any meta-knowledge model, the systems continuously evolve with the simple addition of questions/answers pairs.

2.4.2 Explicit Syntax: Patterns

Finally, the *explicit* syntactic mechanism consists of the design of linguistic patterns. These patterns, usually known as *templates*, could be used both for interpreting sentences and extracting answers. The most extended mechanism to represent templates are the *regular expressions*, a formal mechanism proposed by Stephen Cole Kleene in 1950 that has been broadly applied to different NL approaches.

Although template-based methods are often applied to open-domain problems such as Question Answering [RH02, ZL02, SGS06, CKC07] or Question Classification [Her01, HW03], they also play an important role in closed-domain problems. The Sneiders' template-based systems are arguably the most representative approaches in this regard, including Email classification [Sne10], Database querying [Sne02] and FAQ retrieval [Sne09].

Templates efficiently embody experts' knowledge on the domain and were proven useful for different tasks [DSS11, Sou01]. However, designing templates is a tedious task that is usually carried out manually. Main efforts for automatically obtaining templates were driven in the field of regular expressions inference. For example [Fer09, BNST06, BGNV10] propose methods for automatically inferring the data description DTDs and XML schemas, and [RH02, ZL02] present methods for obtaining the so-called *surface text patterns* to extracts answers in open-domain QA systems. We will propose specific methods to create and refine regular expressions to recognize NL questions in chapter IV.

[2] `www.WebCT.com`

[3] `www.blackboard.com`

[4] `www.moodle.org`

[5] `http://stackoverflow.com/`

[6] `http://answers.yahoo.com/`

3 Objectives

The main goal of this Thesis is to offer innovative solutions to current open-problems of social interest in the field of closed-domain NL approaches, from a merely practical point of view. With this, we intend to reinforce the applicability nuance of this work. As a second goal, we plan to make our study extensible to a broader class of related approaches. To this end, we fix the following transversal objectives:

- To offer an analysis of closed-domain NL approaches, attending to the different levels of knowledge and meta-knowledge. Our aim is to identify the main problems that arise in each case in order to propose concrete solutions in form of applications.
- To investigate techniques that efficiently exploit the different knowledge-levels available.
- Often, customizing or adapting a system entails a great deal of effort. Thus, our aim is to obtain portable strategies able to take advantage of reusable information across domains.
- Implementing and developing rich applications that should be put to test in real environments.
- To propose valuable contributions to the state of the art through scientific publications.

More specifically, we pretend to fulfil with the following partial goals:

- Delimiting the context of term-relevancy is a common concern to many NL approaches. In this regard we plan to investigate the influence of positive correlation in the identification of the most relevant terms to a given domain. This will bring us the necessary background to better approach the upper knowledge levels.
- Since discovering main entities involved in a text is a fundamental part of many NL approaches we plan to propose new solutions to the focus detection problem by means of a previous context analysis. More specifically, we plan to propose methods for ambiguity, ellipsis, and anaphora resolution.
- To investigate methods for reducing the costs associated to the creation, maintenance, and refinement of these systems. That is, systems should be scalable and robust to potential changes in the domain.
- Data stored in collaborative systems should grow according to the users' information needs. Our aim is to propose usage mining techniques providing experts with the necessary information to monitor the system and to improve it, if needed.
- To allow experts revise and refine knowledge resources in critical systems. That is, mechanisms must be interpretable.
- To propose interaction protocols aimed for reducing the customization costs of NL systems, and to facilitate portability across domains.

4 Methodology

This section is to present the methodology we have followed in this Thesis.

We will base our analysis on NL technologies in the different levels of information available. There are two sort of clearly differentiated resources in a NL application: the *knowledge*, and the *meta-knowledge*. On the one side, the knowledge (aka *knowledge base* (KB)) consists of all information explicitly stored in the domain, and could be structured —a Database, or a XML file— or unstructured —a corpus of documents or a set of comments. On the other side, the meta-knowledge represents additional information explaining the knowledge. For example, an ontology defining main entities and relations in the domain, or a lexicon collecting all the terminology in a domain. We will use interchangeably terms *meta-knowledge*, *meta-information*, or *knowledge modelling*, in this dissertation.

We propose an analysis of closed-domains NL systems in basis of these two knowledge-levels. Firstly, we will attend to the presence or absence of structure in the KB. Secondly, we will attend to the presence or absence of any meta-knowledge resource.

This methodology will allow us establish a clear categorization of different techniques in the field of NL technologies. To identify the main features and difficulties that arise in each case, we will take a representative problem from each scenario. This problem will be chosen attending to two main aspects: (i) the problem is a good representative of the category, and (ii) it is currently an open-problem of social interest. This problem will lead us investigate different approaches to efficiently exploit the domain knowledge. In each case, our goal is two-fold: (i) to obtain concrete solutions to the problem at hand, and (ii) to draw general conclusions that are extensible, to some extent, to the broader set of techniques under a similar knowledge configuration.

According to this, following representative problems have been chosen:

- Text Classification or Text Categorization (TC) is a representative problem where no inner or known structure is present in documents, and there is not any meta-knowledge resource available, far from a set of semantic labels. Loosely speaking, this problem consists of attaching zero, one, or various labels to each document. To this aim, a classifier is built based on the observation on a set of previously labelled documents according to expert criteria.

- Sentiment Analysis or Opinion Mining is aimed for automatically generating meaningful summaries of opinion-bearing texts documents. Usually, opinions are presented in free-text format (that is, as unstructured documents) and could even be faced from a Text Classification perspective, where different opinions (positive or negative) represent categories. Given the imminent interest of these kind of applications, a number of variants have been investigated, one of which is devoted to analyse not only the polarity, but also the target of the opinions. To achieve this goal, a domain-lexicon is usually employed. We thus select these techniques, aka *Lexicon-based Sentiment Analysis* as a representative problem with unstructured knowledge and a meta-knowledge resource available.

- There are however systems in which no meta-information is available, but the knowledge is structured. These kind of systems are important in collaborative environments, like e-Learning. FAQ retrieval is a clear example within these methods. A FAQ (Frequently Asked Questions) is a paired list of questions/answers related to a subject that could, in addition, be organized in categories. The problem could be stated as retrieving the most related questions/answer pairs to an user query expressed in NL.

- The last combination corresponds to structure knowledge along with a suited meta-knowledge. Arguably, one of the most representative problems under this setting are Natural Language Interfaces (NLIs). Some of these systems try to translate NL questions to formal queries to a given Database (the structured knowledge). This process is usually supported by a semantic model, usually represented through a domain ontology (the meta-knowledge).

Finally, Figure i.1 summarize the selected problems in basis of the different combinations of knowledge levels.

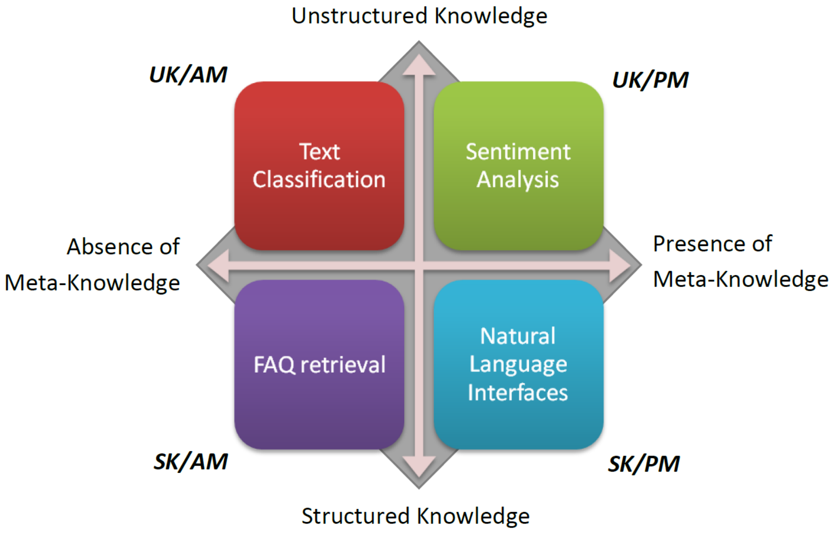


Figure i.1: Selected problems according to the different levels of knowledge

One of our objectives (section 3) consists of exploring the different levels of knowledge in favour of the application. In this respect, it could be remarked that the UK/AM combination offers the lowest level of information. For this reason, we have chosen this combination as an starting point, offering a preliminary study. This study will represent the foundations upon which the subsequent research is based. As could be expected, this route ends up with an study on SK/PM, the highest knowledge-level achievable.

The following guidelines have been strictly followed in this work for each case study:

- Discussion about the main particularities, including advantages and drawbacks, of main methods under the case study.
- Justification of the selected representative problem and formal definition.
- Explanation of formal mechanisms involved in our proposal. Notation and description.
- Review of the state-of-the-art.
- Explanation of the method proposal in detail. Implementation and experimental validation, including a suited comparison against most important related methods in the literature.
- Conclusions drawing and proposal for future research.

More specifically, following steps will be carried out:

Feature Selection: to investigate term-relevancy to a given domain, we will test different filtering methods, selection policies, classifiers, evaluation metrics, and datasets in order to observe the influence of positive correlation in the feature selection task.

Sentiment Analysis: we will delimit in advance the context of the set of user comments, in order to identify main discussion topics. To this purpose, we will attempt the ambiguity, ellipsis and anaphora resolution by exploiting the semantic and morphological information in the domain lexicon.

FAQ retrieval: we will define the minimality and differentiability criteria to infer a set of regular expressions that will be later used to retrieve the most relevant information to an user query. We will later define usage mining techniques to evaluate the quality of the FAQ content by exploiting users' reactions while interacting with the system. Finally, we will propose a method to generate interpretable templates from more elaborated regular expressions. These templates will allow the expert in charge to manually revise and refine the system performance, if needed.

Natural Language Interfaces: we will use free-context formal grammars to reach NL interpretations. We will design a grammar inference method based on reasoning by analogy. Finally, we will exploit hypotheses and deductions mechanisms to allow the system deal with unseen expressions.

Chapter II

Text Classification: Unstructured Knowledge, Absence of Meta-Knowledge

1 Introduction

This short chapter is to offer a preliminary study on closed-domain NL approaches from the lowest knowledge-level. That is, documents present no inner or known structure and there is not any meta-model available. Thus, it could be said that these approaches are blind in the sense that no prior knowledge is available. However, methods under this configuration are extremely promising given the fact that they do not impose any sort of requirement for users, nor modelling for experts in charge of the customization either. It is indeed the case that most of the content in the Internet falls under this assumption, whereby documents contain unstructured free-text, and there is not any meaningful explanation on its content. Main problems under the UK/AM configuration include Text Classification [Seb02, Seb05], Text Retrieval [SB88, Kor97], and Text Clustering [LLCM03, AZ12], to name just a few (see chapter I section 2 to access a broader discussion on UK/AM problems).

In this part of the dissertation, we are interested in investigating what are the characteristics that make a term become relevant to a given domain. In doing this, we plan to take the UK/AM configuration as a starting point, in an attempt to discover these characteristics when no prior knowledge resource is available. Shedding some light to this issue will be a valuable help to our subsequent investigation —as reflected in the following chapters. This preliminary study is thus motivated by obtaining the necessary background to allow us develop rich NL applications exploiting the different levels of knowledge and meta-knowledge according to the proposed methodology (chapter I section 4).

Above mentioned motivations suggest a focused study on the so-called *Feature Selection* (FS) sub-task. Albeit FS could be considered to be a fundamental part for many problems in the UK/AM class, is in the Text Classification field where most efforts have been devoted. Thus, we will here consider the *Feature Selection for Text Classification* problem —more formally introduced in section 3.1— as the most promising field where to pursue our goal.

The role of TC at present is paramount. Given the sheer quantity of text documents that are constantly appearing and the continuous needs for information of the nowadays society, how to properly label texts in basis of its content became an urgent challenge of the utmost importance. In this regard, TC technologies are witnessing an increasing interest that is deservedly supported by the sheer quantity of potential applications, including automatic document organization [GBMAHS02], text filtering and spam filtering [ÇG08], word sense disambiguation [IV98], automatic authorship attribution [SFK00], hierarchical categorization of web pages [EFS08], or language identification [TG12]. A broader discussion on TC applications could be found in [Seb05].

Early works on TC relied on a set of hand-coded linguistic rules describing the classification behaviour of the system [HANS90]. For example, the rule *if* ((‘how’$\wedge$‘install’)$\vee$(‘install’$\wedge$‘program’)) *then* **Category**:=*Installing*, explicitly describes the combinations of words that may determine a decision on the class *Installing*. These rules are arguably easily interpretable, but to create them, however, a sizeable amount of effort was usually required for experts in Knowledge Engineering methods. Furthermore, linguistic interferences among classes, that are hardly perceivable by an expert, may occur —additional comments on linguistic rules will be later offered in chapter IV.2. If that was not enough, how to modify these rules to better fit the domain after changes becomes usually a cost-prohibitive task for unstable domains. How relevant terms could be automatically obtained is thus an important issue we will try to answer, at least partially, in this chapter.

Machine Learning (ML, see [Mit97]) for Text Classification emerged thus in the early 90’s as a promising field aimed for overcoming above mentioned problems. According to A. Samuel, ML is “the field of study that gives computers the ability to learn without being explicitly programmed”. In the case at hand, ML for TC could be regarded as a set of inductive processes to automatically build learners for a category c_i, based on the observation of a given dataset, to classify new unseen documents into c_i or $\overline{c_i}$ —that is, inside or outside the category (see [Seb02, Seb05]). This process is usually known as *supervised learning*, to reinforce the fact that training data were previously labelled by a human expert, that could be somehow regarded as a supervisor of the training task. As a result, the effort required to obtain a suited classifier is reduced to that of properly labelling a set of training data, which is in turn undeniably less costly than manually describing the decision rules.

Inductive processes do usually take the terms appearing in the corpus to conform the input features (aka *Feature Extraction* task) to build the learners. As a result, the extent of the input space tends to be of the order of tons. Although training times to build the learners are inevitably affected by this phenomenon, the truth is that current learner devices along with the nowadays computational power have overcome the prohibitive computational restrictions this fact may impose in real problems [Joa02, Joa06]. However, it is not the training times the only motivation for Feature Selection. It was observed [YP97] that the performance of a TC approach could be improved by applying an aggressive reduction in the term space. This thus suggests FS could benefit TC methods insofar as some noisy interferences could be avoided, while also the overfitting[1] problem tends to be diminished.

The rest of this chapter is organized as follows. Section 2 reviews the main approaches to Feature Selection for Text Categorization. In section 3.1 the problem is formally defined. We present main methods to select features in section 3.2. Classifier learning techniques and evaluation metrics are discussed in section 3.3. We present our empirical study in section 4. Finally, some conclusions and future work are offered in section 5.

[1] Term *overfit* refers to the bias that could result from the intensive training of a given dataset, that may incur in lower test performances. That is, the system behaves well with the training set, but poorly with unseen data.

2 Related Work

Term space reduction (TSR) methods attempt to select the reduced set of features that yields the highest effectiveness when classifying. To this purpose, *wrapper* and *filtering* approaches have been proposed. On the one side, wrapper approaches attempt to optimise the classification performance (in the *training* stage) by iteratively adding or removing a feature from an initial term set. To that end, the same learner device is used for both FS and TC tasks [KJ97]. Although the resulting reduced sets are thus tuned for the learner device in charge of classification, the cost associated to these techniques may become an insurmountable obstacle for most TC applications. On the other side, filtering approaches compute a mathematical function meant to measure the informative contribution of each isolated feature to the TC task. After that, only top-n features are taken. In the rest of this chapter we will only attend to filtering methods. The interested reader could delve deeper into the subject in [TG08, For03].

Early work on FS for TC was mainly motivated by the high computational cost that building a learner over a huge set of features could entail. However, because of the continuous advances in hardware along with the current improvements in automatic learner devices, this restriction does no longer prevail at present [Joa98, Joa06]. In any case, not only training times of learner devices could be benefited from this reduction in the term space, but also the classification performance [YP97] due to the fact that the *overfitting* problem tends to be diminished.

The problem of FS has been considered from a wide variety of fields, including *Ordinal Text Classification* [BES10, MWL$^+$07], Text Clustering [LLCM03, TT10], Sentiment Analysis [ACS08], and Text Summarization [KPK01]. In the context of TC, Yang and Pedersen reported that filtering methods based on entropy —*Information Gain*— and statistical independence —*Chi-square*— (see section 3.2.1) obtained the best performance in a number of datasets [YP97]. In a more recent study, Forman presented his Bi-Normal Separation filtering method [For03] that focuses on highly-imbalanced classes. In his study, Forman reported that Information Gain was preferable for optimising precision, while Bi-Normal Separation obtains better accuracy, F-measure, and recall scores, specially in high-skew classes.

Above mentioned TSR functions pay equal attention to both positive and negative correlations. That is, a frequent word that rarely occurs in a given class could be as informative for the classification task as another one that tends to exclusively appear in one category. Nonetheless, different TSR functions exploiting only the positive correlation have been proposed. Examples of these methods could be found in [NGL97, RS99, GSS00]. These principles will be discussed in more detail in this work.

In addition to TSR functions, how to finally select the reduced set of features before building the learner should be decided. In this regard, *Global* policies justify the same selection of features for all binary classifiers, while *Local* policies take a potentially different subset of features for each one [Seb02]. The *Round Robin* policy was proposed in [For04] as an attempt to overcome the so-called 'siren pitfall' problem. This problem states that strongly predictive features for minority classes could monopolise the selection process, while more difficult categories could thus be ignored.

3 Identifying Relevant Terms to a Category

3.1 Feature Selection for Text Classification: Problem Statement

According to the notation of [Seb02], the *Text Classification* (TC) problem may be formalized as the task of approximating the unknown *target function* $\Phi : \mathcal{D} \times \mathcal{C} \rightarrow \{-1, +1\}$ that indicates how documents ought to be classified, by means of a function $\widehat{\Phi} : \mathcal{D} \times \mathcal{C} \rightarrow \{-1, +1\}$, called *the classifier*, such that Φ and $\widehat{\Phi}$ coincide as much as possible in terms of any given *evaluation metric*. That is, TC is the task of trying to decide a positive or negative label for each pair $\langle d_j, c_i \rangle \in \mathcal{D} \times \mathcal{C}$ as well as an expert would do, where $\mathcal{D}$ denotes the domain of documents, and $\mathcal{C} = \{c_1, c_2, ..., c_{|\mathcal{C}|}\}$ is a set of predefined *categories*. Values $+1$ and -1 are used to indicate membership and non-membership of the document into the category, respectively. The problem is usually stated as a "multi-label" classification problem, that is, each document could belong to zero, one, or several categories (aka *overlapping categories*) at the same time. We will here consider the "flat" version of the problem, whereby no hierarchical relations among different labels are taken into account.

Text Classification is undeniably affected by the *curse of dimensionality* problem. That is, the number of *features* depends on the number of terms appearing in the corpus of documents which, in turn, is likely to be of the order of tons. In this respect, the so-called *Feature Selection* (FS) for Text Classification problem is aimed for reducing the dimensionality of the term space from $|\mathcal{T}|$ to $|\mathcal{S}|$, where $\mathcal{T}$ denotes the initial term space and $\mathcal{S} \subset \mathcal{T}$ denotes the reduced term space, satisfying $|\mathcal{S}| \ll |\mathcal{T}|$ —the so-called *ratio of reduction* $\xi = |\mathcal{S}|/|\mathcal{T}|$ determines the reduction level.

3.2 Reducing the Term Space: Filtering Approaches

This section is to briefly summarize main strategies in filtering approaches. Thus, this discussion is not pretended to be an exhaustive review. The interested reader is rather referred to the works [For03, Seb02]. The present section covers the following topics: main TSR functions (section 3.2.1), and most relevant FS policies (section 3.2.2)

3.2.1 Term Space Reduction Functions

As mentioned before, we are considering here the so-called filtering methods. As those methods operate with isolated features, they could be computed as a function of the *successes* and *failures* that the presence or absence of feature f_k at hand produces with respect to the expected true classification c_i. That is, TSR functions are usually computed as a function of the co-presence and co-absence of t_k with respect to c_i which could, in turn, be summarized in a four-cell contingency table (Table II.1).

$\widehat{\Phi_i^k}$	t_k	$\overline{t_k}$
c_i	TP	FN
$\overline{c_i}$	FP	TN

Table II.1: Classification Contingency Table

Where *TP* and *TN* stand for *true positives* and *true negatives* (successes), while *FP* and *FN* denote the *false positives* and *false positives* (failures), analogously. t_k is used to denote *presence* of feature k, while $\overline{t_k}$ indicates *absence* of the feature. Analogous notation goes for c_i and $\overline{c_i}$.

It is usually the case that TSR functions are expressed as non-linear functions computed on the contingency table in terms of probability. That is, the space of documents is considered to be an event space, and each prediction is considered to be an observation. Thus, the following probabilities could be taken into account (Table II.2) and may serve to facilitate the reading.

Single Probabilities	**Compound Probabilities**
$P(t) = \frac{TP+FP}{D}$	$P(t, c) = \frac{TP}{D}$
$P(\overline{t}) = \frac{TN+FN}{D}$	$P(\overline{t}, c) = \frac{FN}{D}$
$P(c) = \frac{FN+TP}{D}$	$P(\overline{t}, \overline{c}) = \frac{TN}{D}$
$P(\overline{c}) = \frac{TN+FP}{D}$	$P(t, \overline{c}) = \frac{FP}{D}$

Table II.2: Probabilities in the space of documents

Where D represents the total number of observations, according to equation II.1.

$$D = TP + TN + FP + FN \tag{II.1}$$

Further notation may be found in several related work. Following ones are among the most typical ones: *true positive rate* (*tpr* in Equation II.2), and *true negative rate* (*tnr* in Equation II.3).

$$tpr = \frac{TP}{TP + FN} = P(t \mid c) \tag{II.2}$$

$$tnr = \frac{TN}{FP + TN} = P(\overline{t} \mid \overline{c}) \tag{II.3}$$

Because binary classifiers are separately built for each category, the particular distribution of documents through the category at hand becomes a constant. Thus, one could express the 4-cell contingency table as a function of the *tpr* and *tnr* —the remaining values are directly determined by the constant distribution of documents in the category (Table II.3).

$\widehat{\Phi_i^k}$	t_k	$\overline{t_k}$
c_i	*tpr*	1-*tpr*
$\overline{c_i}$	1-*tnr*	*tnr*

Table II.3: Equivalent representation involving two variables.

This new representation will allow us plot[2] in a 3-Dimensional space both TSR functions and evaluation metrics, as proposed in other recent works [For03]. To this purpose we used *Sage*[3], a free open-source mathematics software. Finally, Figure ii.1 plots some of the TSR functions we are considering in this empirical study. What should be highlighted is that only GSS is asymmetric with respect to *tpr* and *tnr*. As will be seen, this TSR function attends only to positive correlation.

[2] Unfortunately, we could not plot Bi-Normal Separation in *Sage* because its formulation depends on infinite series.

[3] `http://www.sagemath.org/`

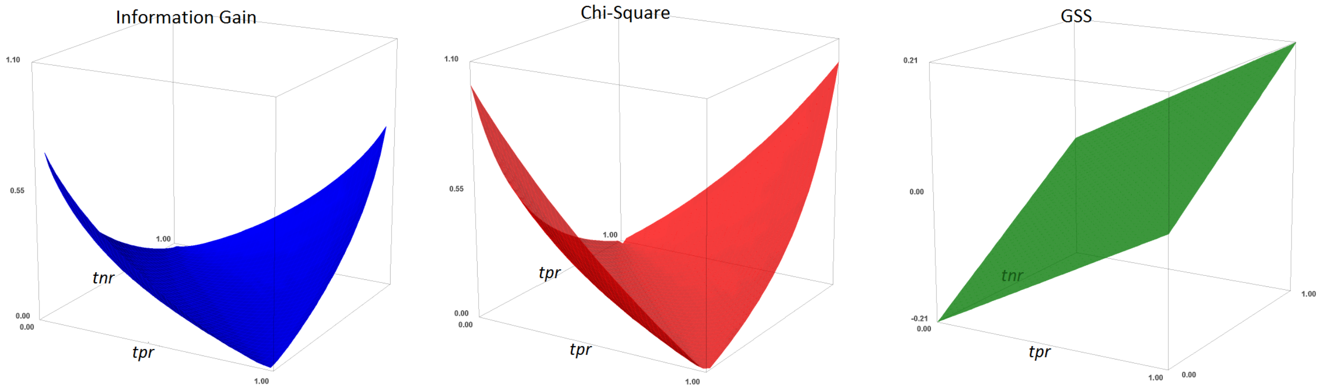


Figure ii.1: 3D Plot Representation of Main TSR Functions: Information Gain (left), Chi-Square (center), and GSS (right).

Information Gain The *Information Gain* metric comes from the field of Information Theory. This metric is based on the *Kullback-Leibler divergence*. Loosely speaking, Information Gain is defined in FS as the reduction in the entropy[4] of a given category c_i achieved after observing the feature f_k:

$$InfoGain(t_k, c_i) = H(c_i) - H(c_i|t_k) = H(t_k) - H(t_k|c_i) \tag{II.4}$$

That is usually rather expressed in terms of the single and compound probabilities as:

$$InfoGain(t_k, c_i) = \sum_{t\in\{t_k,\overline{t_k}\}} \sum_{c\in\{c_i,\overline{c_i}\}} P(t,c)\log_2 \frac{P(t,c)}{P(t)P(c)} \tag{II.5}$$

Figure ii.1 plots Information Gain as a function of the *tpr* and *tnr*.

Chi-Square *Chi-Square*, typically denoted as χ^2, is an statistical test aimed for measuring the divergence from the χ^2 distribution expected with one degree of freedom, assuming statistical independence between the given feature and the given class.

$$\chi^2(t_k, c_i) = \frac{|D|(P(t_k, c_i)\cdot P(\overline{t_k}, \overline{c_i}) - P(\overline{t_k}, c_i)\cdot P(t_k, \overline{c_i}))^2}{P(t_k)\cdot P(\overline{t_k})\cdot P(c_i)\cdot P(\overline{c_i})} \tag{II.6}$$

Figure ii.1 plots Chi-Square as a function of the *tpr* and *tnr*.

Bi-Normal Separation *Bi-Normal Separation* (BNS) was first proposed by George Forman [For03]. This TSR function is defined as:

$$|F^{-1}(tpr) - F^{-1}(fpr)| \tag{II.7}$$

Where F^{-1} is the *z-score* —the standard Normal distribution's inverse cumulative probability function (see Equation II.8).

[4] Note this formulation is equivalent to the so-called *Mutual Information*

$$F^{-1}(p) = \mu + \sigma\ \Phi^{-1}(p) = \mu + \sigma\sqrt{2}\ erf^{-1}(2p-1) \tag{II.8}$$

Being $\mu = 0$ and $\sigma = 1$. erf^-1 is the inverse error function defined by:

$$erf^{-1}(z) = \sum_{k=0}^{\infty} \frac{c_k}{2k+1} \left(\frac{\sqrt{\pi}}{2} \cdot z \right)^{2k+1} \tag{II.9}$$

$$c_k = \sum_{m=0}^{k-1} \frac{c_m \cdot c_{k-1-m}}{(m+1)(2m+1)} \tag{II.10}$$

GSS Loosely speaking, the intuition beyond GSS [GSS00] is that terms should be well-related to a category as long as they tend to appear in the documents labelled with that category, and tend to not appear in the rest. Furthermore, it was stated that the addition of a new document to the dataset may cause the appearance of negative correlations between some terms in that document with respect to other documents belonging to different categories. That suggests those correlations should not be taken into account while looking for the most representative features to a given category.

Thus, GSS metric relies only on positive correlation (Equation II.11), rather than on both positive and negative correlation —like Information Gain, Chi-Square, and Bi-Normal Separation do.

$$GSS(t_k, c_i) = P(t_k, c_i) \cdot P(\overline{t_k}, \overline{c_i}) - P(\overline{t_k}, c_i) \cdot P(t_k, \overline{c_i}) \tag{II.11}$$

Figure ii.1 plots GSS as a function of the *tpr* and *tnr*.

There are other methods in the literature that also rely only on positive correlation [NGL97, RS99]. However we decided to take GSS as a more representative metric, after a pilot study.

3.2.2 Feature Selection Policies

FS Policies determine the manner in which features ought to be selected in basis of a given TSR function. In this section, we briefly review the policies we will focus on.

Global-Max: This policy is arguably the simplest one. It takes iteratively the feature maximizing the score value for the TSR function f considering all categories at once. Thus, the score of a given feature t_k is computed globally as $f_{max}(t_k) = \max_{i=1}^{|\mathcal{C}|} f(f_k, c_i)$

Global-Sum: In this case, the global informativeness score of a given feature is computed additively as $f_{sum}(t_k) = \sum_{i=1}^{|\mathcal{C}|} f(f_k, c_i)$, where f is the TSR function at hand.

Local: This policy consists of taking the top-$|\mathcal{S}|$ best features for each binary classifier according to a given TSR function. That is, each binary classifier is trained by a different subset of features that is locally decided for that category. Thus, there are as many different subsets of features as different classes —exactly $|\mathcal{C}|$.

Round-Robin: This policy, motivated by the homonyms scheduling method, was proposed by Forman [For04] as an attempt to overcome a common problem in FS (the 'siren pitfall') whereby the presence of strongly predictive features in a given class tends to avoid the selection of necessary features to classify some classes. It consists of firstly ranking all features according to a given TSR function for each binary decision sub-task. After that, the best features for each class is picked in turn, until $|\mathcal{S}|$ features have been retrained.

3.3 Classifying Documents

This section is to offer a brief discussion on the classifier learning techniques we will use in this empirical study (section 3.3.1). After that, most popular performance evaluation metrics in the literature will be explained (section 3.3.2).

3.3.1 Learner Devices

A large quantity of different learner devices have been proposed in the literature. So was the case that citing here just a few would be misleading. Rather we will only present those classifiers we will use in our experimentation and redirect the interested reader to a vaster discussion available in [Mit97].

Among the various classifiers, we will attend here to *Support Vector Machines* (SVMs), *Naïve Bayes* (NB), and *AdaBoost*.

SVMs: Support Vector Machines were first proposed by Vapnik in [Vap95], and introduced in the field of TC by Joachims in [Joa98]. Loosely speaking, SVMs consider samples as point in space, and decide the linear classifiers that maximize the functional margin between the two categories. If a simple linear classifier does not succeed, then a *kernel function* maps the inputs to a higher dimensional space. Thus, the classifier is no longer linear, but an hyperplane.

More formally, given a training dataset $\mathcal{D} = \{(x_i, y_i), i = 1..m | x_i \in \mathcal{R}^p, y_i \in \{+1, -1\}\}$, the problem could be stated as an optimisation problem trying to find the hyperplane $\langle w \cdot x \rangle - b = 0$, where $\langle \cdot \rangle$ is the *dot product* and w the normal vector to the hyperplane, that minimizes the $||w||$ subject to $y_i(\langle w \cdot x_i \rangle - b) \geq 1$. This problem could be reformulated as a *quadratic programming optimisation* algorithm taking advantage of the *Lagrange multipliers*. Finally, this formulation was extended in [CV95] with the concept of *Soft Margin*, whereby the degree of misclassification was introduced in the problem.

Popular public implementations of SVMs include *SVMlib*[5] for *Java*, and SVM^{light}[6] in C.

SVM^{perf}: Is a SVM-based method for optimizing multivariate potentially non-linear performance functions that could be directly computed from a contingency table [Joa05]. This algorithm allows to optimise binary classifiers directly to measures like F^1, *Precision*, or *Recall*, in polynomial time. A publicly available implementation of SVM^{perf} could be found in[7]

Naïve Bayes: this probabilistic model comes from the machine learning community [Mit97] and was later applied to the TC task [YL99, MN98]. It is based on the assumption that the presence or absence of any given feature occurs independently of the presence or absence

[5] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[6] http://svmlight.joachims.org/

[7] http://www.cs.cornell.edu/people/tj/svm_light/svm_perf.html

of each other features in the category. That is, each feature contribute independently to the probability of the category. This assumption allows NB classifiers to be trained very efficiently.

More formally, the probability of a given category c_i in a conditional model could be expressed in function of all features as $P(c_i|t_1, t_2, ..., t_{|\mathcal{T}|})$. By applying the Bayes' theorem, the problem could be reformulated as determining the join probability $P(c_i, t_1, t_2, ..., t_{|\mathcal{T}|})$. The *chain rule* along with the *independence* assumption lead to $P(c_i|t_1, t_2, ..., t_{|\mathcal{T}|}) \propto P(c_i) \prod_{k=1}^{|\mathcal{T}|} P(t_k|c_i)$.

AdaBoost: from *Adaptive Boosting* [FS95, FS96], is based on the notion of reinforcing the weight for previously misclassified instances. Those weights, initially equal for all examples, are used to build the subsequent classifier in an iterative process. Those classifiers are usually called *weak* in the sense that they are only required to perform differently from a random classifier. If so, the global classification performance tends to improve in the long time.

More formally, given a training dataset $\mathcal{D} = \{(x_i, y_i), i = 1..m \mid x_i \in \mathcal{R}^p, y_i \in \{+1, -1\}\}$ and a number of iterations T, weights are initialized as $D_1(i) = 1/m$ for the m samples. For each iteration t, the weak classifier $h_t \in \mathcal{H}$, i.e. the classifier that differs most with respect to a random classifier —with 0.5 expected error rate—, is taken. That is, $h_t = argmax_{h_t \in \mathcal{H}} |0.5 - \epsilon_t|$, where $\epsilon_t = \sum_{i=1}^{m} D_t(i) I(y_i \neq h_t(x_i))$ is the error rate, $D_i(i)$ is the weight of the instance i in the iteration t, and I is the indicator function. After that, weights are updated by reinforcing misclassified instances. Finally, the classifier is build as a linear combination of the T previous weak learners, as $H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$, being typically $\alpha_t = \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t}$.

3.3.2 Evaluation Metrics

Unfortunately, how to properly evaluate the performance of a NL-based system is not clear. Most of the current metrics come from the Information Retrieval field, and it is not completely accepted that those metrics reliably reflect the system's performance [Pow07]. However, since it is not among the goals of this preliminary study to offer new evaluation metrics for the Text Classification task, we opted for employing the most popular current ones. In this regard, we will briefly review the main metrics that served for evaluating an extensive number of works in the field.

Once the classifier has been trained, it is used as a predictor for its evaluation. Memberships of the test documents d_j to a given category $c_i \in \mathcal{C}$ is approximated by the binary classifier $\widehat{\Phi_{c_i}}$ that should answer $+1$ or -1, accordingly. Thus, most evaluation metrics could be computed on the following contingency table (Table II.4).

	$\widehat{\Phi_{c_i}} = +1$	$\widehat{\Phi_{c_i}} = -1$
$\Phi_{c_i} = +1$	TP	FN
$\Phi_{c_i} = -1$	FP	TN

Table II.4: Classification Contingency Table

Precision (also known as *Confidence* in Data Mining) denotes the proportion of predicted positive cases that are correctly real positives (Equation II.12).

$$Precision = \frac{TP}{TP + FP} \tag{II.12}$$

Recall (also known as *Sensitivity* in Psychology) is the proportion of real positive cases that are correctly predicted positive. This metric measures the coverage of the classifier in terms of real positives (Equation II.13).

$$Recall = \frac{TP}{TP + FN} \tag{II.13}$$

Accuracy is usually expressed as the proportion of successes, both positive and negatives, to the total number of cases evaluated (Equation II.14).

$$Acc = \frac{TP + TN}{D} \tag{II.14}$$

Categories in TC are usually imbalanced or extremely imbalanced. The interest class does usually contain much fewer documents than the rest of classes together. This distribution skew causes the Accuracy to be affected by the so-called *Accuracy Paradox*, whereby predictors achieving a given accuracy could be preferable than others achieving higher accuracy scores. The reason why, is that the TN counter is usually much higher than the TP, which in turn means a *trivial rejector*[8] is a high-accurate predictor. Thus, other metrics depending on precision and recall are usually preferred. In this regard, $F - score$ or F_1 is defined as the harmonic mean of precision and recall (Equation II.15).

$$F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{II.15}$$

Depending on the application, it could be the case that precision or recall should lead the criterion. For example, it was shown that recall seems to be more relevant in a translation context [FM07]. In these cases, the more general version of the F metric is rather used. As defined by Van Rijsbergen, the F_β measures the effectiveness of a classifier to an user who pays β times more importance to recall than to precision (Equation II.16).

$$F_\beta = (1 + \beta^2) \cdot \frac{Precision \cdot Recall}{\beta^2 \cdot Precision + Recall} \tag{II.16}$$

Since each binary classifier is trained for a given category, metrics summarizing the global classification performance to all categories at once are needed. In this regard, *micro-averages* (μ) and *macro-averages* (M) have been proposed. *Micro-averaging* is computed as the summation of all individual counters in the contingency table of each category. Instead, *Macro-averaging* is evaluated as the average of the individual score of the measure at hand obtained for each category (Table II.5).

Finally, we used *Sage* to plot in a 3-dimensional space above mentioned evaluation metrics (Figure ii.2).

[8] A naive predictor that always returns -1.

$$Precision^{\mu} = \frac{\sum_{i=1}^{|\mathcal{C}|} TP_i}{\sum_{i=1}^{|\mathcal{C}|} (TP_i + FP_i)} \qquad Recall^{\mu} = \frac{\sum_{i=1}^{|\mathcal{C}|} TP_i}{\sum_{i=1}^{|\mathcal{C}|} (TP_i + FN_i)}$$

$$Precision^{M} = \frac{\sum_{i=1}^{|\mathcal{C}|} Precision_i}{|\mathcal{C}|} \qquad Recall^{M} = \frac{\sum_{i=1}^{|\mathcal{C}|} Recall_i}{|\mathcal{C}|}$$

Table II.5: Macro & micro measures

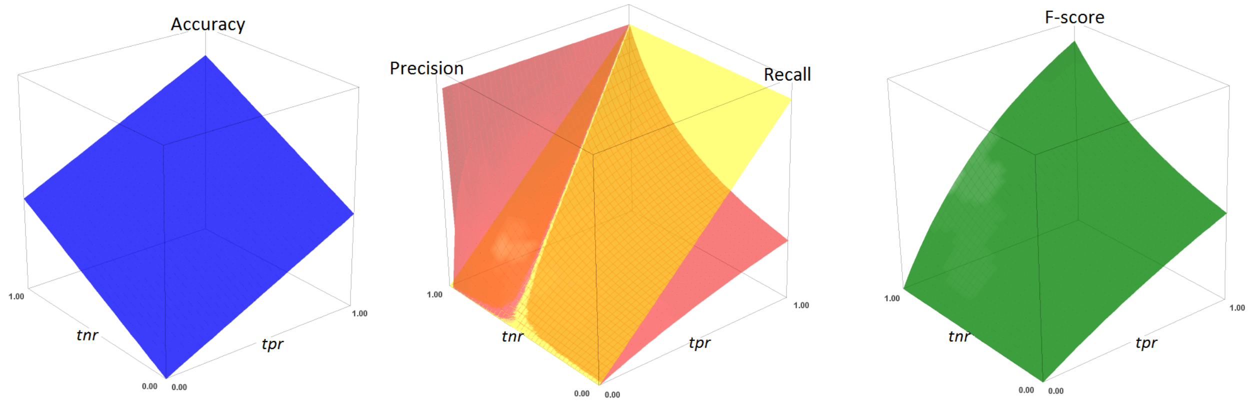


Figure ii.2: 3D Plot Representation of Main Evaluation Metrics: Accuracy (left), Precision and Recall (center), and F-score (right).

4 Empirical Study

4.1 Datasets

As dataset we have used the Reuters-21578, and OHSUMED corpora.

Reuters-21578 is a dataset containing 12.902 news stories. According to the "ModApté" split, the dataset is split into 9.603 training documents and 3.299 testing documents. There are 118 different labels. We have however paid attention only to the 115 categories presenting at least one positive training example. Most categories are high-imbalanced, ranging from $P(c) = 0.3$ to $P(c) = 0.0001$. Indeed, 90% of the categories satisfy $P(c) < 0.01$. This dataset is publicly available[9] and is probably the most widely used benchmark in TC.

The OHSUMED test collection [HBLH94] is a medical corpora containing 348.556 papers published from 1987 to 1991. Although papers do usually present certain structure —title, abstract, author, etc.— this structure is not used as additional information in this study. We have setup our experiments according to [LSCP96]. Concretely, only 233.445 entries with abstract and MeSH indexing terms have been considered. These entries were split into 193.229 training documents taken from 1987 to 1990 and 50.216 testing documents corresponding to 1991. Category labels were directly taken from the MeSH index terms. In particular, only categories containing at least one positive training example in the *Heart Disease* branch of the MeSH index terms were considered.

Since those datasets are publicly available and have been extensively used in other related

[9] http://www.daviddlewis.com/resources/testcollections/~reuters21578/

evaluations, our results become replicable.

4.2 Implementation

We have preprocessed all documents involved in our experiments by removing stop words and punctuation marks. All letters have been converted to lowercase and numbers have been removed. The Porter's stemmer was used to perform the word stemming. Finally, we have applied the broadly used $tf \cdot idf$ word weight criterion to represent the document terms.

JATECS, a JAva library for TExt Categorization[10] developed by the research group of F. Sebastiani and A. Esuli, was used to carry out our experimentation. We choose the standard parameters for each builder except for SVM^{perf}, where after a pilot experiment we decided to use $C = 1$ (instead of $C = 0.01$) as a compromise between training error and margin.

4.3 Experimental Results

In order to shed some light on what are the main characteristics that make a term become relevant to a given knowledge domain, we have designed the following experiments. In a preliminary study, we realised that the most interesting region where to perform FS ranged from $\epsilon = 0$ to $\epsilon = 0.2$. Concretely, we will consider the following fixed set of ratios:

$$\xi = [0.001, 0.005, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.11, 0.12, 0.15, 0.20] \quad \text{(II.17)}$$

Some works have already pointed out SVMs as a consistent top-performing learner for the TC task [YL99, DPHS98] while comparing it against other classifiers including $k-$Nearest Neighbour (k-NN), Neural Networks, or Linear Least-squares Fit. In [TA03] however, Tsamardinos et al. showed that feature selection is affected by both the classifier and the performance evaluation metrics being considered. Moreover, in [SS00, SSS98] it was reported promising performances for AdaBoost in different TC tasks. Furthermore, boosting techniques have been successfully combined with SVMs in [Joa98].

Above mentioned reasons motivated a preliminary comparative of different classifiers including AdaBoost and SVMs. Since this experiment will be evaluated in both macro and micro F_1, we will also consider SVM^{perf} optimising this metric. Finally, we report results for Naïve Bayes learner as an example of a very efficient and classical classifier. Figure ii.3 shows the effect of feature selection by using the Information Gain TSR function along with the Round Robin policy in Reuters dataset.

According to our expectations, SVM and AdaBoost are among the best classifiers. However, it was somehow surprising the fact that SVM^{perf} obtained such worse results while evaluated in F_1 —a metric it was supposed to optimise. Undeniably, the worst performing classifier was Naïve Bayes. From now on, we will consider only SVM and AdaBoost as the binary classifiers.

After that, our aim is to evaluate different FS policies (section 3.2.2). Figure ii.4 shows the experimental results when varying the FS policy in Reuters. In this experiment, we used the standard Information Gain as the TSR function, the SVM classifier, and F_1 as the standard evaluation metric.

In light of the results, it seems that Round Robin is the best policy for Macroaveraging, but not for microaveraging. What should be highlighted is the fact that the Local policy obtains the

[10] `http://hlt.isti.cnr.it/jatecs/`

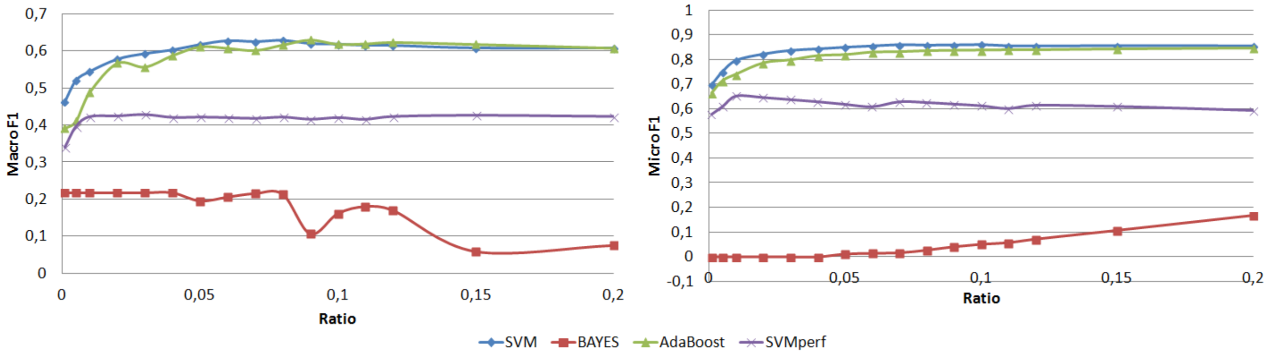


Figure ii.3: Performance Evaluation of the Classifiers in *Reuters* using Information Gain and Round Robin

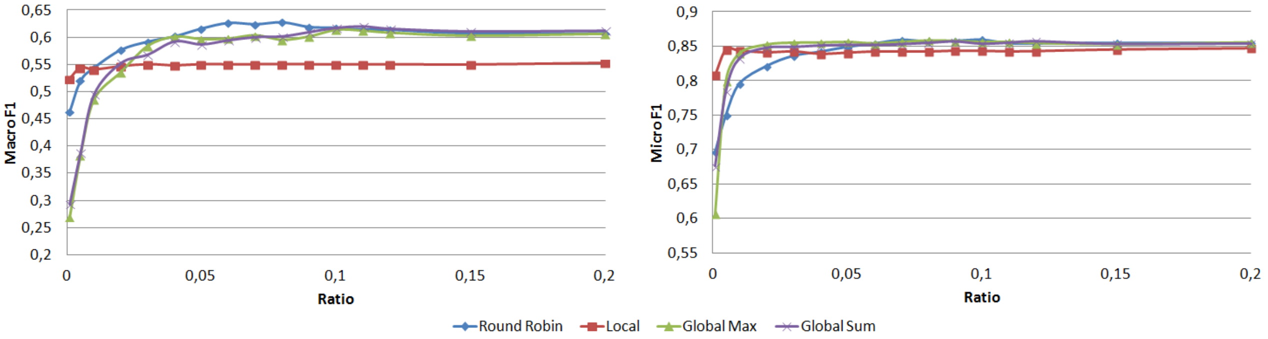


Figure ii.4: Performance Evaluation of the FS Policies in *Reuters* using Information Gain and SVM

best results, both in macroaveraging and microaveraging, for extreme aggressive reductions. These results were obtained in $\xi = 0.001$ and $\xi = 0.005$, that is, each binary classifier was trained with only 24 features or 118 features, respectively. It is also noticeably that this policy does not improve as the ratio increases, obtaining the worst results in further evaluations. In addition, the difference between the Global policies is unnoticeable. Round Robin has proven its effectiveness while dealing with the 'siren pitfall', as reflected in the macroaveraging. That is, when the performance evaluation metric pays equal attention to all categories regardless of the skew, Round Robin seems to be the winner.

Table II.6 shows the percentage differences among FS policies for macro F-score while using SVM and AdaBoost as classifiers. For example, value 5.43 in RR-L column indicates the Round Robin policy obtained 5.43% higher macro F-score than Local policy for the same ratio (row).

Although the Local policy seems to outperform Round Robin for extremely aggressive reduction ratios, this tendency changes gradually as the ratio increases. As it was shown in Figure ii.4, the Local policy seems to be stalled for $\xi > 0.02$ while the Round Robin policy keeps improving until $\xi \simeq 0.08$. The performance improvement among Round Robin with respect to Global policies is more noticeably for smaller ratios. These differences could not be considered relevant for ξ values higher than 0.1 in both classifiers. The Local policy is clearly worse in macro F-score than Global policies for ratios $\xi > 0.03$. As the Local policy is supposedly optimised for each category, we suspect most important features are successfully being selected in early steps, while it tends to fall under the 'siren pitfall' problem in further decisions. Finally, there is no clear indication on the superiority of any of the Global policies with respect to each other. Albeit there could be opposing views in

	SVM						**AdaBoost**					
ξ	RR-L	RR-Gm	RR-Gs	L-Gm	L-Gs	Gm-Gs	RR-L	RR-Gm	RR-Gs	L-Gm	L-Gs	Gm-Gs
.005	-4.58	35.83	34.38	42.34	40.83	-1.07	-17.65	23.77	10.01	50.30	33.59	-11.12
.010	0.49	12.03	10.08	11.49	9.54	-1.75	-4.22	7.53	10.20	12.26	15.05	2.48
.020	5.43	7.58	4.52	2.04	-0.86	-2.84	7.33	11.46	12.36	3.84	4.69	0.82
.030	7.46	1.30	4.27	-5.73	-2.97	2.93	4.23	2.32	5.64	-1.83	1.35	3.24
.040	9.64	0.08	1.67	-8.72	-7.27	1.59	8.35	2.21	0.61	-5.67	-7.14	-1.57
.050	11.85	3.25	5.05	-7.69	-6.08	1.74	12.56	3.90	8.17	-7.69	-3.90	4.11
.060	13.75	4.77	5.44	-7.89	-7.31	0.63	14.27	2.15	5.03	-10.61	-8.09	2.82
.070	13.32	3.33	4.03	-8.81	-8.19	0.68	12.74	-2.27	0.58	-13.32	-10.79	2.92
.080	13.81	5.37	4.36	-7.42	-8.31	-0.96	15.21	2.11	5.03	-11.37	-8.84	2.86
.090	12.38	2.85	1.62	-8.48	-9.57	-1.19	18.06	2.24	5.46	-13.40	-10.68	3.15
.100	12.08	0.59	0.09	-10.25	-10.70	-0.50	15.86	2.11	-0.55	-11.87	-14.16	-2.60
.110	11.58	0.31	-0.86	-10.10	-11.15	-1.17	15.99	1.36	0.32	-12.62	-13.52	-1.03
.120	11.45	0.85	-0.23	-9.51	-10.48	-1.08	19.20	1.60	1.35	-14.77	-14.98	-0.25
.150	10.30	0.66	-0.53	-8.74	-9.81	-1.18	20.31	0.50	0.81	-16.46	-16.21	0.31
.200	9.75	0.10	-0.73	-8.79	-9.55	-0.83	18.54	-0.46	0.45	-16.02	-15.26	0.91

Table II.6: Percentage differences of Macro F-score among FS policies using SVM and AdaBoost classifiers

this respect, we believe Macroaveraging better represents the global performance of a system. For this reason, we will take Round Robin as the FS policy in the next experiments.

Finally, we contrast the performance of TSR filtering function explained in section 3.2.1. In this case, we took Round Robin as the TSR policy. Because Support Vector Machines are low sensitive to high dimensions, one could argue that a SVM would not be an appropriate classifier to investigate the influence of FS. In this respect, we have selected a plot involving both SVM and AdaBoost learners in order to contrast the performance of each TSR function in different learners (Figure ii.5).

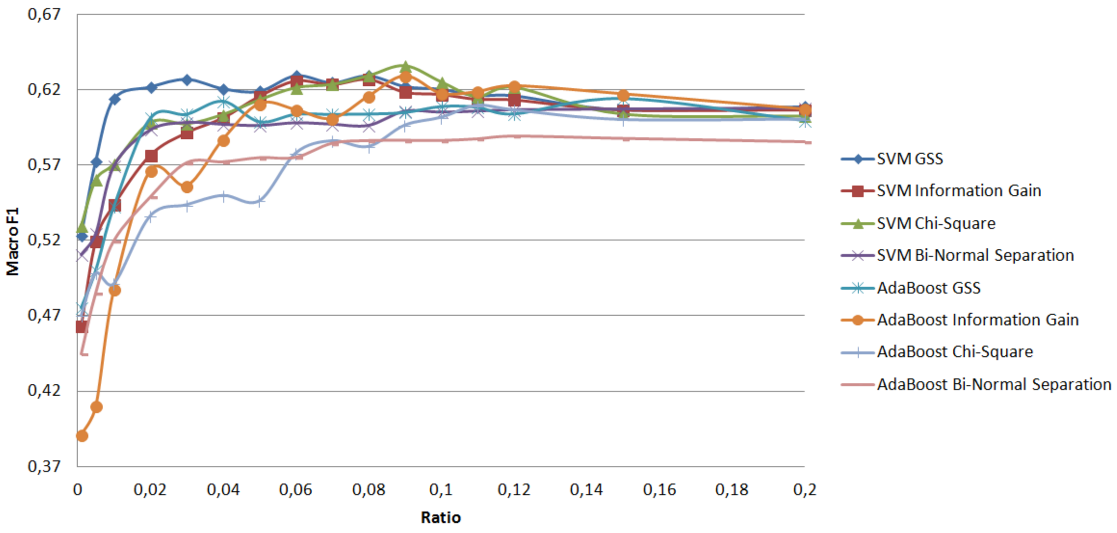


Figure ii.5: TSR functions with Round Robin, using SVM and AdaBoost

As could be seen, SVM seems to outperform the performance of AdaBoost even considering different filtering functions. For now on and for the sake of simplicity, subsequent plots will only represent the case of SVMs. Figure ii.6 shows results for macroaverages and microaverages of Accuracy and F1 measures. Similarly, Figure ii.7 shows results for Precision and Recall.

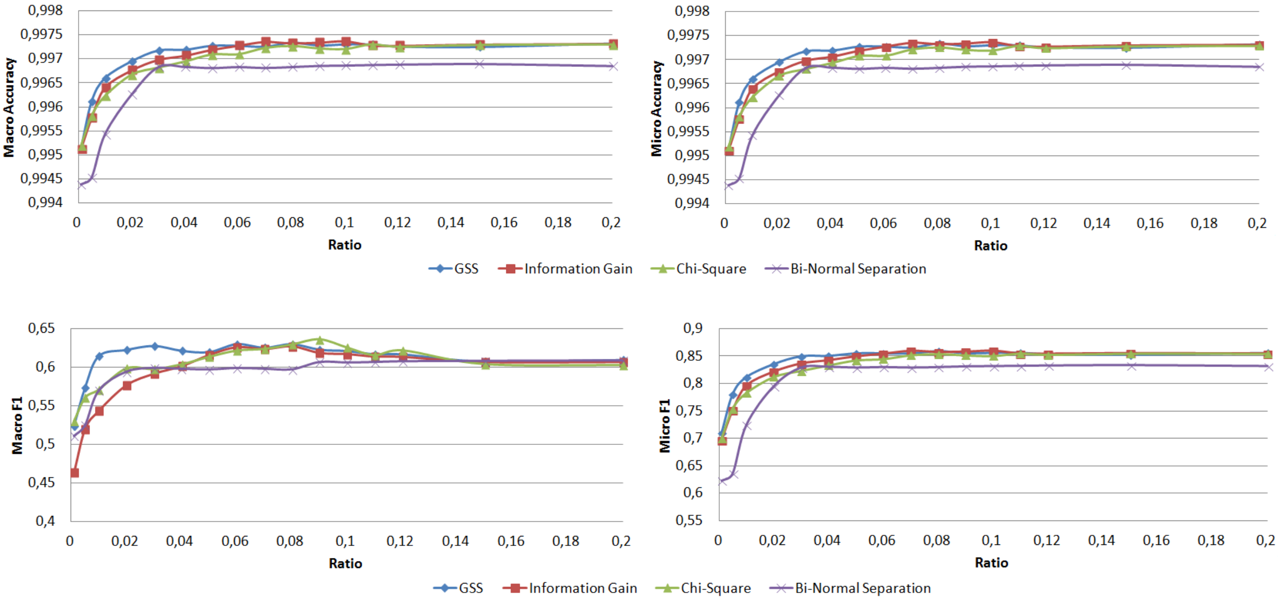


Figure ii.6: Performance Evaluation of the TSR functions with Round Robin and SVM (Accuracy and F-score)

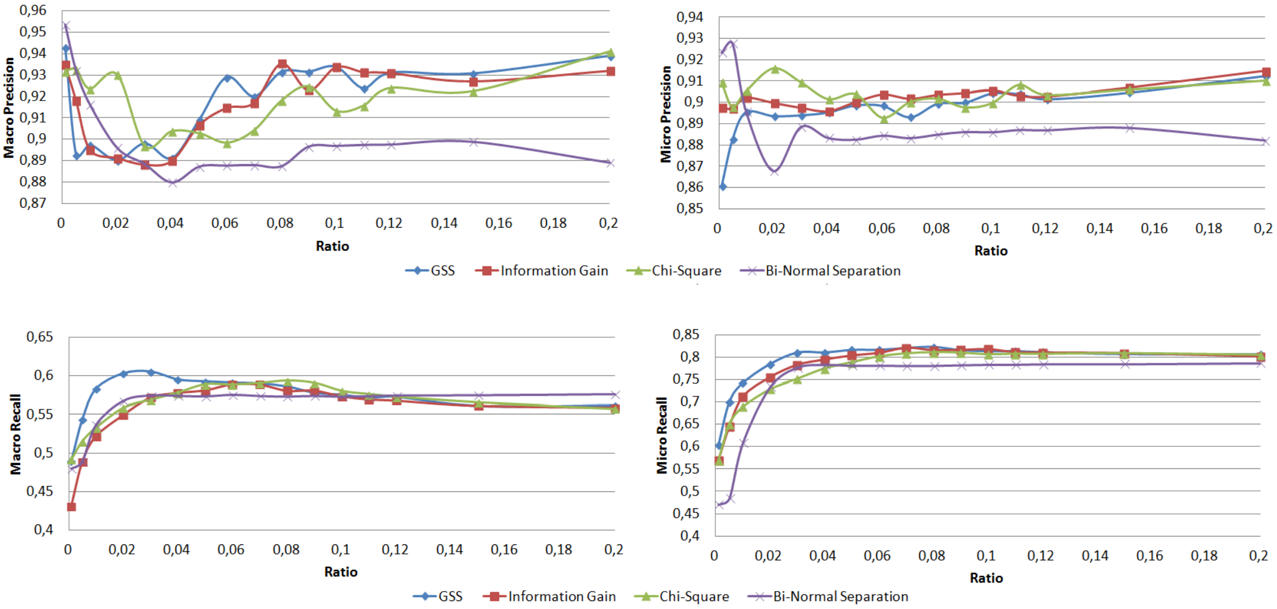


Figure ii.7: Performance Evaluation of the TSR functions with Round Robin and SVM (Precision and Recall)

Table II.7 shows the percentage differences on macro F-score among the various TSR functions under consideration. GSS seems to clearly outperform the rest of TSR functions for aggressive reduction ratios. Indeed, GSS seems to be the best TSR in all cases but while using AdaBoost as the classifier, where Information Gain outperformed GSS for ξ greater than 0.08. In any case, it should be remarked that GSS peaks in smaller ratios, which in turn means SVM with GSS could obtain better performances than AdaBoost with Information Gain even selecting less features (see Figure ii.5). Regarding Information Gain, Chi-Square, and Bi-Normal Separation, we found no clear indication on which of these functions dominates the others —it seems to depend on the classifier at hand. As we are considering the Round Robin policy which selects the same number of features for each category, we think GSS is more effective because it only attends to positive correlation. This could be an indication that negatively correlated features to a category could actually be positively correlated features to a different category. If this is the case, taking them as representative for a category in the Round Robin turn would be unfair —this selection is actually benefiting another category.

	SVM						**AdaBoost**					
ξ	GSS-IG	GSS-Chi	GSS-BNS	IG-Chi	IG-BNS	Chi-BNS	GSS-IG	GSS-Chi	GSS-BNS	IG-Chi	IG-BNS	Chi-BNS
.005	10.23	2.21	9.15	-7.27	-0.98	6.78	21.97	0.41	3.12	-17.68	-15.45	2.70
.010	12.98	7.55	7.80	-4.81	-4.59	0.23	11.23	10.44	4.40	-0.71	-6.15	-5.47
.020	7.89	3.94	4.79	-3.66	-2.87	0.82	6.16	12.14	9.59	5.63	3.23	-2.27
.030	6.02	4.95	4.78	-1.01	-1.18	-0.17	8.52	11.08	5.65	2.36	-2.64	-4.88
.040	3.25	2.84	3.94	-0.40	0.67	1.07	4.35	11.42	7.08	6.77	2.62	-3.90
.050	0.56	0.93	3.79	0.37	3.22	2.84	-1.94	9.63	4.18	11.80	6.24	-4.97
.060	0.58	1.30	5.21	0.71	4.61	3.87	-0.46	4.51	5.02	4.99	5.51	0.49
.070	0.21	0.18	4.58	-0.02	4.37	4.39	0.49	3.02	3.31	2.52	2.81	0.28
.080	0.39	0.00	5.50	-0.39	5.09	5.50	-1.87	3.75	3.05	5.73	5.02	-0.67
.090	0.62	-2.16	2.69	-2.77	2.05	4.95	-3.82	1.45	3.18	5.49	7.28	1.70
.100	0.57	-0.79	2.49	-1.35	1.91	3.31	-1.36	1.25	3.89	2.65	5.33	2.61
.110	0.42	0.13	1.68	-0.28	1.26	1.55	-1.62	-0.16	3.62	1.49	5.33	3.78
.120	0.46	-0.91	1.52	-1.36	1.05	2.45	-2.97	-0.40	2.54	2.65	5.67	2.95
.150	0.03	0.47	-0.11	0.45	-0.14	-0.58	-0.45	2.38	4.55	2.85	5.02	2.11
.200	0.36	1.04	0.12	0.68	-0.24	-0.91	-1.22	-0.10	2.44	1.13	3.70	2.54

Table II.7: Percentage differences of Macro F-score among TSR functions using SVM and AdaBoost classifiers with the Round Robin policy

Although we have experimented on both *Reuters* and *Ohsumed* datasets, all plots above correspond to the former case. Results in *Ohsumed* are consistent with the indications drawn before, but we found more illustrative the examples on *Reuters*. Figure ii.8 presents a zoom of the interesting region involving the TSR functions under consideration and SVMs on *Ohsumed* dataset.

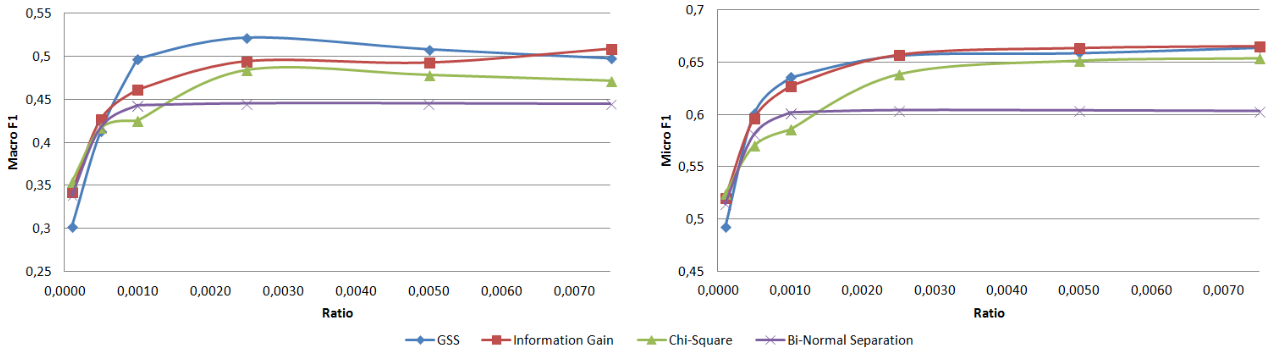


Figure ii.8: Effect of TSR functions with Round Robin and SVM on *Ohsumed* (Zoom)

We found that GSS outperforms the rest of TSR functions in Accuracy, F1, and specially in

Recall —no clear indication could be extracted from the case of Precision in this regard. This could be an indication that attending only to positive correlation could be a better strategy while combined with Round Robin. The reason why, is that Round Robin tends to equally represent each category in the reduced feature space. In this respect, positive correlation may lead to better representations. Thus, a negatively correlated feature is likely to be a positively correlated feature to a different category. If so, attending to negative correlations in a Round Robin manner could be counter-productive.

5 Conclusions and Future Work

The motivation of this chapter was to identify the main properties of relevant terms to a given domain of documents. To this aim, we took the UK/AM configuration as a starting point, in order to tackle the problem from the lowest level of potential knowledge: documents do not present any known structure, and there is not any meta-knowledge resource available, but a set of possible semantic labels.

More concretely, the study has been approached as a Feature Selection for Text Classification problem, the task of reducing the initial term space before building a learner. This learner is aimed for automatically labelling text documents with zero, one, or more pre-existing labels representing semantically-related categories. We have presented an experimental study involving some of the most popular TSR functions, TSR policies, classifiers, evaluation metrics, and datasets. Results revealed SVM is the top-performing classifier, Round Robin is the best TSR policy for macroaveraging, and GSS in combination with Round Robin, the most promising TSR filtering function.

Apart from these clues, additional conclusions could be drawn in light of these results. Our aim was to determine the characteristics of the top-relevant terms to a domain. The lesson learned suggests that, as pointed out by Yang and Pedersen [YP97], classification could be efficiently approached from an extremely reduced feature space. Furthermore, while searching for the most important terms in a domain, all categories should be equally attended —otherwise, some classes may result ignored in the global process. Finally, positive correlation —rather than both positive and negative— seems to be an appropriate criterion to lead the selection of terms to differentiate classes in Natural Language.

Since it was well-accepted that positive and negative correlations should be equally taken into account, we think it is worth performing more experimentation to empirically accept or refuse the validity of our assumptions. It goes for future research investigating how new TSR functions could exploit, in a more sophisticated manner, the positive correlation while keeping track of high-skew categories. Furthermore, it was pointed out in [TA03] that FS is affected by the evaluation measure being considered, and in [XLL$^+$08, Joa05] that the classification effectiveness could be improved if the learner is aware of it. We are thus interested in dealing in depth with these issues in future research.

Next chapter is dedicated to problems and methods dealing with unstructured data, but counting with additional meta-knowledge resources. We will approach this study through Sentiment Analysis, a problem that received several contributions from the ML community. However, our proposal follows a different strategy, along the lines of lexicon-based methods. Thus, we will investigate a different sort of analysis that exploits semantic modelling. Some particular difficulties we will address in next chapter include named entities recognition, the effect of linguistic modifiers, and knowledge modularization.

Chapter III

Lexicon-based approaches: Unstructured Knowledge, Presence of Meta-Knowledge

1 Introduction

The Internet is currently evolving towards the Web 2.0. This trend as well as the fact that the Internet is now within almost everyone's reach because of cheaper hardware has led to a cultural revolution. People from all over the world are now able to interact with each other. As a result, a number of social networks, blogs, forums, etc., where they can enjoy both freedom of speech and easy access to all types of information have emerged. Within such contexts, a large quantity of unstructured information is constantly appearing. This kind of documents is usually known as *user-generated content*, and unarguably represents a promising source of useful information for companies, organizations, and even users. For example, companies and organization could take advantage of automatically generated opinion reports to design better marketing strategies, instead of performing costly user satisfaction surveys. Politicians could also gauge public opinion to decide different political campaigns. Also customers could benefit from these techniques, by obtaining meaningful opinion summaries about a given product or service. The number of potential applications within this context has no end.

It is thus of the utmost importance the design of rich automatic processes able to deal with these kind of documents for different specific purposes. Given there is no *a priori* clue of the documents structure, delimiting the domain context seems to be an appropriate starting point [MCZ12]. That is, trying to model the knowledge domain in advance could be a fruitful strategy. This modelling might consists of representing most relevant entities along with their relations from a semantic point of view, in such a way that an automatic NL process could count with enough information in advance as to analyse some domain-related unseen texts.

Representing these semantic meta-knowledge resources conform a difficult challenge on its own. In this regard, different approaches have been proposed, including formal logics (from Aristotle to nowadays), semantic networks (first proposed by Quilliam, based on the notion of IS-A and inclusion relations), frames (developed in the early 70s to define entities and instances on the notion of *slots*), Entity-Relationship diagrams (proposed by Chen in 1976 to represent DataBase conceptualizations), or Object-Oriented models (defined as part of UML), to name just a few. Probably, the most extended models in current research include *Ontologies* and *Lexicons*. Ontologies are aimed for specifying a shared semantic representation of the conceptualization on a given domain, by formally defining a set of objects, properties, relations, individuals, etc. (domain ontologies will be later explained in more detail in chapter V section 1.3). Domain Lexicons are rather knowledge representation models focusing on the linguistic accesses to domain concepts (section 1.2). Examples of methods to create, enhance, and adapt these semantic meta-knowledge resources could be found in [ES06a, ALM$^+$03] —we will deal with this particular issue in chapter V.

The main goals we set for this part of the Thesis include how to properly represent the linguistic knowledge resources and how these resources could be efficiently used for an automatic process to automatically identify the main entities involved in a text document. To this aim, we will deal with certain NL-related difficulties, among which we pay special attention to the *anaphora*, *ellipsis*, and *ambiguity* phenomena from a computational point of view. Note these problems represent a fundamental aspect to success in recognizing most important named entities. All these considerations will be faced under the premise of closed-domains. To this purpose and according to the proposed methodology, we have chosen a representative problem. In this regard, the automatic analysis of opinions expressed in texts have received a great deal of attention on recent years. This problem, known as *Sentiment Analysis* or *Opinion Mining* [PL08, CZ10] (see section 1.1) has received a great number of contributions from the Text Classification [PLV02, MC04] and Computational Linguistics [WWB$^+$04, HM97] communities due to the increasing number of potential applications that could benefit users, companies, or political parties in decision making [CM06], marketing strategies [HL04a], or political strategies [MM08, AG05], respectively. We thus propose the study of Sentiment Analysis (focusing on news items) as a representative problem under the UK/PM knowledge level in this chapter. More specifically, we will focus on Feature-based Sentiment Analysis (explained below), a branch of Sentiment Analysis that is usually approached through lexicon-based methods to keep better track of domain-specific entities.

The rest of this chapter is structured as follows. Section 1.1 offers an introduction to the Sentiment Analysis problem. The concept of Domain Lexicon will be explained in section 1.2. We present our study on Sentiment Analysis in section 2 and conclude with some final discussions in section 3.

1.1 Sentiment Analysis: Problem Statement

The computational study of opinions, sentiments, and emotions expressed in texts is known as Sentiment Analysis or Opinion Mining [PL08, CZ10]. This includes the automatic extraction of opinions and the analysis of sentiment. Although opinion is very broad concept, Sentiment Analysis has thus far mainly focused on positive and negative sentiments. This research area evaluates words and sentences in opinion-expressing documents with a view to studying their *subjectivity*, *polarity*, and *strength*: (i) subjectivity is the extent to which a text is objective, and whether it contains sentiment expressions with subjective views; (ii) polarity is the extent to which the text expresses a positive or negative sentiment; (iii) strength is the degree of polarity or intensity of the opinion. Furthermore, another objective is the discovery of the main topics on which user opinions are expressed. Thus, main goals in SA include determining the sentiment of a given document, finding a meaningful manner to present sentiment summary reports, or answering opinion-oriented queries such as "Which documents express opinions on X?". In this regard, direct applications of SA techniques include classification of product reviews (see below), recommendation systems, subjective question-answering, opinion spam detection, subjective-mail identification, opinion summarization, reputation tracking, and business-intelligence (for sales prediction, reputation management, or political strategies), to mention a few.

Earlier studies on Sentiment Analysis have focused on the classification of product reviews

[Den08, MLD08, HL04a, PLV02, MC04]. Their goal is the extraction of positive or negative sentiments in user opinions of a product or some of its features in order to classify the reviews of the product. This task, which is generally known as *overall sentiment*, operates on a document level. Therefore, the sentiment extracted is atomically attached to the reviewed product. This type of evaluation mostly centers on polarity (positiveness or negativity) and optionally, on its strength or intensity. This research area is certainly useful for companies offering products or services. Instead of conducting costly market studies or customer satisfaction analyses, companies are able to analyse published reviews to determine user affinity to their products. In fact, related research studies have focused on the following: (i) the acceptance of user reviews rather than those in other more conventional information sources [BS01]; (ii) the commercial interest deposited in online user opinions of products and services [Hof08]; (iii) the growing influence of these views on the purchasing decisions of other users [CM06].

In addition to overall sentiment, there is another strategy in Sentiment Analysis that also considers the subcomponents and attributes of the product or service individually. Such approaches provide a review of each feature [JL06, MLD08, HL04a, DLY08], rather than giving a single overall rating. This kind of analysis is known as *feature mining* or *feature-based* sentiment analysis, and is often applied to product reviews.

Sentiment Analysis has been approached from different strategies, including mainly the machine-learning approaches [PLV02, MC04, Tur02a, TZ08] and the dictionary-based approaches [DA07, Den08, ES06a]. On the one side, ML approaches rely on a previous training stage aimed for automatically learning the underlying patterns to correctly classify unseen documents. To this purposed, some well-known learners have been applied, including Support Vector Machines, Naïve Bayes, Maximum Entropy models, or regression models [PLV02, ZV06, MHN$^+$07]. On the other side, dictionary-based approaches extract the polarity of each sentence in a document. Afterwards, the sense of the opinion words in the phrase is analysed in order to group polarities, and thus classify the sentiment of the text. Generally speaking, the techniques that follow this approach are based on lexicons, and use a dictionary of words mapped to their semantic value [DA07, Den08], such as MPQA lexicon [WWH05], WordNet [MBF$^+$90], or SentiWordNet [ES06a], an enhanced lexical resource for supporting sentiment classification. Although these dictionaries are usually handmade, some approaches use “seed words” or ML approaches to automatically expand knowledge [HM97, GSS07]. Further discussions on main methods proposed from the ML and dictionary-based perspectives could be found in section 2.2.

This section was meant to offer a brief description of the Sentiment Analysis problem, its main approaches and principal difficulties. It was not however meant to be an exhaustive review of the field. The interested reader is rather referred to the excellent surveys of B.Pang and L.Lee [PL08] or the more recent of B. Liu [Liu12].

1.2 Domain Lexicons

According to Rochelle Lieber, “the Lexicon consists of a list of all unanalyzable terminal elements and their lexical entries. Inflectional stems variants are listed, with relationships among them expressed by means of devices called morphological rules” [Lie80]. As lexicons heritage from linguistic theories, there are as many different representations as different theories. This causes a lack of standardisation in lexical resources, in contrast to other well-known knowledge representation mechanisms such as OWL, RDF, or SQL. To give an example, in [Pus91] it was argued that the tendency of solely representing the morphological level, with independence of the syntactic and semantic level, is advocated to fail. Apart from these advanced linguistic theories that fall outside the scope of this research, it is well-accepted for most linguistic theories that a lexicon may contain the *words*, *bound morphemes* —like affixes and postfixes—, *idiomatic expressions*, and *collocations*, separately from the syntactic level, e.g., a grammar.

From a more technical point of view, a lexicon will here be understood as a data structure containing the necessary lexical resources for any particular NLP system. The more advanced processing is required, the more complex the data structure might be. As a result, different representations may range from simple plain-dictionaries to complex semantic representations bringing together hierarchical structures with relations among entities in the direction of ontologies. In any case, given this dissertation is not meant to be a treatise on linguistics, we will address the issue from a mere computational point of view, mostly focusing to the problem at hand, e.g., Sentiment Analysis. In this particular case, lexicons are rather referred to as *Sentiment lexicons*.

Usually, two clearly differentiated levels are considered in sentiment lexicons: *sentiment words* —also called *opinion words*, *polar words*, or *opinion-bearing words*—, and *features* —supposedly objective terms naming concepts in the domain [QLBC09]. Sentiment words are usually accompanied by *valence shifters* [CC08b, DNML10], linguistic particles that may modify the meaning of the attached word (for example, "less" could diminish the sentiment strength of the following word, while "strongly" could reinforce it). Term *sentiment words* also includes idioms or sentiment phrases. Usually, sentiment words could be divided in *base type* (like "beautiful", "wonderful", or "poor") and *comparative* (like "better", "worse", or "best") [Liu12, chapter 6]. Furthermore, features could be hierarchically structured according to PART-OF [JL06], or IS-A [MRCZ12b] relations. For example, "screen" is a component of "mobile phone", while "singer" is a subclass of "person".

Since the definition of features is inevitably affected by the domain extension, some techniques have been proposed to discover new expressions semiautomatically, including [LMS06] to detect new objective expressions, [NPI11, BV04] to detect new subjective expressions, or [TW11a] to adapt the strength and polarity of subjective expressions from one domain to another. In this regard, some techniques have been proposed to enhance already existing lexicons [QLBC09, ALSZ06, ALM$^+$03]. According to [Liu12], different strategies to create lexicons could be classified in Manual approaches [Tur02a, ES06a, KH06, VBGHM10], Dictionary-based approaches [Tur02a, ES06a, KH06, VBGHM10], and Corpus-based approaches [HM97, WWH05, KN06] (see chapter I section 2 to access a broader discussion on each approach).

Finally, several general-purpose sentiment lexicons have been constructed and are publicly available. Some examples are listed below: General Inquirer lexicon[1] [SKA68], Sentiment lexicon[2] [HL04a], SentiWordNet[3] [ES06b], and MPQA[4] [WWH05]

[1] (http://www.wjh.harvard.edu/~inquirer/spreadsheet_guide.htm

[2] (http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html

[3] http://sentiwordnet.isti.cnr.it/

[4] (http://www.cs.pitt.edu/mpqa/subj_lexicon.html

2 Opinium: A Lexicon-based Framework for Sentiment Analysis of News

2.1 Introduction

The recent evolution of the Web 2.0 has generated a large quantity of social media, including social networks, blogs, forums, etc., where users can enjoy both freedom of speech and easy access to all types of information. Within such contexts, users are able to express their opinions on any relevant topic. This type of scenario includes the interactive press or online news sites with publications on current events. These sites encourage user communities to say what they think about breaking news in categories ranging from sports to controversial social debates. The proliferation of user comments, favoured by anonymity, generates large quantities of unstructured information, the so-called *user-generated content*, and its analysis helps to shape a social barometer pertaining to any issue. The study of this barometer provides a popularity measurement of news in a global context, based on user comments. For example, a headline concerning a new economic measure not only generates evaluations of the measure itself, but also of the government that enacted it. Nonetheless, given the sheer quantity of news published on the Internet, it is very difficult, if not impossible, to manually analyse it all. An automatic method is thus needed that is capable of processing and analysing the information conveyed in news comments. This chapter describes a new approach to the automated analysis of user-generated content.

Sentiment Analysis or Opinion Mining (see section 1.1) emerged thus as a promising field aimed for dealing with opinion-bearing texts. Among the various approaches in the literature, studies more closely linked to the context of our problem, such as socio-political studies, show that the "geopolitical" web can improve the extraction of citizens' opinions regarding the most important public issues of debate, particularly political issues [MM08, AG05]. These studies are even able to classify news items based on their evaluation [BCP$^+$07, JL06, WWB$^+$04]. Nevertheless, the extraction of sentiment from any informative text is very difficult because news items are supposedly objective and free of polarity. Thus, it seems more appropriate to focus on comments in order to analyse opinions of the news.

Most Sentiment Analysis techniques can be divided into ML approaches and dictionary-based approaches. Even though ML approaches have made significant advances [PLV02, MC04, Tur02a, TZ08] in sentiment classification, applying them to news comments require labelled training data sets. The compilation of these training data requires considerable time and effort, especially since data should be current. In order to alleviate this task, applications to generate annotated corpus data were proposed [WWC05].

In contrast, dictionary-based approaches have important advantages, such as the fact that once they are built, no training data are necessary. Unfortunately, they also have certain drawbacks. First, most are designed as glossaries of general language words, often based on WordNet, and thus they do not contain either technical terms or colloquial expressions (Example *a*). Secondly, since they are unable to consider context-dependent expressions, such systems achieve very limited accuracy in multi-domain scenarios [Den09] because the connotation of certain words can be either positive or negative, depending on the context in which the words are used (Example *b*).

Feature extraction in Feature-based analysis is usually based on the strong assumption that a review is of a single product [HL04a, GSS07] or a comparison of several products of the same type [PE05], such as comparing the features of different cameras. For this reason, although the term "feature" is appropriate in a product review, this is not necessarily the case in news analysis because

	Comment	Analysis
a	"The prime minister has *lost his mind*".	Negative opinion of the subject "The prime minister", using colloquial language.
b	"The battery life is *too short*". "It wasn't so bad. The wait was *short*".	The sentiment of the adjective "short" is negative in the first phrase and positive in second one.

there may be more than one object evaluated. Other authors have proposed different terms such as "topic" [KH04] or "aspect" [KIM07]. This research study uses "focus", which is more appropriate in this context to refer the entity target of opinion.

Our proposal is that all focuses in the news comments with their independent valuation must be analysed. Thus, the extraction of features should be multifocal. The same expression could refer to features of different objects depending on the context. However, the computational analysis of opinions is inevitably affected by inherent difficulties presented in natural language. Ambiguity, anaphora, and ellipsis, are examples of context-dependant problems attached to natural language. The following example is meant to illustrate each case:

Problem	Example	Analysis
Ambiguity	"animal"	brutal person (subjective), or animal (objective)?
Ellipsis	I found it too *expensive*	expensive is a feature of *price* (omitted).
Anaphora	*It* is quite large	anaphoric pronoun *it* must be resolved to decide whether the sentiment is positive or negative.

Comment-oriented polarity extraction also provides new challenges for Sentiment Analysis, as there is a high probability that users express their opinions of focuses that do not explicitly appear in the body of the news article. In addition, given that comment holders (people that express opinions) are usually anonymous, this often leads to malicious users expressing offensive opinions, or even using their comments to advertise their own websites. Although some news media sites allow users to denounce such behaviour, this is not always the case. Accordingly, useful information is sometimes mixed with noisy data that makes analysis more difficult. Detecting and filtering out irrelevant information in user-generated content is a subtask of vital importance when performing sentiment analysis known as *Opinion Spam Detection* [JL08]. Our approach includes a comments filter that discards comments that are likely to be noisy.

Our lexicon-based approach consists of a practical system to deal with the difficulties of the analysis of user comments on news articles. As a starting point, we consider that each comment may convey an opinion on the general topic content in the news item or even on another specific topic related to the general topic. Comments on Example *c* may be on the following news headline: "The X football team wins the European Championship".

	Comment	Analysis
c	"The Y team is much better!" "Player Z is the best of the team".	Both comments convey opinions on focuses different from the main focus. They must be analysed separately after obtaining the focuses of the comment.

Examples above lead us to think that delimiting context is crucial to support the analysis of sentiment expressed in user comments. We thus propose *Opinium* [MRCZ12b], a lexicon-based news sentiment analyser algorithm that relies on a previous context analysis [MCZ12] to summarize the sentiment expressed in a set of comments (Figure iii.1).

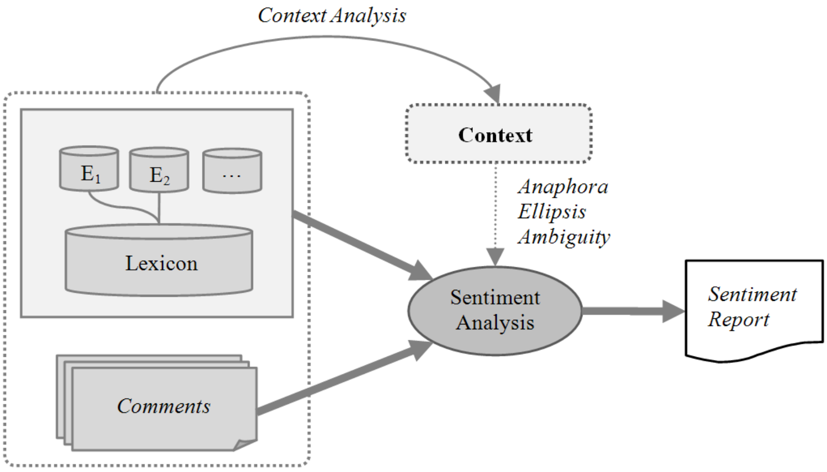


Figure iii.1: Context-based Sentiment Analyser method

The structure of our lexicon is specifically designed to analyse news comments. It has been built bearing in mind the cross-domain nature of news items allowing the adaptation of the lexicon.

The rest of this chapter is organized as follows. An overview of previous work on sentiment analysis is presented in section 2.2. Section 2.3 describes the structure of our application, and section 2.4 provides a detailed explanation of our lexicon and its structure. The method of analysis is outlined in section 2.5. Finally, section 2.6 discusses the experimental validation of our method, and section 2.7 concludes with a discussion of results and future research.

2.2 Related Work

This section describes the state of the art in Sentiment Analysis, and discusses the most relevant research in the field. First, we focus on the user-comment approaches that are most closely related to our research. This is followed by a discussion of machine-learning methods, dictionary-based approaches, and their variants. Finally, we give an overview of Feature Mining, a variant of Sentiment Analysis, which as of late has received a considerable amount of attention from the research community. Note that machine-learning methods falls beyond the scope of this study, e.g., they do not need any meta-knowledge resource. However, given the imminent proliferation of these methods in the field of SA, ignoring them would be completely unfair.

In [HSL07], the authors performed an automatic blog posts summary of the information contained in user comments. They used a graph-based system of weights that draws on the words in the most cited comments related to the most popular topics. Their system then selected the blog post phrases containing the most representative words obtained previously. In [Del06], Delort extracted clusters of comments. Interesting clusters were selected manually and used to extract the blog sentences most closely related to the comments in the cluster. [ARSX03] classified users on opposing sides of an online discussion by means of a graph linking them to comments of the type “answer to”. In their study, [MM08] ranked users according to their political orientation (leftwing, rightwing, other), based on comments made in American policy forums. For this purpose, the authors used a variation of the PMI-IR method, Naïve Bayes, and a social network analysis method, based on graphs.

Machine-learning and dictionary-based approaches have been applied to product reviews with promising results. Turney developed an unsupervised learning algorithm to classify texts as recommended or not recommended [Tur02a]. This algorithm, known as Pointwise Mutual Information and Information Retrieval (PMI-IR), calculated the semantic orientation of a sentence by assigning the numerical value resulting from the information shared by the sentence and the word "excellent" minus the information shared by the sentence and the word "poor". The text was classified as *recommended* if the average value was positive, and the magnitude of this numerical value was regarded as indicative of the strength of its semantic orientation. This algorithm has been used in many subsequent investigations. [PLV02] used three machine-learning techniques to classify IMBD movie reviews as positive or negative. Their conclusion showed that these three techniques (Naïve Bayes, classification maximum entropy, and SVM) outperformed the human-generated baseline, and that SVM was the technique that yielded the best results. In [ZV06], the authors developed several regression models to predict a review's usefulness, on the assumption that this usefulness is orthogonal to its polarity. They concluded that shallow syntactic features were the most influential utility predictors, and remarked that a review's usefulness depended heavily on linguistic style. In their approach, [ES05] proposed a semi-supervised method of performing a binary classification of texts as positive or negative, assuming that terms with a similar orientation tend to have similar definitions (glosses). [BCP$^+$07], proposed a linguistic approach to determine the strength and polarity of a topic in a given text, and applied a technique based on adjective and adverb combinations (AACs). They presented three scoring axioms which defined the strength and polarity of a given AAC. Nasukawa and Yi offer an alternative method, and propose extracting sentiments associated with subjects that recur throughout the text, rather than providing a valuation of the document as a whole [NY03]. Accordingly, they used a syntactic parser to identify relationships between sentiment expressions and the subjects on which they give an opinion. Afterwards, a sentiment lexicon was used to establish the polarity of the sentiments expressions. Other examples of sub-sentential machine learning methods include fully-supervised work [MHN$^+$07] or weakly-supervised models [YYC10, TM11].

Within dictionary-based approaches [TBT$^+$11], systems generally use pre-developed dictionaries containing the polarity of words or phrases. The most frequently used resource is currently SentiWordNet [ES06a], which has been employed in a large number of research studies. In [Den08], the authors use SentiWordNet to determine the polarity of phrases in a multilingual context and classify documents according to polarity. [OT09] also used SentiWordNet in a study in which the dataset is a set of film reviews. They performed a similar classification, and concluded that the results provided by SentiWordNet were close to the results obtained with handmade lexicons. Meanwhile, Dang et al. developed an algorithm that combines content-free (lexical, syntactic and structural features), content-specific (keywords and phrases by n-grams) and sentiment techniques (SentiWordNet) for the classification of online product reviews [DZC10]. These researchers concluded that the combination of machine-learning techniques and dictionary-based techniques substantially improved sentiment classification. [GSS07] implemented a lexicon-based system for news and blogs analysis built on top of the Lidia text analysis system. They propose a method to expand candidate seed lists opinion words through WordNet. Several automatic techniques to create lexicons have been proposed [NPI11, TW11b]. However there is no evidence that those lexicons perform better than manually-built ones in cross-domain scenarios [TBT$^+$11]. In [WT11] the authors propose a framework for cross-domain sentiment classification, defining a "bridge" between one existed domain to a target domain. There have been some papers evaluated on the MPQA corpus[WWH05]. Those studies have included lexicon and machine learning based approaches [WWH05] and have tackled structured opinion extraction [CBC06]. There are also approaches that use taxonomies for product sentiment analysis based on component-subcomponent scheme [CNZ05].

Feature Mining is a variant of Sentiment Analysis that also focuses on capturing the particular sentiment evoked by an object. However, it is based on the valuation of its features and subcomponents. Jindal and Liu proposed a comparative sentence-based method [JL06] capable of extracting the set of relations between text entities (e.g. products or product features) by using two types of Sequential Rules. The features of the products were thus identified and specified. The use of this method in product reviews is very interesting because comparisons of products are very common on the Internet.

Other research based on feature mining is that of [MLD08]. In this study, the researchers extracted product features and related opinions from unstructured reviews. Their algorithm received a semi-structured set of reviews with prior knowledge that was gradually enriched with results using linguistic similarity features. [HL04a] used data mining techniques and natural language processing techniques to obtain the feature polarity of a product that had been reviewed. It was assumed that the main features and their valuation explicitly appeared as nouns or noun phrases in the text of the product review. Frequency distributions were used to find the features by proximity. Afterwards, the polarity of the comments on each feature was obtained, and the results summarized, using WordNet.

[DLY08] considered the problem of context in feature mining. Since the same opinion word can have different orientations in different contexts, both opinion words and features were treated as a tuple called "opinion context". In this way, their system provided a valuation of features based on such tuples by using a lexicon. The system also provided techniques to correctly determine the tuple when ambiguity occurred in opinions related to one feature. In [AGI07], the authors used a hybrid of conventional text-mining techniques and an econometric model (similar to hedonic regression) to estimate the strength and polarity of product features. [HL04b] proposed a set of techniques to detect product features, their frequency of occurrence in the document, and the valuation given by the users. In [YNBN03], the authors developed a feature term extraction system based on a mixture language model and likelihood ratio. A sentiment lexicon was used to assign sentiment phrases to features. Also, important works on topic models for opinion-topic extraction could be consulted in [LCDZ11, TM08, LFW$^+$08]. These use varying levels of supervision, being evaluated on reviews.

2.3 High-performance System for News Analysis

In this section, the general architecture of *Opinium* is explained. Our goal was to extract users' opinions by analysing their comments. We also aim to analyse not only the sentiment in the entire document (overall sentiment), but also the sentiment evoked by each discussion topic (sentiment focus). Therefore, two major problems should be addressed: (i) the identification of those discussion topics on which users have expressed their opinions; (ii) the extraction of their sentiments in such a way as to avoid interferences between them in the analysis. However, to achieve this, knowledge in the lexicon must be adapted to the news domain being analysed. Spam comments should also be detected and discarded in order to prevent noisy data interferences.

We thus propose a Lexicon-based Sentiment Analysis algorithm (Figure iii.2). The Lexicon used was hand-built from the study of 250 news items. It consists of a structured dictionary of linguistic expressions that can be enhanced with domain extensions. As shall be seen, those extensions contain lexical information on current issues (politics, sports, law, etc.). In the adaptation stage, generic knowledge from the initial lexicon is extended in order to adapt it to the specific domain. This is performed by adding domain-dependent terms and entities. A filter is then applied to the news with a view to excluding all the inappropriate comments so that they will not influence the analysis. Both knowledge in the Lexicon and user comments are preprocessed, using various Natural Language Processing (NLP) techniques. Then, an Opinion Focus Detection algorithm is applied to the body of the news and on the comments from users, in order to discover the main discussion topics (opinion focuses). The Sentiment Analysis module analyses each sentence in the comments to create a set of tuples that abstractly represents users' opinions by assigning sentiment expressions to opinion focuses. Finally, these tuples are used by the Mining module, which computes the overall sentiment as well as each focus sentiment on the basis of polarity and strength.

In the following sections, each of these stages is discussed in detail. A description of the structure of our lexicon is also provided as well as of the representation mechanisms adopted for storing the linguistic expressions.

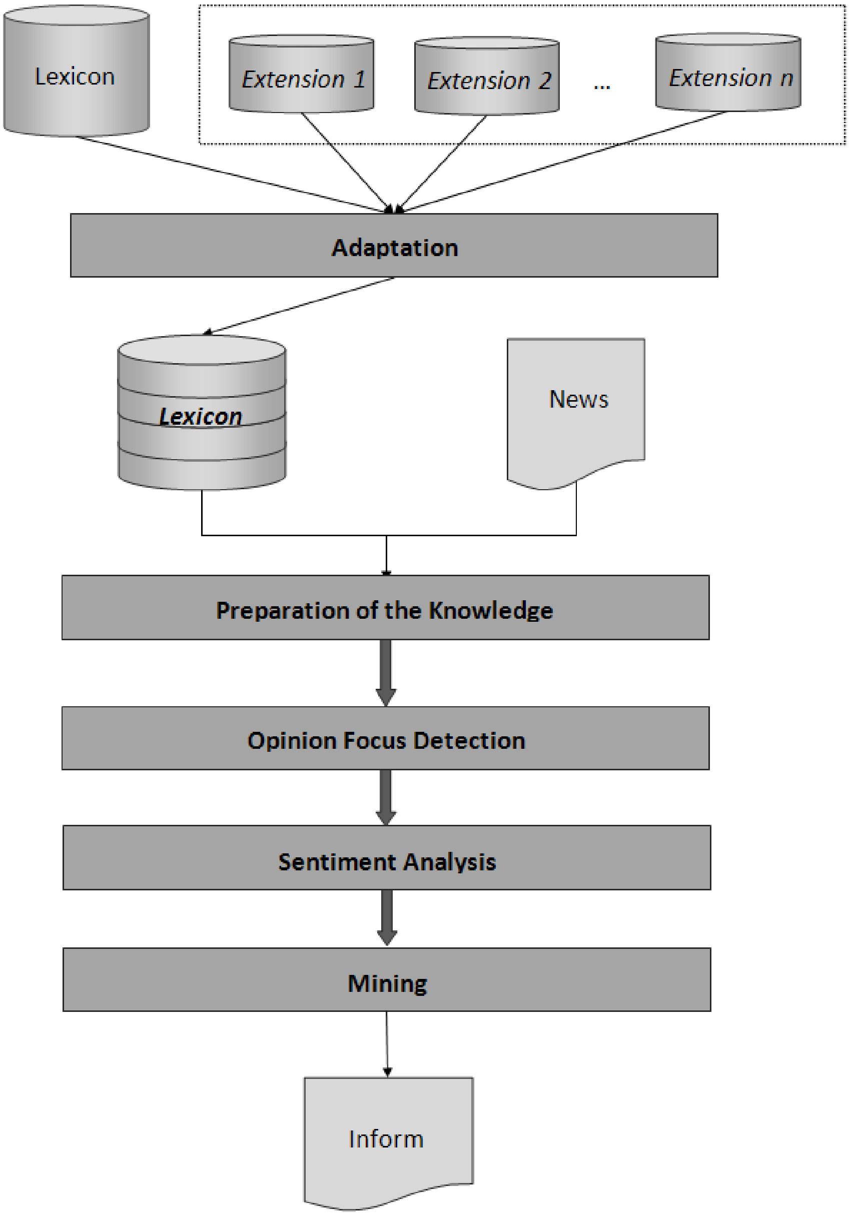


Figure iii.2: *Opinium* architecture.

2.4 Linguistic knowledge: Lexicon

As mentioned earlier, many of the proposals addressing the problem of Sentiment Analysis use dictionaries or lexicons on which the sentiment expressed in a set of words is mapped. These words, known as *opinion words*, are used with a shallow parsing perspective in order to capture the sentiment of users' sentences. Our Lexicon also stores a set of objective expressions (terms that do not express any opinion), which are used to detect the focus references in the comments. The main differences from other lexicons, such as SentiWordNet, is that linguistic expressions are stored as well as single words. In addition, the terms in our Lexicon have been collected by hand from real comments allowing us to capture the colloquial language that predominates in user language. Since this colloquial language is full of non-standard expressions, we believe that the analysis of news comments would be incomplete if only standard dictionaries were used. Other authors have already proposed lexicons that rely not on standard dictionaries, supporting colloquial language and multi-word expressions [VBGHM10].

In the following sections, we first discuss the structure and design of our Lexicon, and then explain how the sentences are mapped onto the Lexicon.

2.4.1 Lexicon Structure

Many authors have already broadly addressed the problem of classifying words to build lexicons incorporating semantic orientation of individual words and contextual valence shifters [CC08b, DNML10]. Moreover several algorithms to enhance initial dictionaries have been proposed and tested [NPI11, ALSZ06]. In this study we focus on defining a hierarchical model based on relations between entities. The highlight of our lexicon consists of its practical and interpretable extensible framework.

Our Lexicon is structured as a taxonomy of objects and features that represents the most recurrent opinion focuses in the news. An *object* may be a person, an entity, a product, etc. The *features* are heritable characteristics of each object. For example, RESISTANCE, SPEED, and STRENGTH are some of the features of the SPORTSPERSON object. For the sake of simplicity, the term focus is used interchangeably to refer to both objects and features. Capital letters are used in the notation. Figure iii.3 shows the section in the Lexicon related to SPORTSPERSON and SOCCER_PLAYER. Note that the object FORWARD inherits SIGNING and PASSING features from SOCCER_PLAYER, and RESISTANCE and SPEED from SPORTSPERSON. Even though a taxonomy relations model with objects has previously been used in product reviews [QLBC09], our taxonomy is different in that no part-of relations (component-subcomponent) are defined. Instead, it uses hierarchical relations, such as in Object-Oriented models (class-subclass).

Additionally, the lexicon contains not only generic valuation expressions but also the particular expressions that users could employ to evaluate a specific focus. For example, "jalopy" is a subjective expression with a negative connotation of the implicit object CAR.

Classical techniques on Sentiment Analysis only consider the polarity of opinions (Negative, Neutral or Positive). According to [WWH04], the strength of the opinion should also be represented. For this reason, this study considers valuations such as VERY_NEGATIVE, NEGATIVE, NEUTRAL, POSITIVE, and VERY_POSITIVE.

The Lexicon is thus divided into three Classes: (i) a set of hierarchically-related Objects (O); (ii), a set of object Features (F); (iii) a set of Valuations (V).

To select the focuses in our lexicon, we collect the most recurrent discussion topics recent news

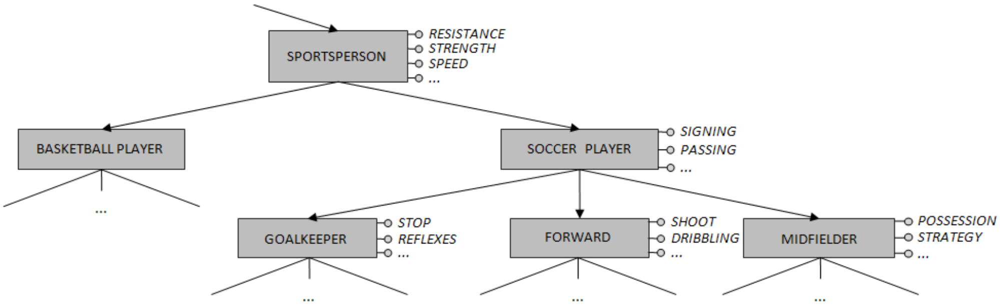


Figure iii.3: Section of the Lexicon related to SPORTSPERSON and SOCCER_PLAYER objects.

about sports, politics, economics, current events, and entertainment. The main nodes in our lexicon taxonomy (and some of their child nodes) are the following: PERSON (politician, sportsman, artist, etc.), LEGISLATION (norm, law), ENTITY (institution, group, public organisms, etc.), LOCATION (country, state, etc.), HAPPENING (event, accident, crime, etc.), PRODUCT (entertainment, technology, Internet, etc.), and DISCUSSION_TOPIC (immigration, abort, gay marriage, etc.). As shall be seen, it is complex to keep such knowledge updated. For this reason, we have designed a generic lexicon, which only contains abstract knowledge, and various extensions, containing modularized and updated knowledge. Consequently, the general lexicon only includes abstract entities, such as PERSON or its child-node SINGER, and lexicon extensions include concrete entities, such as L.PAUSINI, referring to the Italian singer *Laura Pausini* defined in a music extension (Figure iii.4). In order to identify new concrete entities and features to update the extensions some semiautomatic approaches such as [YNBN03], or [ZXKJ11] in Chinese texts, could be helpful. From a more general point of view, techniques like *XKey* were proven useful to identify taxonomies [DCFFCGGPP06].

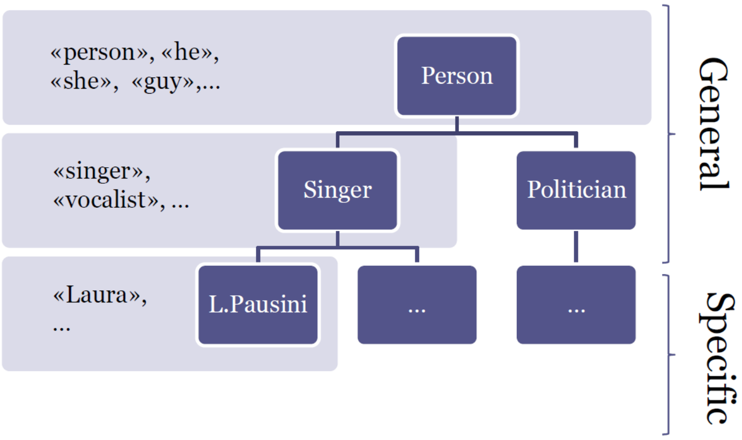


Figure iii.4: Meta-knowledge levels in the lexicon.

2.4.2 Linguistic Expressions

Each class in the lexicon (O, F, or V) has a set of associated linguistic expressions. The same expression can belong to several classes. Consequently, the following types of expressions are considered: *objective expressions* (not associated with valuations) that refer to objects or features, and *subjective* *expressions* (associated with valuations) that express a sentiment. Table III.1 shows examples of these types of expression. Furthermore, the expressions of a node are inherited by its child nodes, so that if *politician* is an expression for the focus POLITICIAN, it is also an expression for its child node PRESIDENT.

Type	Example Expressions	Class
Objective	*product*	focus PRODUCT
	costs	feature PRODUCT.PRICE
Subjective	*masterpiece*	valuation VERY_POSITIVE
	piece of junk	focus PRODUCT, valuation NEGATIVE
	a bit expensive	feature PRODUCT.PRICE, valuation NEGATIVE

Table III.1: Examples of subjective and objective expressions.

As previously mentioned, our lexicon not only maps single words, but also multi-word expressions. In addition, words in these expressions are not necessarily standard language in contrast to algorithms that extend lists of polarized seed words using WordNet.

As discussed in [QLBC09], maintaining a universal lexicon for all application domains is (practically) impossible. Attempting to exhaustively store all possible expressions that users could employ to name entities would be unrealistic, especially considering the fact that current issues are constantly changing, and new terms are being coined to refer to products, politicians, athletes, etc. Instead of trying to represent all this knowledge in a static container, our intention is to define an extendible and modularized model. To avoid dependence on the current situation, only abstract focuses are defined in the lexicon and particular focuses are defined in the lexicon extensions. Each concrete entity X defined in an extension is accompanied by an "extends Y" directive, indicating that X is a subclass of Y. In this way, the focuses BASKETBALL_PLAYER, and its child node POWER_FORWARD are defined in the general lexicon, and PAU_GASOL, offspring node of POWER_FORWARD, is defined in a basketball extension. Besides inheriting the expressions associated with the parent focus (e.g. *player* or *power forward*), new objective expressions, such as his name, and subjective expressions, such as his nicknames, should also be added. Moreover, these set of expressions could be enhanced semiautomatically by using some techniques such as [LMS06] to detect new objective expressions, [NPI11, BV04] to detect new subjective expressions, or [TW11a] to adapt the strength and polarity of subjective expressions from one domain to another.

With this model, the task of keeping the knowledge current becomes easier. For example, let us suppose that in politics, the opposing party Y wins the elections, and the political party X then becomes the opposition. In this case, the knowledge could be updated just by modifying the hierarchical relations between objects (Figure iii.5).

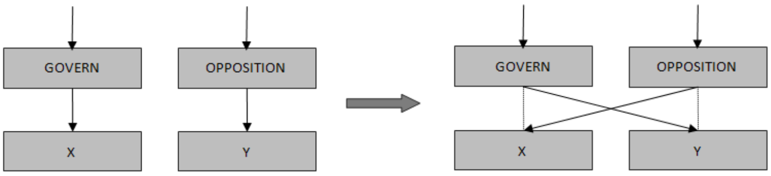


Figure iii.5: Example of a political scenario and knowledge maintenance.

Contextual valence shifters (e.g. negators, adverb intensifiers, etc.) play a key role in the improvement of the representation of the expressions. Those kind of linguistic modifiers are also supported on this study. Table III.2 briefly shows some examples of how linguistic patterns display a larger language. To access a vaster discussion on this mechanism [CC08b, DNML10] could be consulted.

Modifier	Examples	Rules
MOD_VERY	*very*, *absolutely*, *extremely*, ...	MOD_VERY POSITIVE → VERY_POSITIVE MOD_VERY NEGATIVE → VERY_NEGATIVE
MOD_LESS	*a little bit*, *relatively*, *not quite*, ...	MOD_LESS VERY_POSITIVE → POSITIVE MOD_LESS VERY_NEGATIVE → NEGATIVE
MOD_NEG	*not*, *nothing*, *not at all*, ...	MOD_NEG POSITIVE → NEGATIVE MOD_NEG NEGATIVE → POSITIVE MOD_NEG VERY_POSITIVE → NEUTRAL[a] MOD_NEG VERY_NEGATIVE → NEUTRAL
...	...	...

[a] For example *it is not perfect...*

Table III.2: Modifiers and linguistic rules.

2.5 Algorithm of Analysis

This section provides a detailed explanation of the analytical method. First, the preparation stage, which involves preprocessing and filtering comments, is described. Then we go on to explain how the lexicon is used to identify focuses and how the sentiment analysis module classifies them. Finally, the mining module and the interpretability of the results are described.

2.5.1 Preparation of the Knowledge

In this section, the preparation of the Knowledge Module is explained. This module includes a filtering stage and a preprocessing stage.

Filtering Stage The purpose of the Filter Module is to identify and automatically dismiss the noisy (offensive, non related or advertising) comments so that they will not influence the analysis. The following types of comments were regarded as inappropriate, and thus were filtered out:

Comments containing swear words In colloquial language, swear words are very common. This type of comment should be revised carefully since discarding them all would lead to an excessive loss of information and be detrimental to the analysis. For this reason, only comments containing potentially offensive words that quote other user's comment are deleted, because it is very likely that they are expressing an opinion about another user instead of about the news focuses.

Comments containing URLs Users generally insert links in their comments. Often, those comments have advertising purposes. Therefore, it is likely that they will be accompanied by positive terms that should not influence the analysis (*Sign in now at <advertising url> where* *you will find the best prices!*). The elimination of those advertising comments is a heuristic used in order to eliminate opinions on irrelevant topics.

Banned comments Certain media allows users to report inappropriate behaviour from other users. This filter automatically eliminates these comments.

Preprocessing Stage In the preprocessing stage, various NLP techniques are applied. First we use the PoS tagger [MS99] to improve the performance of the Stemmer. By stemming the comments and linguistic expressions stored in the Lexicon, generality is gained. Finally, the Splitter is applied to separate the sentences of the comments. The GPL Library FreeLing 2.2[5] was used to implement the preprocessing.

2.5.2 Opinion Focus Detection Module

In the Opinion Focus Detection Module, the lexicon detects which focuses are the object of valuations. To illustrate this, an example of a real comment on politics is now considered (Table III.3). To simplify tracking, this comment has not been preprocessed.

Interpretation Context Context is defined here as the set of focuses that disambiguate the interpretation of subjective expressions. For instance, “big” has a negative connotation if it is used to evaluate the feature SIZE of the object MOBILE_PHONE, but it has a positive connotation if it is used to express an opinion about the feature SIZE of the object TV. For this reason, delimiting the context (MOBILE_PHONE or TV in this example) is crucial. The context could be analysed from a comment-level (local context $\mathcal{C}_l$) or document-level (global context $\mathcal{C}_g$).

A focus is considered to be implicit if it does not explicitly appear in an opinion-bearing sentence. However, as previously mentioned, in our model, subjective expressions could also be attached to objects and features. There are cases when more than one focus is attached to the subjective expression. The use of information in the local context to disambiguate the focus is possible in most cases. In this section we discuss how the context model supports the sentiment analysis in an attempt to solve some ellipsis, anaphora, and disambiguation problems.

Firstly, the objects in the lexicon containing some expression that appears in the document (body or comments) are marked as initial candidates (Example 2.a) in order to initialize the interpretation context. It is very likely that this initial set may be too broad because of the ambiguity of certain expressions. To reduce the candidate set, two heuristics are applied: *disambiguation analysis* and *frequency analysis.*

Disambiguation Analysis The Disambiguation Analysis uses the implicit knowledge contained in the hierarchical structure of the lexicon to tackle with the named entities co-reference. For example, consider that the only named entities in a comment are the followings: PRODUCT, MOBILEPHONE, and IPHONE. Since they are hierarchically related in the lexicon, this analysis replaces PRODUCT and MOBILEPHONE references with IPHONE label, because they are probably a co-reference (hypernym) of IPHONE. Note that if other particular mobile phone were also referenced, then this heuristic could only replace PRODUCT with MOBILEPHONE since it is the only valid co-reference to all other related entities. It should be pointed out that, since replacing references modifies their frequency, it is possible for the Frequency analysis to discard new named entities.

[5] http://nlp.lsi.upc.edu/freeling/

More formally, assuming the set of all named entities in the comment to be $\mathcal{C}_l$, then the disambiguation analysis replaces a named entity e with its hyponym e' if e is in the set defined by Equation III.1, where $desc(x)$ denotes the set of all the descending focuses of x in the lexicon hierarchy.

$$\{e \in \mathcal{C}_l \mid \exists e' \in (desc(e) \cap \mathcal{C}_l) : (desc(e) \cap \mathcal{C}_l - \{e'\}) \subseteq desc(e')\} \tag{III.1}$$

Figure iii.6 may serve to clarify the explanation above. Note that this heuristic deals with the ambiguity problem since it helps to precise references to hypernyms terms.

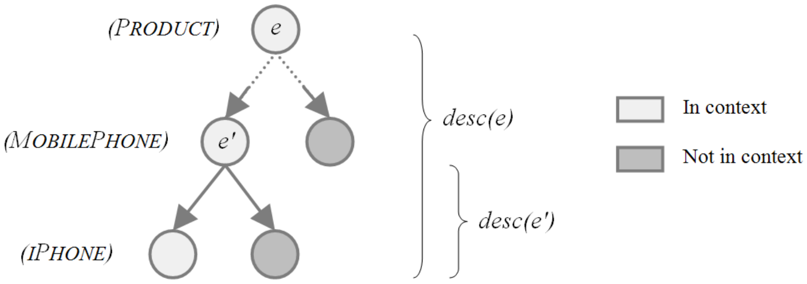


Figure iii.6: Disambiguation heuristic

Frequency Analysis Frequency analysis relies on the assumption that rarely referenced entities should not be taken into account to compute the global sentiment analysis summary. Low referenced entities could be caused by noisy linguistic interferences in the matching process of linguistic expressions. Furthermore, low referenced entities could be involved in user comments containing punctual arguments or exemplifications that actually lend support to an opinion on the main topics (for example named entity CAR in comment *It is shameful that even in crisis politicians have those private cars!*, concerning *Budget Deficit* issue). In any case, it does not make much sense to consider those entities in the final sentiment summary. Thus, those named entities e whose frequency of appearance in the document do not exceed a threshold are removed from the context (Equation III.2).

$$\mathcal{C} := \mathcal{C} - \{e \in \mathcal{C} : frec(e) \leqslant \gamma\} \tag{III.2}$$

In [RGG11] it is suggested that frequency-based methods fail while selecting relevant terms for feature mining. However, our heuristic does not contradict this research since it is combined with a disambiguation analysis heuristic that modifies the frequency of candidate features in context.

Simultaneous Analyses Unfortunately, this process is often affected by situations where the ambiguity is more complex and the previously mentioned analyses are not sufficient. For example, if POLITICIAN, PRESIDENT, and MEMBER_OPPOSITION are in the candidate set, POLITICIAN cannot be discarded because both PRESIDENT and MEMBER_OPPOSITION focuses are descendants of POLITICIAN in the lexicon. Accordingly, an expression referring to the POLITICIAN focus would be ambiguous since it could refer to either focus. However, as some of the focuses in the local context could have been deleted after applying the frequency analysis, the disambiguation analysis could possibly discard more focuses (Example 2.d).

Since both heuristics eliminate focuses, they perform simultaneously until there are no more changes to be made in $\mathcal{C}_l$. After that, features are searched by identifying objective expressions associated with any features of the object focus in $\mathcal{C}_l$ (Example 3.e). Finally, the global context $\mathcal{C}_g$ is created as the union of each local context.

Trace		$\mathcal{C}_l$
Comment: "...What's wrong with this man! Always the same movie. This nation needs urgently a change of president, someone with greater leadership capacity! Spain needs that its politicians improve the current situation and the opposition party does nothing!"		Ø
3.a Initial Focus Detection	*man* → PERSON *movie* → MOVIE *nation* → COUNTRY *president* → PRESIDENT *politicians* → POLITICIAN *Spain* → SPAIN *opposition party* → OPPOSITION	{PERSON, MOVIE, COUNTRY, PRESIDENT, POLITICIAN, SPAIN, OPPOSITION}
3.b Disambiguation Analysis	**ROOT/PERSON** **ROOT**/PRODUCT/ENTERTAINMENT/MOVIE **ROOT/LOCATION/COUNTRY** **ROOT/PERSON**/POLITICIAN/PRESIDENT **ROOT/PERSON**/POLITICIAN **ROOT/LOCATION/COUNTRY**/SPAIN **ROOT/PERSON**/POLITICIAN/OPPOSITION	{MOVIE, PRESIDENT, POLITICIAN, SPAIN, OPPOSITION}
3.c Frequency Analysis	**f(MOVIE)=Very low** f(PRESIDENT)=High f(POLITICIAN)=High f(SPAIN)=Medium **f(OPPOSITION)=Low**	{PRESIDENT, POLITICIAN, SPAIN}
3.d Disambiguation Analysis II	**ROOT/PERSON/POLITICIAN**/PRESIDENT **ROOT/PERSON/POLITICIAN** ROOT/LOCATION/COUNTRY/SPAIN	{PRESIDENT, SPAIN}
3.e Feature Discovery	*leadership capacity*→PRESIDENT.LEADERSHIP	{PRESIDENT, PRESIDENT. LEADERSHIP, SPAIN}

Table III.3: Detecting Local Context of a Comment.

As shall be seen in the next section, focuses that achieve a low reliability analysis are also deleted from $\mathcal{C}_g$ in the mining stage, and their presence in the results summary is thus avoided.

2.5.3 Sentiment Analysis Module

The Opinion Focus Detection module is meant to deal with ambiguities. How to deal with anaphora and ellipsis is still up to be done. The Sentiment Analysis module assigns a sentiment value to each focus in the interpretation context. To do this, it previously attempts to resolve some *anaphoric expressions*. Later, the feature-based analysis is carried out.

In order to analyse the overall sentiment of the comments and also assign sentiment expressions to opinion focuses, our algorithm follows five stages: (i) Expression Labelling; (ii) Tuples Extraction; (iii) Anaphora Resolution, (iv) Sentiment Calculation, and (v) Tuples Clustering and Filtering. Much of the current research on Sentiment Analysis focuses on document-level analysis and sentence-level analysis. However, since comments represent our basic information unit, we also perform an intermediate comment-level sentiment analysis.

To facilitate explanation, each stage is applied to a real comment, after the following global context has been previously detected: $\mathcal{C}_g$:={LA_LAKERS, NY_NICKS, TEAM, L_WALTON, L_ODOM, FORWARD, BASKETBALL_PLAYER, SPORTSMAN, DRIBBLING, PASSING}.

```
(s1) Things are breaking well for The Lakers.
(s2) They are fantastic!
(s3) Odom is a great forward, but I think that in dribbling
     and passing, Walton is the best.
```

Labelling Expression Stage In the Labelling Expression Stage, every expression in each comment that could refer to a focus of the global context is automatically identified and labelled. Taking the context and the valuation class V as a starting point, all the linguistic expressions associated with them in the lexicon are consulted. It should be underlined that each expression of a node is inherited by their offspring nodes. These expressions are used to detect and label references to the focuses in the comments. In a similar way, the valuations are marked. After labelling the sample comment, some expressions are marked with more than one label, such as *forward*, which can refer to the focuses FORWARD, L_WALTON or LAMAR_ODOM.

(s1) [Things are breaking well]$_{POSITIVE}$ for [The Lakers]$_{LA\_LAKERS}$.

(s2) They are [fantastic]$_{VERY\_POSITIVE}$!.

(s3) [Odom]$_{L\_ODOM}$ is a [great]$_{POSITIVE}$ [forward]$_{FORWARD\ or\ L\_ODOM\ or\ L\_WALTON}$, but I think that in [dribbling]$_{DRIBBLING}$ and [passing]$_{PASSING}$, [Walton]$_{L\_WALTON}$ is [the best]$_{VERY\_POSITIVE}$.

Tuples Extraction Stage In this stage, all of the analysis tuples for each sentence are computed. Such tuples atomically capture each user's opinion and are used for further mining. Before describing the process, tuples of analysis must be defined more formally.

An analysis tuple T is a tern $\langle o, f, v\rangle$ where $o \in O$ is an object satisfying $(o \in \mathcal{C}_g) \vee o = \emptyset$; $f \in F$ is a feature satisfying $(f \in (features(o) \cap \mathcal{C}_g)) \vee f = \emptyset$, and finally, $v \in V$ is a valuation that can also be null. Although other authors also consider parameters t (date of the opinion) and h (opinion holder, also called opinion sources) [WWC05], those parameters are irrelevant to our problem because users often report their comments on a date close to the news and tend to remain anonymous.

The algorithm is initialized by replacing each labelled expression with a tuple. When more than one label is possible, the most general one is selected (i.e. the one that corresponds to the first common predecessor). The distance that separates each pair of consecutive tuples, which is measured as the number of non-labelled words between them, is recorded. The third sentence of the example comment after initialization is shown below:

[⟨L_ODOM,∅,∅⟩ **2** ⟨∅,∅,POSITIVE⟩ **0** ⟨FORWARD,∅,∅⟩ **5** ⟨BASKETBALL_PLAYER, DRIBBLING,∅⟩ **1**
⟨BASKETBALL_PLAYER,PASSING,∅⟩ **0** ⟨L_WALTON,∅,∅⟩ **1** ⟨∅,∅,VERY_POSITIVE⟩]

Anaphora Resolution An anaphoric expression, also called here *free-valuation expression*, is an expressions referring to another one that is replaced by a deictic form. Ellipsis could be understood as one particular type of anaphora where the referred expression is not replaced but omitted. Anaphora resolution is a context-dependant task that implies the identification of the antecedent. On this study, we assume that each subjective expression evaluates certain object or feature that is already contained in the lexicon. Thus, we address the anaphora resolution problem by identifying previously mentioned entities in the discourse.

All sentences containing subjective expressions that do not contain any named or implicit entity are identified as anaphoric sentences. In other words, all tuples attached to an anaphoric sentence will be of the form ⟨ø,ø,v⟩, where $v \in V$ is a valuation. The last referred entity (if any) in previous sentences is considered the antecedent and a new tuple is added to the representation of the sentence. Only the three previous sentences are considered in the search.

Note *s2* is an anaphoric sentence in our previous example. Table III.4 shows how the tuples representation of the sentence is enhanced to resolve the anaphora.

Original Sentence	*s2*="*They* are fantastic!"
Extracted Tuples	*s2*≡[⟨ø,ø,VERY_POSITIVE⟩]
Previous Sentence	*s1*="Things are breaking well for The Lakers."
Antecedent's Tuples	*s1*≡[⟨ø,ø,VERY_POSITIVE⟩ ⟨LA_LAKERS,ø,ø⟩]
Antecedent	LA_LAKERS
Anaphora Resolution	*s2*:=[⟨ø,ø,VERY_POSITIVE⟩] ∪ [⟨LA_LAKERS,ø,ø⟩]

Table III.4: Example of anaphora resolution

Sentiment Calculation The algorithm joins the tuples by proximity. It follows the heuristic that one valuation is likely by referring to the nearest focus (Table III.5). The pseudo-distance -1 is used to indicate that two tuples were already joined. The process ends when all the distances between tuples are -1.

Where t_i is the ith tuple and $n_i = distance(t_i, t_{i+1})$ in the list of tuples L. In lines 2-3, the simple joining of focuses and valuations is performed. This union follows the heuristic that a valuation followed by a focus is likely to be a direct valuation of this focus. In lines 4-5, two close tuples are joined by verifying their hierarchical relationship. If they are hierarchically related, the most specific focus replaces the other. Where $t_i = \langle o_i, f_i, v_i \rangle$ and $t_{i+1} = \langle o_{i+1}, f_{i+1}, v_{i+1} \rangle$ are the input tuples, rule $U(t_i, n_i, t_{i+1})$ modifies $t_i \leftarrow \langle U_o(o_i, o_{i+1}), f_i, U_v(v_i, v_{i+1}) \rangle$ and $t_{i+1} \leftarrow \langle U_o(o_{i+1}, o_i), f_{i+1}, U_v(v_{i+1}, v_i) \rangle$ returning the sublist $[t_i, -1, t_{i+1}]$ where U_o represents the union of objects (Equation III.3) and U_v, the union of valuations (Equation III.4). After applying rule U, duplicate tuples are eliminated. In lines 6-8, valuations are propagated to the left and right through the already joined tuples in order to avoid the influence of the order in the result. Finally, the next pair of the nearest tuples is searched in line 9, and their distance is then subtracted from all other distances in lines 10-11. Table III.6 shows the trace of the algorithm applied to the previous sample array of tuples. In this way, the opinion expression *great* is assigned to Odom, and *the best* is assigned to Walton, also evaluating the features PASSING and DRIBBLING.

$$U_o(o_x, o_y) = \begin{cases} o_y \text{ iff } o_y \in desc(o_x)) \\ o_x \text{ in other case} \end{cases} \tag{III.3}$$

```
procedure Extraction($L = [t_0, n_0, t_1, n_1 ... t_{k-1}, n_{k-1}, t_k]$)
    do:
        foreach $\{t_i = \langle\langle ø,ø, v_i\rangle, n_i = 0, t_{i+1} = \langle f_{i+1}, p_{i+1},ø\rangle\}$
            $L \leftarrow t_0, ..., n_{i-1}, \langle f_{i+1}, p_{i+1}, v_i\rangle, n_{i+1}, ..., t_k$
        foreach $\{n_i = 0\}$
            $L \leftarrow t_0, ..., n_{i-1}, U(t_i, n_i, t_{i+1}), n_{i+1}, ..., t_k$
        while $\exists([...\langle f_i, p_i, v_i\rangle, -1, \langle f_{i+1}, p_{i+1}, v_{i+1}\rangle...])\ AND\ (v_i =ø\ XOR\ v_{i+1} =ø)$
            if($v_i =ø$) $v_i \leftarrow v_{i+1}$
            else $v_{i+1} \leftarrow v_i$
        $n_{min} := min_i\{n_i > 0\}$
        foreach $n_i \neq -1$
            $n_i \leftarrow n_i - n_{min}$
    while $\{\exists n_i > -1\}$
```

Table III.5: Tuples Extraction Algorithm

$$U_v(v_x, v_y) = \begin{cases} v_y \text{ iff } v_x =ø \\ v_x \text{ in other case} \end{cases} \tag{III.4}$$

Trace	**Explanation**
[⟨L_ODOM,ø,ø⟩ 2 ⟨**FORWARD,ø,POSITIVE**⟩ 5 ⟨BASKETBALL_PLAYER,DRIBBLING,ø⟩ 1 ⟨BASKETBALL_PLAYER,PASSING,ø⟩ 0 ⟨L_WALTON,ø,ø⟩ 1 ⟨ø,ø,VERY_POSITIVE⟩]	Simple U.
[⟨L_ODOM,ø,ø⟩ 2 ⟨FORWARD,ø,POSITIVE⟩ 5 ⟨BASKETBALL_PLAYER,DRIBBLING,ø⟩ 1 ⟨**L_WALTON,PASSING,ø**⟩ **-1** ⟨**L_WALTON,ø,ø**⟩ 1 ⟨ø,ø,VERY_POSITIVE⟩]	U
[⟨L_ODOM,ø,ø⟩ **1** ⟨FORWARD,ø,POSITIVE⟩ **4** ⟨BASKETBALL_PLAYER,DRIBBLING,ø⟩ **0** ⟨L_WALTON,PASSING,ø⟩ -1 ⟨L_WALTON,ø,ø⟩ **0** ⟨ø,ø,VERY_POSITIVE⟩]	Bring tuples 1 unit closer
[⟨L_ODOM,ø,ø⟩ 1 ⟨FORWARD,ø,POSITIVE⟩ 4 ⟨**L_WALTON,DRIBBLING,ø**⟩ **-1** ⟨**L_WALTON,PASSING,ø**⟩ -1 ⟨**L_WALTON,ø,VERY_POSITIVE**⟩**]**	Simple U. and U
[⟨L_ODOM,ø,ø⟩ 1 ⟨FORWARD,ø,POSITIVE⟩ 4 ⟨**L_WALTON,DRIBBLING, VERY_POSITIVE**⟩ **-1** ⟨**L_WALTON,PASSING,VERY_POSITIVE**⟩ **-1** ⟨**L_WALTON,ø,VERY_POSITIVE**⟩]	Valuation Propagation
...	...
[⟨L_ODOM,ø,POSITIVE⟩ -1 ⟨L_WALTON,DRIBBLING, VERY_POSITIVE⟩ -1 ⟨L_WALTON,PASSING,VERY_POSITIVE⟩ -1 ⟨L_WALTON,ø,VERY_POSITIVE⟩]	Final

Table III.6: Assignation of sentiment to focuses trace in s3.

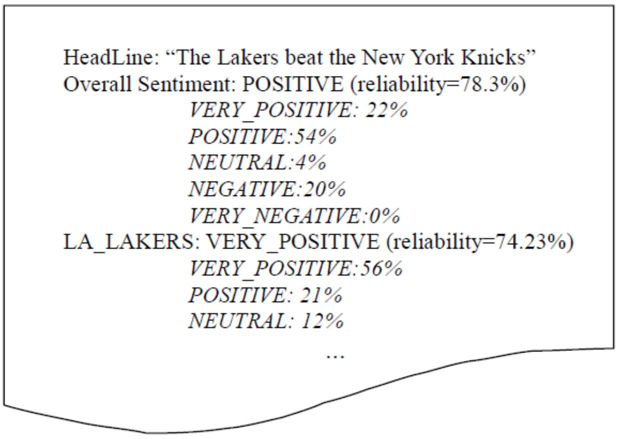
Tuples Clustering and Filtering Stage In this stage, all the focuses extracted for each sentence of a comment, are clustered, and a single set of tuples that abstractly represents user opinion is created.

In this process, the disambiguation analysis is applied again, but only considering the focus involved in the comment, namely, $\mathcal{C}_l$ instead of the entire context $\mathcal{C}_g$. For example, considering the tuples and extracted from s1 and s2, the result of this analysis replaces TEAM with LA_LAKERS due to the fact that it is not involved in the comment even though NY_NICKS is also in the global context. Finally, the result of applying the Sentiment Analysis to the sample is {⟨LA_LAKERS, Ø, POSITIVE⟩, ⟨LA_LAKERS, Ø, VERY_POSITIVE⟩, ⟨L_ODOM, Ø, POSITIVE⟩, ⟨L_WALTON, DRIBBLING, VERY_POSITIVE⟩, ⟨L_WALTON, PASSING, VERY_POSITIVE⟩, ⟨L_WALTON, Ø, VERY_POSITIVE⟩}.

2.5.4 Sentiment Mining Module

Once the set of tuples for each comment has been obtained, any mining technique is feasible. We compute the average sentiment for each focus involved in the comments. The general sentiment in the entire news is measured as an average of the specific opinions reported for each particular focus in the comments. To compute mean values, the following weightings are assigned to valuation labels: VERY_NEGATIVE:=-10, NEGATIVE:=-5, NEUTRAL:=0, POSITIVE:=5 y VERY_POSITIVE:=10.

Finally, a sentiment report is obtained (Figure iii.7). This report contains the overall sentiment, and the focus sentiment for each focus in the global context. Each feature sentiment is accompanied by a reliability measurement, obtained as the percentage of comments that have provided information related to this feature. Furthermore, not only the average sentiment is reported, but also the partial sentiments given to each possible valuation. This allows the study of polarity distributions in the users' opinion. As previously mentioned, in this stage, the latter focus filter is applied, and focuses whose reliability measurement is lower than 5% are discarded.



HeadLine: "The Lakers beat the New York Knicks"
Overall Sentiment: POSITIVE (reliability=78.3%)
VERY_POSITIVE: 22%
POSITIVE:54%
NEUTRAL:4%
NEGATIVE:20%
VERY_NEGATIVE:0%
LA_LAKERS: VERY_POSITIVE (reliability=74.23%)
VERY_POSITIVE:56%
POSITIVE: 21%
NEUTRAL: 12%
...

Figure iii.7: Fragment of sentiment report.

2.6 Evaluation Performance

This section presents the computational results obtained in the experiments that were performed to evaluate the *Opinium* system.

2.6.1 System Evaluation

Our system was evaluated by means of a broad test set of comments about current news items. The resulting reports were manually validated by 20 volunteers. They were requested to read the comments and then check the consistency of the reports obtained. It was found that in 87.6% of all cases, the results were fully consistent. In 7.4% of the cases the analysis was incomplete, and in the remaining 5%, the results were incorrect. More detailed information of this evaluation grouped by categories is shown in Table III.7. However, those results are not conclusive enough for the following reasons: (i) agreeing with the volunteers feedback, it is hard and costly for human to determine the analysis of general-domain sets of comments; (ii) since we intend to validate a whole system, those results remind insufficient.

	Sports	**Politics**	**Economy**	**Society**	**Entertainment**
Correct	87%	88%	93%	84%	86%
Incomplete	8%	9%	4%	7%	9%
Incorrect	5%	3%	3%	9%	5%

Table III.7: Experts Validation Summary

Since our system includes clearly differentiated stages, we propose a module comparison against some of the most representative sentiment analysis algorithms rather than a direct comparison by means of a black-box evaluation. Table III.8 summarizes the capabilities of our system and some of the most representative sentiment analysis approaches: PMI-IR [Tur02a], Sentiment Analyzer (SA) [YNBN03], FBS [HL04a], and SO-CAL [TBT+11].

Capabilities	Opinium	PMI-IR	SA	FBS	SO-CAL
Overall Sentiment	**Y**	Y	Y	Y	Y
Feature Mining	**Y**	N	Y[a]	Y	Y
Topic feature discovery	**Y**	N	Y	Y	N
Strength Measure	**Y**	Y	N	N	Y
Multidomain	**Y**	N	N	N	Y
Expert Supervision	**Y**	Y	N	Y[b]	Y

[a] Although the algorithm is able to perform Feature Mining, the lexicon is not available.

[b] A training dataset is required in CBA [LHM98] exclusively in the Topic features discovery module. It is not necessary for the other modules.

Table III.8: Comparison of the capabilities of the main related algorithms

Both our system and SO-CAL are lexicon-based approaches that require a considerable human effort to build the lexicon. This effort is essentially equivalent to providing annotated training resources in ML methods. According to [TBT+11], lexicon-based methods perform robustly in cross-domain scenarios, and can be easily enhanced with external sources of knowledge. There are methods for automatically adapting polarity lexicons to new domains [QLBC09]. In contrast, ML methods do usually require more effort to annotate new data resources.

2.6.2 Module Comparison

Our method is mainly composed of two modules (the Focus Detection Module and the Sentiment Analysis Module). In order to objectively validate our system, we designed a set of experiments that allowed us to evaluate each module separately (crystal-box evaluation) for news comments datasets. Additionally, the overall sentiment of the entire news corpus is also given so that it can be compared to the results of conventional Sentiment Analysis methods which only calculate overall polarity. Unfortunately, the properly annotated datasets are not available. Furthermore, a comparison against SO-CAL was not even considered since this method and ours are complementary. Indeed, we plan to mix up our lexicon with SO-CAL methodology in further research. Thus, we selected the following comparison algorithms and experiments for each module:

- **Focus Detection Module:**
  - **The Sentiment Analyzer** consists of three main modules: (i) candidate feature module; (ii) detection module; (iii) feature selection and Sentiment Analysis module. The last stage uses a lexicon that is unavailable. For this reason, only the candidate feature selection and the feature selection modules are considered in the comparison. The authors explain three methods of extracting noun phrases following different heuristic patterns (BNP base noun phrases, dBNP definite base noun phrases, and bBNP beginning definite base noun phrases). We considered the bBNP because it was demonstrated that this heuristic outperformed the others.
- **Sentiment Analysis Module applied to focuses (Focus SA):**
  - **FBS (Feature-Based Summarization)** uses the association miner CBA algorithm to obtain candidate features. The CBA classifier is not considered here because it requires a labelled training dataset of news. However, to obtain the sentiment in each feature, it applies an algorithm that exploits synonymy and antonymy relations in WordNet to expand the initial set of polarized words (seed adjectives). Then, the polarity of each feature is calculated.
- **Sentiment Analysis Module applied to the entire document (Overall SA):**
  - **PMI-IR (Pointwise Mutual Information - Information Retrieval)** is one of the most widely used algorithms in the literature. It is used to classify the polarity of the entire document.
  - **FBS** also classifies the entire document as positive, neutral, or negative.

The data The experiments were performed on a set of 500 current news items randomly selected from various news media. These news articles were manually labelled by 10 student volunteers who were unaware of the structure or the content in the Lexicon. For each news item, the following information was requested to the students to reflect user opinions:

- Classification of general user opinions in {VERY_NEGATIVE, NEGATIVE, NEUTRAL, POSITIVE, VERY_POSITIVE}.

- Set of the main discussion topics (focuses) in each news comments.
- Discussion topics (focuses) involved in each comment and the valuation that summarizes the user's opinion of each one.

The documents concerning the creation of the lexicon and the documents involved in the experimental evaluation were randomly extracted from the news media, 20 Minutos[6] and Mail Online[7] from 05/05/2010 to 06/07/2011. The news items chosen belong to a wide variety of categories (sports, politics, economics, society and culture) and each contains at least 50 comments (Table III.9). It should be underlined that none of these news items intervened in the creation of the Lexicon. Our lexicon built by means of the manual analysis of 2442 comments in 250 news, contains 380 objects (including 182 abstract focuses and 198 concrete focuses) and 128 features, with a total of 4762 linguistic expressions (1516 objective expressions and 3246 subjective expressions).

	Sports	**Politics**	**Economy**	**Society**	**Entertainment**
Percentage	27%	18%	13%	25%	17%
Comments Average	301.5	177.1	146.1	158.5	142.7

Table III.9: News distribution by kind and average number of user comments

Performance of the Experiments Using the labelled news as prototypes, the Focus Detection Module and the Sentiment Analysis Module were evaluated using the F-measure metric (Equation III.7). This metric is the weighted harmonic mean of precision (Equation III.5) and recall (Equation III.6), which are the most common metrics used in the literature. Precision is the fraction of the focuses detected that are considered relevant. Recall is the fraction of the focuses detected that are relevant to the labelled focuses.

$$Precision = \frac{\#(\{labeled\ focuses\} \cap \{retrieved\ focuses\})}{\#\{retrieved\ focuses\}} \tag{III.5}$$

$$Recall = \frac{\#(\{labeled\ focuses\} \cap \{retrieved\ focuses\})}{\#\{labeled\ focuses\}} \tag{III.6}$$

$$F - measure = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{III.7}$$

To evaluate the Overall Sentiment Analysis, the Accuracy metric is used, which considers the polarity of the prototype and the polarity calculated (Equation III.8). The accuracy of a measurement system is the degree of closeness of the measurements of a quantity to its actual (true) value.

$$Acc = \frac{\#\{succesfully\ labeled\ news\}}{\#\{news\}} \tag{III.8}$$

[6] www.20minutos.es

[7] http://www.dailymail.co.uk

Results of the study of the news datasets This section discusses the results obtained for each module and the comparison with the selected algorithms. Table III.10 shows the results obtained by our Focus Detection Module as compared to the Sentiment Analyzer. Table III.11 shows the results of our Sentiment Analysis Module in the Overall SA experiment in comparison to the PMI and FBS. Table III.12 shows the results of the Sentiment Analysis Module in the Focus SA as compared to the FBS. Finally, Figure iii.8 provides a summary of the comparison results obtained for each experiment.

	News																	
	Sports			**Politics**			**Economy**			**Society**			**Entert.**			**Total**		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Opinium	.89	.78	.84	.83	.82	.82	.94	.82	.88	.75	.71	.73	.69	.62	.65	.82	.75	.78
S.Analy.	.79	.62	.84	.70	.62	.66	.82	.73	.77	.68	.66	.67	.67	.67	.67	.73	.65	.69

Table III.10: Focus Detection results.

	News					
	Sports	**Politics**	**Economy**	**Society**	**Entert.**	**Total**
Opinium	.68	.98	.98	.96	.94	.89
PMI	.43	.99	.98	.92	.88	.81
FBS	.54	.51	.77	.76	.64	.64

Table III.11: Overall Sentiment Analysis results.

	News																	
	Sports			**Politics**			**Economy**			**Society**			**Entert.**			**Total**		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Opinium	.93	.63	.75	.80	.61	.70	.91	.72	.80	.80	.62	.70	.85	.60	.71	.86	.63	.72
FBS	.61	.33	.75	.59	.38	.46	.63	.42	.50	.56	.35	.43	.76	.37	.50	.62	.36	.45

Table III.12: Focus Sentiment Analysis results.

In the Focus Detection experiment, both the Sentiment Analyzer and our system achieved good results. The Sentiment Analyzer has the advantage of operating without any external knowledge. However, in view of the results, our structured lexicon, which is used to detect discussion topics, appeared to perform better. With respect to the overall sentiment experiment, results indicate that calculating the overall sentiment based on the particular sentiment of each discussion topic was a better approach to the news analysis problem than the performance of a global calculation as is the case of the PMI.

However, the fact that we were obliged to select comparison algorithms that were specifically designed for product reviews means that the results should be interpreted accordingly. Although the starting point of Sentiment Analysis is common to both approaches, the news to which the analysis was applied has a number of difficulties for which our algorithm is particularly robust. The multi-domain scenario motivates context detection and disambiguation processes as well as the use of colloquial language in news comments, all of which are examples of such difficulties. In this regard, we calculated that only 40.75% of the sentences in our news data set contained adjectives belonging to standard language. The same is also true of various other scenarios, such as debate forums, social networks, and blogs. This proliferation of colloquial language means that in such

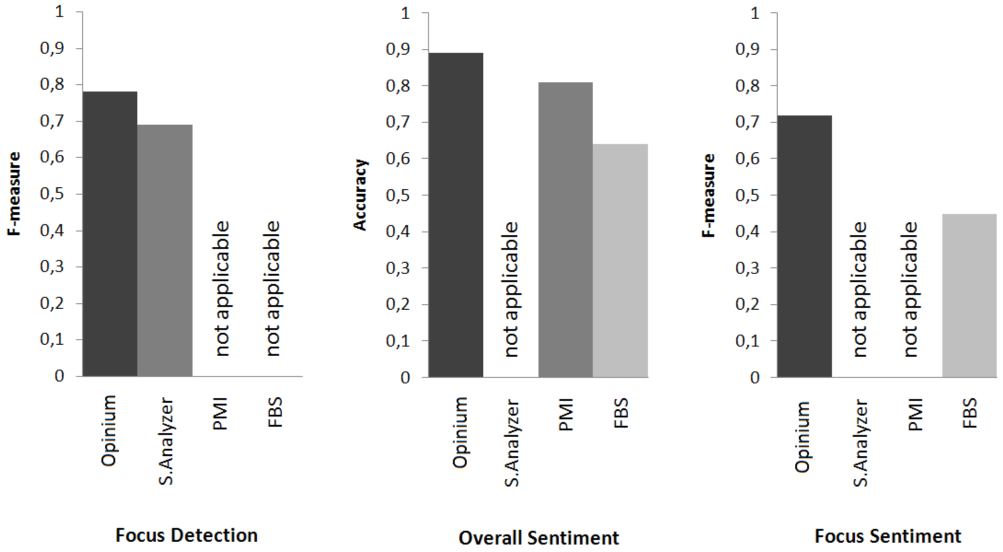


Figure iii.8: Result Summary Comparative.

scenarios, the FBS algorithm is problematic to apply since it would be forced to work without sufficient information. However, this difficulty is specific of this kind of user-generated content. It goes without saying that the validity of the FBS in product reviews has been tested and confirmed [HL04a].

We found a proportion of 0.6815 *free valuations* per comment. As commented above, a free valuation is considered to be any subjective expression not explicitly referring to any focus. In this regard, our algorithm was able to resolve 66.9% of these anaphoric valuations. These resolutions caused an improvement in the results of (about) 5%.

It should be underlined that the results shown were obtained after applying the Filtering Stage. We have computed that 7.561% of the comments were discarded. Of these, 57.248% correspond to comments containing swear words and 42.752% correspond to comments containing URLs. Since most of the discarded comments involved swear words (with negative sentiment), the influence of discarding them usually reminds in a positive variation of the overall sentiment score. Indeed, we have computed a positive variation of around 2.6% in our experiments.

2.7 Conclusions and Future Work

In this chapter, a complete lexicon-based system to the sentiment analysis of user comments on current news items has been proposed. Since these kind of systems work with unstructured data, they usually rely on a higher-level of meta-knowledge representation to perform the analysis.

In this regard, one of the goals of this study was to provide a well-structured knowledge model to support the analysis. Our lexicon is structured as a hierarchy of objects and features, and it is based on multi-word expressions. It contains two types of knowledge sources: *generic entities* and *domain extensions*. Since knowledge is modularized, it can be manually maintained just by adding or modifying some of their objects, features, or hierarchical relations.

Furthermore, hierarchical structure of the lexicon provides implicit information that could be useful to resolve certain ambiguities and ellipsis. We took advantage of frequency analysis and semantic information in the lexicon to keep track of context. Handling context allowed us to address the antecedent resolution in the anaphora problem and to resolve certain ambiguities. As a result, feature-based sentiment analysis became more reliable.

NL presents several particularities that difficult automatic processes to extract the sentiment of opinions, such as its multi-focus scenario or the use of colloquial language. Our system includes a Focus Detection Module that is able to identify the main discussion topics. It also contains a Sentiment Analysis Module which is able to analyse the strength and sentiment of the entire news article as well as of each of its focuses. Although the lexicon that we used is specifically designed to deal with news comments, the knowledge in our system is highly modularized. By properly adapting the knowledge, the system could carry out Sentiment Analysis in other domains.

The system has obtained promising results in experimental validation. Given that our system includes various modules, we designed specific experiments to compare each module with some of the most relevant algorithms in Sentiment Analysis. The results obtained in these experiments showed that our approach performed better in scenarios where colloquial language predominates.

Although results are promising, there is much work still to be done. Exploiting semantic relations through WordNet has proven to be a successful heuristic for automatically expanding opinion word sets. In this line, in future work, we plan to study the automatic expansion of subjective expressions of our lexicon, or even the automatic detection of new entities through web mining. We are also interested in investigating techniques that take advantage of a simpler meta-knowledge model in order to infer a taxonomical model, along the lines of [DCFFCGGPP06], that is tuned for the document corpora.

There is no doubt that Shallow Language approaches lead to lighter analyses than NLP or other techniques, and their effectiveness has been proven in several fields. However, these techniques are not able to detect shades of meaning, such as irony or sarcasm. For this reason, it is difficult to accurately calculate sentiment in such cases. In further research, our aim is to study these shades of meaning, and thus improve our Sentiment Analysis Module.

3 Final Discussions: Applications and Future Work

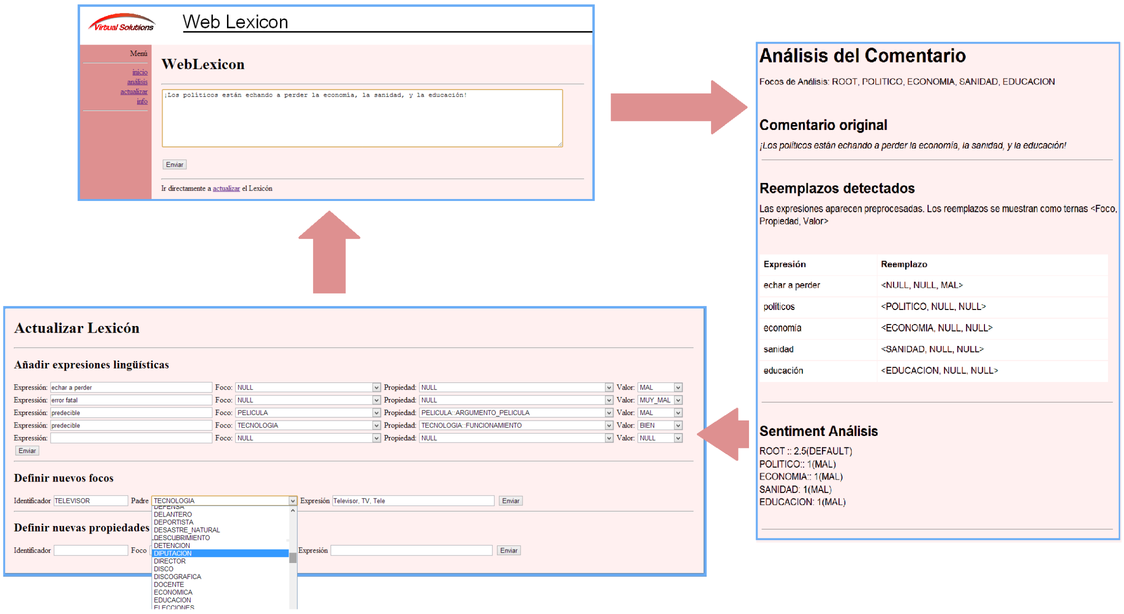
This section is to offer some final remarks on different closed-domains NL approaches to unstructured knowledge exploiting meta-domain resources that were developed by our researching group. Thus, it should be pointed out that the author of this dissertation is not the main author of these applications, but has actively contributed to their development.

We first present a tool for creating and debugging lexicon resources for Sentiment Analysis in section 3.1. Later, we present a working project aimed for integrating our SA framework into social networks in section 3.2. A domain extractor utility is briefly explained in section 3.3. Some final remarks and future work are finally drawn in section 3.4.

3.1 Web-Lexicon: A Tool for Enhancing and Debugging Lexicons

How to properly design sentiment lexicons has received a great deal of attention from the fields of text classification and computational linguistic communities. In this regard, both generic lexicons [ES06a, NPI11] and techniques to expand domain-specific lexicons [QLBC09, ALSZ06, ALM^{+}03] have been proposed. However, since news are inherently affected by the continuous appearance of new terms and concepts, there is usually the case that one does not count with enough training data at his/her disposal. Thus manually adding some new focuses and its linguistic expressions could be a suitable solution. In this section, we briefly present a tool for manually enhancing and debugging lexical resources for sentiment analysis.

This tool, called *Web-Lexicon* allow the expert in charge of customizing the lexicon to insert a comment, obtain its sentiment analysis, and modify the lexicon to debug the analysis if needed (Figure iii.9). Modifications imply adding new linguistic expressions (objective or subjective), adding new objects, or new properties.



WebLexicon

¡Los políticos están echando a perder la economía, la sanidad, y la educación!

Ir directamente a actualizar el Lexicón

Análisis del Comentario

Focos de Análisis: ROOT, POLITICO, ECONOMIA, SANIDAD, EDUCACION

Comentario original

¡Los políticos están echando a perder la economía, la sanidad, y la educación!

Reemplazos detectados

Las expresiones aparecen preprocesadas. Los reemplazos se muestran como ternas <Foco, Propiedad, Valor>

Expresión	Reemplazo
echar a perder	<NULL, NULL, MAL>
políticos	<POLITICO, NULL, NULL>
economía	<ECONOMIA, NULL, NULL>
sanidad	<SANIDAD, NULL, NULL>
educación	<EDUCACION, NULL, NULL>

Sentiment Análisis

ROOT :: 2.5(DEFAULT)
POLITICO:: 1(MAL)
ECONOMIA:: 1(MAL)
SANIDAD: 1(MAL)
EDUCACION: 1(MAL)

Actualizar Lexicón

Añadir expresiones lingüísticas

Expressión:	echar a perder	Foco:	NULL	Propiedad:	NULL	Valor:	MAL
Expressión:	error fatal	Foco:	NULL	Propiedad:	NULL	Valor:	MUY_MAL
Expressión:	predecible	Foco:	PELICULA	Propiedad:	PELICULA::ARGUMENTO_PELICULA	Valor:	MAL
Expressión:	predecible	Foco:	TECNOLOGIA	Propiedad:	TECNOLOGIA::FUNCIONAMIENTO	Valor:	BIEN
Expressión:		Foco:	NULL	Propiedad:	NULL	Valor:	NULL

Definir nuevos focos

Identificador TELEVISOR Padre TECNOLOGIA Expresión Televisor, TV, Tele

Definir nuevas propiedades

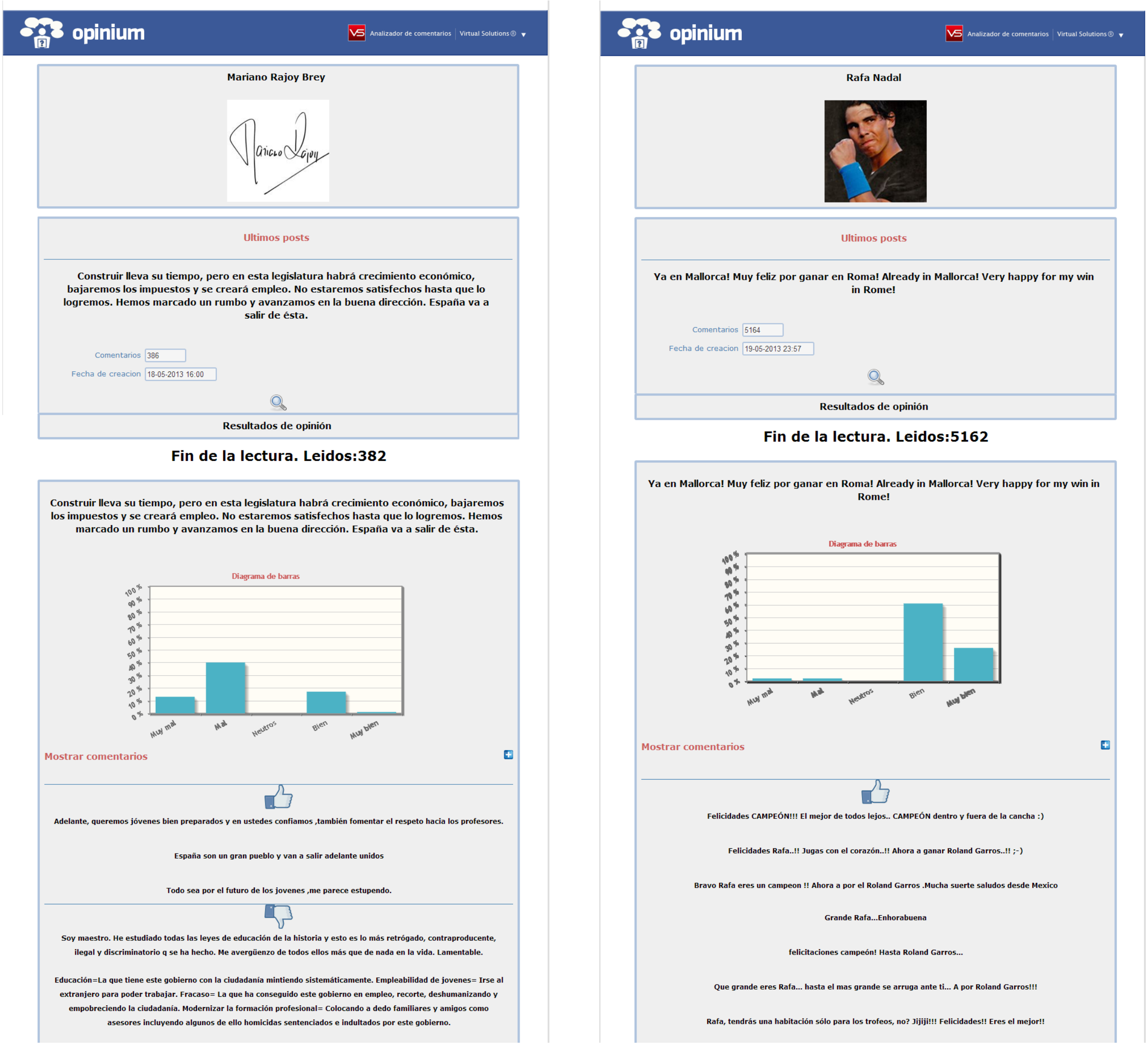
Identificador Foco Expresión

Figure iii.9: Web-Lexicon functionality

3.2 Opinium on Facebook

There is currently a commercial version of *Opinium* being developed by a private company. This application will allow users to automatically obtain a sentiment summary on their own posts. Arguably, the most interesting feature is to automatically analyse posts of public figures, that often gather comments of the order of hundreds or tons.

Figure iii.10 shows an example of the sentiment analysis of two well-known public figures: a politician and a tennis player. The amount of comments pertaining to each different sentiment label is breakdown in a bar-char. After that, the most positive and negative comments are highlighted.



Mariano Rajoy Brey

Ultimos posts

Construir lleva su tiempo, pero en esta legislatura habrá crecimiento económico, bajaremos los impuestos y se creará empleo. No estaremos satisfechos hasta que lo logremos. Hemos marcado un rumbo y avanzamos en la buena dirección. España va a salir de ésta.

Comentarios 386
Fecha de creacion 18-05-2013 16:00

Resultados de opinión

Fin de la lectura. Leidos:382

Construir lleva su tiempo, pero en esta legislatura habrá crecimiento económico, bajaremos los impuestos y se creará empleo. No estaremos satisfechos hasta que lo logremos. Hemos marcado un rumbo y avanzamos en la buena dirección. España va a salir de ésta.

Mostrar comentarios

Adelante, queremos jóvenes bien preparados y en ustedes confiamos ,también fomentar el respeto hacia los profesores.

España son un gran pueblo y van a salir adelante unidos

Todo sea por el futuro de los jovenes ,me parece estupendo.

Soy maestro. He estudiado todas las leyes de educación de la historia y esto es lo más retrógado, contraproducente, ilegal y discriminatorio q se ha hecho. Me avergüenzo de todos ellos más que de nada en la vida. Lamentable.

Educación=La que tiene este gobierno con la ciudadanía mintiendo sistemáticamente. Empleabilidad de jovenes= Irse al extranjero para poder trabajar. Fracaso= La que ha conseguido este gobierno en empleo, recorte, deshumanizando y empobreciendo la ciudadanía. Modernizar la formación profesional= Colocando a dedo familiares y amigos como asesores incluyendo algunos de ello homicidas sentenciados e indultados por este gobierno.

Rafa Nadal

Ultimos posts

Ya en Mallorca! Muy feliz por ganar en Roma! Already in Mallorca! Very happy for my win in Rome!

Comentarios 5164
Fecha de creacion 19-05-2013 23:57

Resultados de opinión

Fin de la lectura. Leidos:5162

Ya en Mallorca! Muy feliz por ganar en Roma! Already in Mallorca! Very happy for my win in Rome!

Mostrar comentarios

Felicidades CAMPEÓN!!! El mejor de todos lejos.. CAMPEÓN dentro y fuera de la cancha :)

Felicidades Rafa..!! Jugas con el corazón..!! Ahora a ganar Roland Garros..!! ;-)

Bravo Rafa eres un campeon !! Ahora a por el Roland Garros .Mucha suerte saludos desde Mexico

Grande Rafa...Enhorabuena

felicitaciones campeón! Hasta Roland Garros...

Que grande eres Rafa... hasta el mas grande se arruga ante ti... A por Roland Garros!!!

Rafa, tendrás una habitación sólo para los trofeos, no? Jijiji!!! Felicidades!! Eres el mejor!!

Figure iii.10: Opinium on Facebook, an example

3.3 Domain KeyWord Extraction

In [RMCZ12] we presented a new method for extracting key-terms in the so-called *support documents* (SD) —tutorials, how-to-guides, walkthroughs, or FAQ files. Automatic Keyword Extraction (AKE) methods are proven useful to support text classification tasks, query searches, or maintenance of collections of texts. SDs are text documents referring to specific issues, usually of limited or private diffusion. This method treated separately two different sort of relevant terms: domain-specific and multi-domain terms. The former refers to proper nouns or technical terms closely related to specific aspects of the domain. In contrast, multi-domain terms are those presenting a shared meaning through different domains, that is, are of common knowledge outside the boundaries of the domain at hand.

On the one side, domain-specific terms were detected through a frequency analysis, devoted to contrast the frequency in language with the frequency of each term appearing in the document. On the other side, the system identified multi-domain terms in the collection by exploiting the free-online encyclopaedia Wikipedia[8]. Thus, Wikipedia was considered to be a meta-knowledge resource of the public knowledge.

Each Wikipedia article describes a complete entity, called *concept* according to the Wikipedia terminology. In turn, each concept is compound by an identifier name, a set of alternative synonymous, and the category to which it belongs. Thus, Wikipedia could be regarded as a hierarchical meta-model resource. Thank to this rich meta resource we could create a controlled dictionary of terms that allowed us to take advantage of the well-known metric called *key-phraseness*, the disambiguation procedures, and its hierarchical information.

Although empirical results were reported using FAQ lists as support documents, the truth is that the method does not take into account any particular structure of the underlying knowledge, so it could certainly be classified among the UK/PM approaches.

3.4 Final Remarks

The UK/PM is arguably the most recommended configuration to perform some kind of advanced analysis on texts. Since the input documents do not present any sort of inner structure, these techniques could take advantage of modelling the domain knowledge in advance. This analysis is aimed for preparing the necessary meta-knowledge resources to improve the subsequent analysis of text items. This meta-knowledge is usually specified in terms of concepts hierarchies and relations, along with the lexical resources that gain access to it.

This chapter has covered some important related aspects including the knowledge modularization (general versus specific entities), the identification of the main focuses of a text, or the effect of valence shifters in language. We have concluded the chapter offering a brief summary on different UK/PM approaches including an application to enhance and debug lexicons, an application to analyse the sentiment of opinions in social networks, and a method to extract the most important key-concepts in a document.

Albeit one of the most interesting challenges to UK/PM approaches concerns with how to create or enhance proper knowledge resources, we have addressed the problem of *delimiting the context* in news items in order to keep better track of opinions targets in the so-called sentiment analysis problem. In doing this, we have paid special attention to some concrete domain-dependant NL problems, including *anaphora*, *ellipsis*, and *ambiguity* phenomena. It is however worth mentioning that methods to improve the meta-domain knowledge will be later proposed in chapter V. But before that, we plan to study a different knowledge configuration: those systems that, on the contrary, count with structured data but there is no meta-knowledge at its disposal. Thus, the next point to deal with in this dissertation concerns with SK/AM approaches (chapter IV).

[8] `http://www.en.wikipedia.org/`

Chapter IV

Collaborative NL approaches: Structured Knowledge, Absence of Meta-knowledge

1 Introduction

The automatic information recovery has recently gained an increasing interest [DW07, SC07, GZ09a] due to the continuous information needs of the current society and the new advances in communication. Concretely, these technologies lure companies and organizations that invert considerable effort and money to offer high-quality services. Let us imagine a system where all the information is individually stored in Information Units (IU) which are simple enough as to be directly interpreted by non-technological people. Let us also assume the system is incremental, that is, the knowledge could be naturally enhanced just by adding new IUs, and this new knowledge becomes now accessible to all users. The inherent knowledge structure establishes then a meeting point between knowledge managers and information consumers. These kind of systems would be of the utmost interest in fields such as *e-Learning*, or *Collaborative Environments* [MRPVR07, AHBLGS$^+$12, DSS$^+$02] where knowledge is generated and consumed collaboratively. This chapter is devoted to closed-domain NL collaborative systems, where knowledge is structured and there is no additional meta-knowledge source available (SK/AM). This feature, far from being a drawback, may rather avoid technological barriers and constraints to users that are no longer expected to be experts in NLP or information technologies either.

Our goal is to approach certain peculiar difficulties that arise in collaborative systems through the study of an specific open-problem. The Internet is usually regarded to as a means where to look up any sort of information. In this respect, Question Answering appears as an extremely promising and ambitious field motivated by providing answers to NL questions in an open-domain perspective, that is, on any possible topic. However, it is often the case that users are interested in a specific knowledge area —such as programming languages. If so, they might rather rely on specific trusted forums —such as *Stackoverflow*[1]— to post their questions, waiting for another users to answer it. In turn, each time an answer is given, it becomes potentially useful for other users. Knowledge, thus, grows in a collaborative and controlled manner. How to efficiently access and manage knowledge in

[1] `http://stackoverflow.com/`

such systems becomes an interesting challenge.

We selected the so-called *FAQ retrieval* problem in this case as a representative collaborative scenario reflecting most important related aspects. Typically, a FAQ retrieval system is aimed for recovering the most related questions/answers pairs to a given NL question. We will face the problem from a *query log* perspective, that is, each FAQ entry is enhanced by a set of NL related queries collected during the life of the system. Query logs associated to each FAQ entry are processed to compose a set of linguistic patterns, aka *templates*, that are later used to match users questions. Loosely speaking, these linguistic patterns are composed by the most relevant words in the original questions. Thus, the problem could be stated as how to automatically decide which are the most relevant words to a question with respect to a given domain. Note that this problem bears strong resemblance to the Feature Selection for Text Classification problem, to which certain preliminary indications where already drawn in chapter II. In particular, we observed that (i) all classes should be equally attended while searching for relevant words, (ii) positive correlation is arguably the best indicator of term relevancy to a category, and (iii) learners could be effectively trained even from extremely-reduced term spaces. The present chapter is to offer a continued study under a different knowledge configuration.

Consequently, the SK/AM configuration suggests there is an effective mechanism to represent each IU. More concretely, we are interested in tackling the following partial objectives:

- Scalability: designing a framework to facilitate users add new IUs. The retrieval algorithm should be able to assimilate these new IUs on the fly (section 2).
- FAQ management: the system should incorporate analytic tools to discover knowledge gaps, in order to support managers in improving their FAQs quality (section 3).
- Interpretability: we will investigate how the system could be manually supervised in critical systems in order to improve its performance. Because this may entail certain human effort, methods alleviating as much as possible the associated costs will be proposed (section 4).

Before going in depth in the above mentioned issues, we will formally define the FAQ retrieval problem (section 1.1). Later, we offer a brief description of WordNet —a thesaurus-based tool for NLP (section 1.2). After that, we present the common notation to deal with regular expressions, the formal mechanism we are using here to tackle the problem (section 1.3). Section 1.4 is dedicated to formally demonstrate that certain desirable properties are actually achievable from a computational point of view. We conclude this chapter offering some final remarks including some related working projects and our proposals for future research (section 5).

1.1 FAQ retrieval: Problem Statement

With the aim of overcoming the costs of call centres, companies usually try to anticipate most typical customer's questions and answer them in advance. Those users questions and expert answers about specific domains are often collected and organized in Frequently Asked Questions (FAQs) lists as an alternative Customers Relationship Management (CRM) strategy.

Often, FAQ retrieval approaches require some kind of knowledge modelling. Thus, FAQ managers are requested to define templates [Sne99], ontologies [LFQL11, LLL10, HSCD09], keywords [Whi95], Information Entities [LHK98], or so on. As those systems imply many expert-knowledge requirements, the management of the FAQs remains out of reach to non-technical people. In order to override this restriction, methods that do not require artificial domain-specific modelling have been proposed. Those systems are primarily statistical [BCC$^+$00, KS08a, KS06] and usually operate computing similarity measures. Our definition of the problem falls along the lines of these techniques that do not require any external knowledge resource. Although a FAQ is traditionally considered to be a list of Question/Answer pairs, we will also consider a set of reformulations —aka query logs— associated to each question in order to improve the lexical variety of each FAQ entry.

More formally, let Σ be a fixed finite alphabet of symbols. Σ^* denotes the set of all possible finite strings (also called *words*) over Σ, including the empty string ε. Thus a FAQ F with n entries is represented by $F = \{F_1, F_2, \ldots, F_n\}$, where F_i is the ith FAQ entry in F. The ith entry will be represented as $F_i = (Q_i, A_i)$, where $Q_i = \{S_i^0, \ldots, S_i^k\}$ is the reformulation set of the ith entry. S_i^0 is the original question, directly extracted from the initial FAQ, and S_i^j, $j = 1, \ldots, k$ are the k linguistic reformulations collected from users questions during the life of the system. Finally A_i is the answer given to the ith entry of the FAQ. For the sake of simplicity, we will use $S \in F_i$ to denote S is one of the reformulations in Q_i. Each question is a sentence formed by a sorted list of words $S_i^j = [w_1, w_2, \ldots, w_\ell]$, where $w_i \in \Sigma$. Finally, we will call $\mathcal{F}$ to the abstract set of all possible FAQs.

Given an user question $U = [w_1, w_2, \ldots, w_u]$, the FAQ retrieval problem could be stated as how to define a FAQ retrieval engine that implements a calculable procedure R defined by:

$$R : finite(\Sigma^*) \times finite(\mathcal{F}) \longrightarrow \pi(\mathcal{F}) \tag{IV.1}$$

Where R is the function that computes an enumeration, here noted as π, of the FAQ entries reflecting their relative relevance w.r.t. the user question. As will be seen, predicate $finite$ restricts the applicability of the function only to finite sets in order to make it become calculable.

1.2 WordNet

WordNet is a publicly available[2] lexical resource broadly use by a huge number of NLP systems (such was the case that citing here just a few would be thus uninformative). Albeit it was defined by the English language, it has been adapted to many languages in the so-called EuroWordNet project[3].

WordNet collects nouns, verbs, adjectives, and adverbs under the concept of *synsets* —numeric codes that univocally identify different synonymous sets of words. These synsets are linked according to different semantic relations conforming thus a network of meaningfully related work. This network allows NLP techniques to implement rich procedures exploiting distances between words.

WordNet contains 117000 synsets that are linked to other synsets through conceptual relations. Additionally, a brief definition of each synstet, called *gloss* serves to briefly describe the synonym set. Most important relations among synsets include *hyperonymy* and *hyponymy* (also called IS-A relations). Moreover, common nouns and specific instances are explicitly differentiated in WordNet. In addition to IS-A relations, PART-OF relations are also represented in the net. Finally, adjectives are also linked through antonymy relations, that is, opposite polarity in the semantic of adjectives is also reflected in the net.

[2] `http://wordnet.princeton.edu/`

[3] `http://www.illc.uva.nl/EuroWordNet/`

1.3 Regular Expressions: notation

A regular expression, first defined by Stephen Cole Kleene in [Kle56], is an automata that identifies a regular set. R_Σ is the set of all regular expressions over Σ. The constant $\emptyset$ represents the empty regular set. Each symbol $a \in \Sigma$ is a well-formed regular expression. Let α and β be two arbitrary well-formed regular expressions, in this research we note disjunction as $\alpha|\beta$ instead of $\alpha + \beta$, and concatenation as $\alpha\beta$ instead of $\alpha.\beta$. The Kleene star α^* denotes zero or more repetitions and α^+ denotes one or more repetitions. $\alpha?$ represents the optional regular expression. In this chapter, we use $*$ as a shorthand for Σ^*, and parentheses for grouping. E is an arbitrary set of regular expressions. Finally, the language accepted by a given regular expression α is denoted as $L(\alpha)$ and is expressed as a regular set:

$$
\begin{aligned}
&L(\emptyset) = \emptyset \\
&L(\varepsilon) = \{\varepsilon\} \\
&L(\alpha \,|\, \beta) = L(\alpha) \cup L(\beta) \\
&L(\alpha\, \beta) = \{u\, v \,|\, u \in L(\alpha), v \in L(\beta)\} \\
&L(\alpha^+) = \{r_1 r_2 \cdots r_n \,|\, n \geq 1, r_i \in L(\alpha)\} \\
&L(\alpha^*) = L(\alpha^+) \cup \varepsilon \\
&L(\alpha?) = L(\alpha) \cup \varepsilon \\
&L(*) = \Sigma^* \\
&L(E) = \textstyle\bigcup_{r \in E} L(r)
\end{aligned}
$$

We will use interchangeably the terms *template* and *regex* to refer to regular expressions.

1.4 Theoretic Framework

This section is devoted to deal with the problem of FAQ retrieval from a theoretical point of view. Our aim is to prove that a suitable base of templates satisfying certain desirable properties is actually achievable to differentiate all FAQ entries, provided that the sample training questions comply with certain restrictions.

Recognizer. The Recognizer Π is the function:

$$\Pi : 2^{R_\Sigma} \times 2^{\Sigma^*} \longrightarrow 2^{\Sigma^*} \tag{IV.2}$$

Defined by:

$$\Pi(E, A) := \{s \in A \,|\, \exists r \in E, s \in L(r)\} \tag{IV.3}$$

For all $E \subseteq R_\Sigma$, $A \subseteq \Sigma^*$, $\Pi(E, A)$ is the subset of words in A accepted by any regular expression in E.

Definitions: Let $E \subseteq R_\Sigma$ be a set of regular expressions, and $A, B \subseteq \Sigma^*$ two sets of words. We define the following predicates:

1. $E\ accepts\ A \longleftrightarrow A \subseteq L(E)$
2. $E\ excludes\ A \longleftrightarrow A \cap L(E) = \emptyset$
3. $E\ discriminates\ (A, B) \longleftrightarrow (E\ accepts\ A) \wedge\ (E\ excludes\ B)$
4. $A\ justifies\ E \longleftrightarrow \forall r \in E, \exists s \in A$ such that $s \in L(r)$

That is, (i) each word in A is in the language accepted by some regex in E, (ii) none of the words in A is in the language accepted by any of the regex in E, (iii) words in A are *discriminated* from words in B if the regular language accepted by E accepts A and excludes B, and (iv) membership to the set E is *justified* by A if every regex in E accepts at least one word in A. These predicates lead us to define formally the concept of learner machine.

Proposition 1: If $A \cap B = \emptyset$ then $\exists\, E \subseteq R_\Sigma$ such that $E\, discriminates\, (A, B)$

Proof 1: To demonstrate proposition 1, we will consider two calculable functions P_1 and P_2.

For an arbitrary word $s = w_1 w_2 \cdots w_n$, $w_i \in \Sigma$, the function $P_1 : \Sigma^* \longrightarrow R_\Sigma$ calculates the trivial regular expression $r = w_1.w_2.\cdots.w_n$, $r \in R_\Sigma$, as a concatenation of symbols in s. The regular set defined by this regex is a singleton satisfying $L(r) = \{s\}$.

For an arbitrary set of words $A \subseteq \Sigma^*$, P_2 is the function $P_2 : 2^{\Sigma^*} \longrightarrow 2^{R_\Sigma}$ defined by $P_2(A) := \{P_1(s), s \in A\}$. This function is calculable if A is finite.

1. From definitions of functions P_1 and P_2 follows $P_2(A)$ *accepts* A
2. Since each trivial regex in $P_2(A)$ accepts one and only one word, which is in A, and since $A \cap B = \emptyset$, follows $P_2(A)$ *excludes* B.
3. From (i,ii) follows $P_2(A)\, discriminates(A, B)$ □

Proposition 2: If $A \cap B = \emptyset$ then $\exists\, E \subseteq R_\Sigma$ such that $E\, discrimines\, (A, B)$ and $A\, justifies\, E$

Proof 2: From (proof 1) and considering that if $r \in P_2(A)$ then, by definition, $r = P_1(s)$ where $s \in A$ and $s \in L(r)$, follows $P_2(A)\, discriminates(A, B)$ and $A\, justifies\, P_2(A)$ □

Definition 1: Learner Machine. According to Gold's learning paradigm, we will define a learner machine as a procedure that calculates a set of regular expressions covering all (finite[4]) positive examples $I^+ \subseteq \Sigma^*$, and not covering any of the negative ones $I^- \subseteq \Sigma^*$. More formally, being $finite(\Sigma^*)$ the notation for all finite sets of finite words in Σ^*, a learner machine will be a procedure that calculates a function

$$M : finite(\Sigma^*) \times 2^{\Sigma^*} \longrightarrow 2^{R_\Sigma} \tag{IV.4}$$

Satisfying the following restrictions

$$\forall\, I^+, I^- \subseteq \Sigma^*; \quad M(I^+, I^-)\ discriminates\ (I^+, I^-) \tag{IV.5}$$

[4] Although Gold defines its learning paradigm for an infinite succession of examples, the learner is defined by each step i, where exactly i positive examples have been presented.

$$\forall\, I^+, I^- \subseteq \Sigma^*; \quad I^+\; justifies\; M(I^+, I^-) \tag{IV.6}$$

We will use $I = I^+ \cup I^-$ to refer to the entire set of examples. For the sake of simplicity, we will use M to refer to a learner machine and the function that it calculates. We say that $M(I^+, I^-)$ is the set of templates calculated by M for I^+ with respect to I.

It should be remarked that our learner could obtain various regular expressions to satisfy the above restriction. Usually, classical approaches are requested to obtain only one regex. However, since all obtained regular expressions could be joined using the disjunction operation, this consideration does not diminish the generality of the problem.

Property: Correctness. Given two set of words such that $A \subset B \subseteq \Sigma^*$, and a set of templates $E \subseteq R_\Sigma$, we will say that E *correctly* accepts A with respect to the broader set B *iff* the recognizer Π calculates exactly A as the subset of words in B accepted by E.

$$E\, correct_{A\, wrt\, B} \longleftrightarrow \Pi(E, B) = A \tag{IV.7}$$

Proposition 3: There exists a learner machine M_1 such that $\forall\, A, B \subseteq \Sigma^*$ with $A \subset B$, and being A finite, $M_1(A, B - A)$ calculates a set of templates that *correctly* accepts A with respect to B.

Proof 3: We define $M_1(X, Y) := P_2(X)$ for any $X, Y \subseteq \Sigma^*$.

1. If $A, B \subseteq \Sigma^*$ with $A \subset B$, follows $A \cap (B - A) = \emptyset$

2. From (i, proposition 1), follows $M_1(A, B - A)$ *discriminates* $(A, B - A)$

3. From (i, proposition 2), follows $A\; justifies\; M_1(A, B - A)$

4. From (ii, iii), follows M_1 is a learner machine.

5. From (ii, iv), follows $\Pi(M_1(A, B - A), B) = A$ □

Proposition 4: There exist infinite learner machines M_μ such that $\forall\, A, B \subseteq \Sigma^*$ with $A \subset B$ and B is finite, $M_\mu(A, B - A)$ calculates a set of templates that *correctly* accepts A with respect to B.

Proof 4: If B is a finite set of words, then there exist infinite words outside B. We will prove that there are learner machines that calculate templates accepting not only A but also words outside B. For any non-empty $X, Y \subseteq \Sigma^*$, $X \cap Y = \emptyset$, and $s \in X$, $u \notin (X \cup Y)$, we will define $M_\mu(X, Y) := M_1(X, Y) \cup \{(r \,|\, \mu)\}$ where $r = P_1(s)$, and $\mu = P_1(u)$. Note that $L(M_\mu(A, B - A)) = L(P_2(A)) \cup L(\{(r \,|\, \mu)\}) = A \cup \{s, u\}$.

1. Since $A \subseteq L(M_\mu(A, B - A))$ and $(B - A) \cap L(M_\mu(A, B - A)) = \emptyset$ follows that $M_\mu(A, B - A)\, discriminates\, (A, B - A)$

2. Even considering the regex $\{(r \,|\, \mu)\}$ the set $A\, justifies\, M_\mu(A, B - A)$.

3. From (i,ii) follows M_μ is a learner machine.

4. Since Σ^* is a non-finite set of words and B is finite, there are infinite calculable trivial regular expressions $\mu \in R_\Sigma$ such that $\mu = P_1(u)$, $u \in \Sigma^* - B$.

5. From (iii,iv) follows $\Pi(M_\mu(A, B - A), B) = A$ □

Our main goal is to bring the system the ability to generalize the language in such a way that it could recognize related sentences never seen before —step (iv) in proof 4 lends support to this idea. To this end, we add a constrain to the problem whereby the language covered by a set of regular expressions is broader than the initial positive example set.

Property: Generalization A set of templates $E \subseteq R_\Sigma$ generalizes two set of words $A, B \in \Sigma^*$, denoted as $E\, generalizes(A, B)$, if:

$$\begin{aligned} &E\, discriminates\, A, B \\ &A\, justifies\, E \\ &A \subsetneq L(E) \end{aligned} \tag{IV.8}$$

It is assumed that the positive example set I^+ is just a (finite) sampling over some unknown vaster set of positive words $\overline{I^+}$. Our aim is to approximate this set by generalization. In this regard, a learner machine M is said to generalize $I^+, I^- \in \Sigma^*$ if $M(I^+, I^-)\, generalizes\, (I^+, I^-)$ (see Figure iv.1). Since the expert know by intuition the boundaries of $\overline{I^+}$ and $\overline{I^-}$, he/she plays a key role while repairing the templates. Reparations aims to: (i) avoid overlapins with $\overline{I^-}$ (see **(1)** in Figure iv.1), and (ii) provide a better coverture of $\overline{I^+}$ (see **(2)** in Figure iv.1).

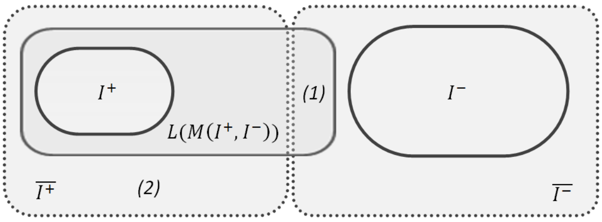


Figure iv.1: The effect of Generalization coverture. (1) overlaping area, (2) lack of coverture

Proposition 5: There exist learner machines M_g such that $\forall I^+, I^- \subseteq \Sigma^*$ with $I^+$is finite, $I^+ \cap I^- = \emptyset$, and $I^+ \cup I^- \neq \Sigma^*$, then $M_g(I^+, I^-)$ calculates a set of templates that *generalizes* (I^+, I^-).

Proof 5: The demonstration is similar to that of proof 4. Even if $I = I^+ \cup I^-$ is not finite, there will always be words $u \in (\Sigma^* - I)$ that lead us define $\mu = P_1(u)$, such that M_μ from proof 4 is a learner machine. Since $L(M_\mu(I^+, I^-)) = I^+ \cup \{u\}$ follows that $I^+ \subsetneq L(M_\mu(I^+, I^-))$, that is, $M_\mu(I^+, I^-)\, generalizes\, (I^+, I^-)$. Finally, M_g is exactly M_μ from proof 4 □.

Property: Non-redundancy Since templates should be revised by an expert, obtained regular expressions sets should be as interpretable as possible. Unfortunately, interpretability criterion is not well-defined. However, the number of obtained regular expressions is arguably related to this concept —the more regular expressions obtained, the less interpretable. In this regard we consider a constraint whereby no redundant regular expressions could belong to the same set of templates.

A set of templates $E \subseteq R_\Sigma$ is said to be *non-redundant iff* all its templates contribute in the regular language accepted by E. That is:

$$E\, non\, redundant \longleftrightarrow (r \in E \longrightarrow L(E - \{r\}) \subsetneq L(E)) \tag{IV.9}$$

Beyond interpretability, there are further reasons to consider this restriction. Let us suppose that the following regular expressions were obtained for a given entry: $r_1 = what * name * (director|company)$, and $r_2 = *name*$. Note also that $L(r_1) \subset L(r_2)$. If an expert revises and repairs a subsumed regular expression, e.g. $r_1 = what * name * director * (company)?$, the performance of the system may not be altered since queries such that *"What is the name of the company?"* are still in the language defined by r_2. However, this restriction does not assume that subsumed expressions should be removed. Indeed, subsumed expressions may be more precise (these concepts will be later addressed).

Proposition 6: There exist learner machines M_n such that if $\forall I^+, I^- \subseteq \Sigma^*$ with $I^+ \cap I^- = \emptyset$, and $I^+ \cup I^- \neq \Sigma^*$, and I^+ is finite, then $M_n(I^+, I^-)$ *generalizes* (I^+, I^-), and $M_n(I^+, I^-)\, non\, redundant$.

Proof 6: To demonstrate proposition 6 we take any of the learner machines M_g as defined in proof 5. For any $X \subset \Sigma^*$ let us consider the procedure *reduction*(X) that inspects, in lexicography order, all regexes $r \in X$, and removes each (redundant) regex r satisfying $L(X - \{r\}) = L(X)$. Let us define $M_n(X, Y) :=$ *reduction*$(M_g(X, Y))$ to the procedure that calculates this set of *non-redundant* templates. Since I^+ is finite, $|M_g(I^+, I^-)| = |I^+| + 1$, so the set of templates $M_n(I^+, I^-)$ is also finite and calculable. Note that the language accepted by $M_n(I^+, I^-)$ is exactly the language accepted by $M_g(I^+, I^-)$.

1. Since M_g is a learner machine and $L(M_g(I^+, I^-)) = L(M_n(I^+, I^-))$ follows that $M_n(I^+, I^-)\, discriminates (I^+, I^-)$.
2. From $M_n(I^+, I^-) \subseteq M_g(I^+, I^-)$ and $I^+\, justifies\, M_g(I^+, I^-)$ follows $I^+\, justifies\, M_n(I^+, I^-)$.
3. From (i,ii) follows M_n is a learner machine.
4. After removing redundant regexes follows $M_n(I^+, I^-)\, non\, redundant$.
5. From $L(M_g(I^+, I^-)) = L(M_n(I^+, I^-))$ and $M_g(I^+, I^-)\, generalizes\, (I^+, I^-)$ follows $M_n(I^+, I^-)\, generalizes\, (I^+, I^-)$ □

1.4.1 Learning Templates that Cover a Base of Questions

In this section, we specify the learning problem applied to a base of questions. To this end, we will formally define the concept of *base of question*, some desirable properties to ensure that the problem is actually feasible, and how learner machines could be applied to address it.

Definition of Base of Questions From now on, the alphabet Σ will be the set of all words of any natural language[5] (e.g. English or Spanish). Thus, every $s \in \Sigma^*$ will be a linguistic construction of

[5] This terminology may be confusing for the reader. Note that, regarding regular expressions notation, a *word* consists of a combination of symbols of Σ. Regarding a base of questions, a *question* is a combination of *words* of this natural language. In this case, *words* of the natural language conforms the set of symbols Σ.

this natural language, including both grammatically correct constructions (sentences) and incorrect constructions. In any case, Σ^* includes all possible sentences of this natural language. In this study we are only dealing with (correct) sentences in training.

Base of queries. Let $\mathcal{B} = \{C_1, C_2, \ldots, C_n\}$ be a *base of queries* with n entries. Each entry $C_i = \{s_1^i, s_2^i, \ldots, s_m^i\}$, represents a particular topic in the domain (for example, a FAQ entry) that consists of a non-empty set of NL questions $s_j^i \in \Sigma^*$ covering this topic. Any question in C_i is said to be a reformulation of the rest of questions in C_i. We use

$$[\mathcal{B}]^{\cup} = \bigcup_{C_i \in \mathcal{B}} C_i \tag{IV.10}$$

To denote all the questions contained in a base of questions $\mathcal{B}$.

Property: consistency of a base of queries. A base of questions $\mathcal{B}$ is said to be *consistent* if $\mathcal{B}$ is a partition of $[\mathcal{B}]^{\cup}$, that is $\forall i \neq j$, $C_i \cap C_j = \emptyset$

Property: completeness of a base of queries. A base of questions $\mathcal{B}$ is said to be *complete* if $\mathcal{B}$ contains, exhaustively, all possible questions relevant to the domain.

Property: finiteness of a base of queries. A base of questions $\mathcal{B}$ is said to be *finite* if $[\mathcal{B}]^{\cup}$ is finite. Note that, since each entry is non-empty, *finiteness* implies that $\mathcal{B}$ is also a finite set of entries.

It should be remarked that, in practice, a base of questions is not *complete*. Questions in $[\mathcal{B}]^{\cup}$ are considered just a sampling over all possible questions relevant to the domain. For the same reason, real bases of questions are evidently *finite*. However, as usually questions in $[\mathcal{B}]^{\cup}$ are collected and manually classified in any entry of $\mathcal{B}$, *consistency* is not assured —human mistakes could cause the misclassification of some questions. However, *consistency* is easily achievable by reconsidering the membership of duplicate questions in $\mathcal{B}$.

1.4.2 Learner Machines Meets Question Recognition

Coverage of a base of questions. Given a base of questions $\mathcal{B} = \{C_1, C_2, \ldots, C_n\}$, a partition of templates $\mathcal{T} = \{T_{C_1}, T_{C_2}, \ldots, T_{C_n}\}$ is a *coverage* of $\mathcal{B}$ if T_{C_i} *accepts* C_i for each $1 \leq i \leq n$.

Coverage with learner machines. Given any base of cases $\mathcal{B} = \{C_1, C_2, \ldots, C_n\}$ and any learner machine M, the partition of templates $\mathcal{T}_M(\mathcal{B}) = \{T_M^{\mathcal{B}}(C_1), T_M^{\mathcal{B}}(C_2), \ldots, T_M^{\mathcal{B}}(C_n)\}$, where

$$T_M^{\mathcal{B}}(C_i) := M(C_i, [\mathcal{B}]^{\cup} - C_i) \tag{IV.11}$$

is a *coverage* of $\mathcal{B}$, that will be called *the coverage* of $\mathcal{B}$ by M.

Extensions of properties correctness, generalization, and non-redundancy for a coverage. A coverage $\mathcal{T}$ of a base of questions $\mathcal{B}$ is said to be (i) *correct*, (ii) *generalizable*, and (iii) *non-redundant* if and only if (i) for any $C_i \in \mathcal{B}$, T_{C_i} accepts C_i *correctly*, (ii) for any $C_i \in \mathcal{B}$, T_{C_i} *generalizes* $(C_i, [\mathcal{B}]^{\cup} - C_i)$, and (iii) for any $C_i \in \mathcal{B}$, T_{C_i} is *non-redundant*, respectively.

Finally, the theoretical result that is the basis of our proposal could be formulated: if certain properties are assured in the base of questions, then the problem can be solved.

Theorem: Reachability of the solution. For any *finite*, *consistent*, and *not complete* base of questions $\mathcal{B}$, there exist learner machines M such that $\mathcal{T}_M(\mathcal{B})$ is *correct*, *generalizable*, and *non-redundant*.

Proof: This proof is based on the demonstration that there exist M learner machines that calculate templates complying with the above mentioned properties, for all entries C_i at once.

Propositions 6 demonstrates that for any sample set $I = I^+ \cup I^-$, if $I^+ \cap I^- = \emptyset$, and $I^+ \cup I^- \neq \Sigma^*$, and I^+ is finite, then exist M_n learner machines complying with $M_n(I^+, I^-)$ *generalizes* (I^+, I^-) and $M_n(I^+, I^-)$ *non redundant*. In addition, as proof 6 is based on any of the learner machines of proof 4, $M_n(I^+, I^-)$ $correct_{I^+ \, wrt \, I}$.

Obviously, since $\mathcal{B}$ is *finite* and *consistent*, this is also true for the particular case $I = [\mathcal{B}]^\cup$ and a fixed i, where $I^+ = C_i$ and $I^- = [\mathcal{B}]^\cup - C_i$. That is, $T^{\mathcal{B}}_{M_n}(C_i)$ $correct_{C_i \, wrt \, [\mathcal{B}]^\cup}$, $T^{\mathcal{B}}_{M_n}(C_i)$ *generalizes* $(C_i, [\mathcal{B}]^\cup - C_i)$, and $T^{\mathcal{B}}_{M_n}(C_i)$ is *non-redundant*.

To demonstrate that a learning machine could comply with all properties for all entries C_i at once, we will consider the learning machine M_{th} defined as

$$\forall X, Y \subseteq \Sigma^*; M_{th}(X, Y) := P_2(X) \cup \{(r \mid \mu)\} - \{r\} \tag{IV.12}$$

where $r = P_1(s)$ for any $s \in X$, and $\mu = P_1(u)$ for any[6] $u \notin (X \cup Y)$.

Note that the procedure calculated by the learning machine M_{th} is equivalent to the procedure calculated by M_n in proof 6 (because the only redundant regex in M_g is r). Then, considering this learning machine,

$$T^{\mathcal{B}}_{M_{th}}(C_i) \; correct_{C_i \, wrt \, [\mathcal{B}]^\cup}$$

$$T^{\mathcal{B}}_{M_{th}}(C_i) \; generalizes \; (C_i, [\mathcal{B}]^\cup - C_i), \text{ and}$$

$$T^{\mathcal{B}}_{M_{th}}(C_i) \text{ is } non\text{-}redundant$$

With independence of a particular i, because $I = [\mathcal{B}]^\cup$ is common to all i. Thus,

$$\mathcal{T}_{M_{th}}(\mathcal{B}) = \{T^{\mathcal{B}}_{M_{th}}(C_1), T^{\mathcal{B}}_{M_{th}}(C_2), \ldots, T^{\mathcal{B}}_{M_{th}}(C_n)\}$$

Is a *correct*, *generalizable*, and *non-redundant* coverage for all entries $C_i \in \mathcal{B}$ at the same time □.

[6] Note that a procedure obtaining such example u is calculable. If this procedure explores systematically whether a word does not pertains to $[\mathcal{B}]^\cup$, it would end in, at worst, $|[\mathcal{B}]^\cup| + 1$ steps.

2 Learning the Minimal Differentiator Expressions for FAQ Retrieval

2.1 Introduction

A common strategy to overcome the costs of call centres is to collect and answer most typical user questions on a given domain. Those lists of frequent questions and expert answers, called FAQ (Frequently Asked Questions) is made available to users as a preliminary filter to ease congestion in call centres.

Those lists, however, present a number of drawback as a service. According to Sneiders [Sne99], manual searches on traditional FAQ lists impose certain limitations. The higher the FAQ, the more tedious is for the user to identify a relevant entry. On the contrary, the smaller the FAQ, the more unlikely is the presence of useful information. Finding a convenient balance between these opposite points is not trivial. In this scenario, an automatic Information Retrieval (IR) method could make the difference: automatic searches allow the FAQ content to be increased —thus incurring in a better domain coverage— while users are released from the tedious task of reading the FAQ entries.

As commented before, the FAQ retrieval problem is aimed for answering NL questions by recovering the most related questions/answer pairs in the FAQ. Those pairs are usually displayed as ranked lists, according to their relevance to the user query. This is therefore a representative example of NL closed-domain system where knowledge is structured on questions/answer entries —which may be, in turn, organized in different categories— and there is, in principle, no meta-knowledge resource describing domain entities or semantic relationships. And we say *in principle* because some recent proposals to that end do actually take into account some kind of additional knowledge resource, e.g., domain ontologies [WTRM08] or *keyword*[7] sets [Sne09]. As pointed out before (chapter III), designing and maintaining this meta-knowledge resources may entail considerable effort that could also lead to a restriction on the manner the FAQ could be enhanced. In fact, providing the domain the opportunity to be easily extended to face possible lack of coverages undeniably represents a valuable goal to be pursued. How to allow this enhancement and how to discover possible knowledge gaps will therefore represent the main targets of this part of the dissertation. Additionally, we will offer a method to improve the performance on that scenarios where the system performance is critical (section 4).

There are different manners to address this problem. Some related techniques, such as statistical ones [BHK+97], present the advantage of simplifying the knowledge modelling task, but they are not easily interpretable. Other methods use Case Based Reasoning (CBR) [LHK98] or ontological [WTRM08] models to address the IR problem. These models are often related to the concept of keyword or linguistic rules. These kinds of approaches are arguably more interpretable, but selecting keywords or designing linguistic rules is a costly task that is usually carried out manually. Therefore, a complete differentiation among the cases is not always assured using these methods. Furthermore, if the knowledge of the domain grows up, modifying these patterns might become a very complex task. Let us consider the following FAQ example to illustrate our aim. The FAQ under consideration concerns with an Operating System (OS) with only two entries (*Installing* and *Removing*). In this example, a set of linguistic rules and a set of keywords will also be considered (Table IV.1).

If a user asks *Could you explain to me how I can install a new program in my computer?* then the entry *Installing* should be retrieved. Note that keywords “install” or rule *if*(‘how’ $\wedge$ ‘install’)

[7] Most relevant words describing the main concepts in the domain.

FAQ entry	**Sentence**	**Linguistic Rules**	**KeyWord**

Installing	How do I install a program?	*if*('how' $\wedge$ 'install') *then...*
		if('install' $\wedge$ 'progam') *then...*
			{install, program, remove}

Removing	How do I remove a program?	*if*('how' $\wedge$ 'remove') *then...*
		if('remove' $\wedge$ 'progam') *then...*	

Table IV.1: Domain entries

then... suffice to classify the question. However, let us consider the query *I want to install a new update, should I remove the older version of my program?* (related to entry *Installing*). In this case there is an overlapping between words in the sentence and keywords ("install", "program", "remove"). The same goes for the rules *if*('install' $\wedge$ 'progam') *then...* and *if*('remove' $\wedge$ 'progam') *then...*. These linguistic interferences are hardly perceivable by an expert, even if he/she considered those sentences in advance. Therefore, an automatic method capable of detecting such interferences would be desirable. Thus, such method could take advantage of collecting sentences, which is a low cost task, in order to refine its performance. For example, our method would be able to find out the following relations between words:

"how - install", "update - program", "install - new - update" → *Installing*
"how - remove" → *Removing*

In this chapter, we focus on finding the relations between words that allow the complete differentiation among cases avoiding overlapping. We intend to generalize the concept of keyword by looking for relevant expressions instead of just looking for relevant words. To this purpose, we will take the study presented in chapter II as an starting point. In this chapter, we propose a new FAQ retrieval approach. The user question is answered by searching the most similar Question/Answer pairs in the FAQ. The main advantage of our approach consists of automatically obtaining those interpretable multi-word expressions which are used to implement the similarity measure. This study will be drawn under the theoretical considerations exposed in section 1.4. Thus, our proposal could be regarded as a Learner Machine. However, in this first approach, we will just pay the necessary attention to the restrictions a Learning Machine imposes: minimal differentiator expressions are a *correct*, *generalizable*, and *non-redundant* coverage for all entries in the FAQ. Other issues such as how to optimise the *generalization degree*, or how to consider *interpretability* will be later addressed in section 4.

The rest of this chapter is organized as follows: Section 2.2 reviews previous work in the area of FAQ retrieval. Section 2.3 presents a general scheme of our proposal. In Section 2.3, the way in which the Minimal Differentiator Expressions are extracted and how they are used in the classification stage will be explained. Section 2.4 offers an evaluation of our system using three different FAQs. Finally, the conclusions and future work are presented in Section 2.5.

2.2 Related Work

According to Sneiders [AS05], different strategies to deal with the problem of FAQ retrieval can be classified into three types: Natural Language Processing (NLP), Statistical techniques, and Template-based QA.

NLP techniques aim to obtain a formal representation of NL to give back a concise answer. They use language-processing techniques as well as stemming, morphological normalization, Part Of Speech Tagging (POST), or thesaurus. Some examples of NLP techniques could be found in [Ott92, MHG$^+$02, WMC09, TR08]. Although its more direct application is Natural Language Interface to DataBases (NLIDB) [ART95], languages translation or extraction of information in texts [RHM$^+$02], it has been also applied to the problem of FAQ retrieval, analysing both user and FAQs questions. This analysis considers syntax, morphology, and lexis by a language model based on linguistic rules, generally set by hand, and optionally with the help of domain ontologies [Gru93, GZ09b]. FAQ Finder [HBML95] was the first FAQ retrieval system in this line. This system combines keyword analysis with two similarity metrics among questions to achieve the recovery of relevant FAQs from the USENET news group. First, it calculates a vector space metric with tf·idf [SM86] considerations. After that, it calculates the semantic distance between pairs of words with the help of WordNet [MBF$^+$90] (semantic web of words), which consists of counting the distance between links of hyperonymy/hyponymy among groups of synonyms. Further researches have been carried out in this line [BHK$^+$97, JR05, GHW07]. [Win00] mixes domain-dependent knowledge with general lexical rules. It makes a first lexical analysis that is used to execute a classification into categories through a second semantic analysis. Each category has an associated pattern, which is used to give back the FAQ and then, the most relevant answer is retrieved. Distance between words is calculated by means of a hierarchical dictionary. [YCH07] combines an ontology of the terms in the domain with a probabilistic keywords comparison measure to retrieve the FAQ with the highest matching with regard to the terms extracted from the user's question. Other applications, as well as [CCV$^+$07], operate with graphs built-up over domain ontologies and linguistic sentences which are used to compare them. DynJAQ [CRM07] focuses in answering questions about Java. It integrates case-based knowledge into a graph-based representation to implement learning graphs. FRACT [KS08b] only uses POST and a syntactic analyser in order to perform clustering over FAQs. It only uses query logs from users as knowledge resource. Collecting those query logs is easier than performing a knowledge modelling. It is based on Latent Semantic Analysis (LSA), a method to extract and represent the contextual-usage meaning of words. Its main advantage is that it is not domain dependent.

Statistical techniques take advantage of large number of questions contained in training datasets. These techniques establish semantic distance measures between the FAQ entry and the user question. The main problem of these techniques consists of identifying whether two questions with different wording are semantically related. Different methods to define syntactic links between linguistic structures with the same semantic (synonyms) are proposed in the literature. The most usual techniques in this line of work include stemming, deleting stop words, and identifying n-grams. The followings works are examples of statistical FAQ retrieval systems [JR05, JCL05, SB06, BCC$^+$00]. FallQ [LHK98] focuses on closed domains. It uses the expert knowledge to define keywords in order to build Information Entries (IEs). The main difference from classical techniques lies in the classification elements. They use sentences instead of documents. The expert knowledge dependence is the main disadvantage of this method. SPIRE [DR97] combines CBR with IR techniques. It works in two different stages: firstly, CBR is used to obtain a reduced number of documents and then, these documents are sent to the INQUERY retrieval engine module and they are processed performing IR techniques. Its main disadvantage consists of the need to label text passages. In [Sne99], a shallow language understanding strategy is mixed with keywords based strategies. Auto-FAQ [Whi95] works in a similar way, a keyword comparison criterion is applied to implement the question matching. Others methods like [KS06] use automatic clustering of the registered questions in order to increase the number of question to have more variability in the examples. International conferences like Text REtrieval Conference (TREC) [Voo01] have promoted the development of new techniques that mix statistical methods and NLP. An example can be found in [KEW01]; in this work, a syntactic parser is applied to classify the kind of question. After that, a comparison between keywords is used. [XJC08] calculates the probability of translation between Question/Answer pairs.

Finally, template-based systems are aimed for covering the knowledge with a set of linguistic templates which are later used for matching. START [Kat97] with Omnibase [KFY$^+$02] is an example of this approach. It employs a lexical-level (handling synonymy and IS-A relations) and a syntactical-level comparator. The Sneider's template-based systems [Sne02, Sne99] are other examples of this kind of systems. These systems use matching with both regular expressions and keywords. Furthermore, [Yan09b] combines a template-based approach with a domain ontological model based on keywords. This system needs a knowledge domain model in which domain concepts are represented. Designing these templates is a complex task that is usually carried out manually by an expert. Some examples of commercial application systems in this line include ASK[8] or KiwiLogic[9].

2.3 FAQ Retrieval Engine

To deal with the problem of FAQ Retrieval, many of the current models use high-level knowledge bases that were built ad-hoc for specific domains. However, adapting those systems from one domain to another usually takes a lot of time and effort. These kind of systems will rather be addressed in chapter V. Instead, we propose here an incremental knowledge model that learns by addition of new sentences. These sentences or reformulations are collected in a very simple way, registering the questions of the users during the life of the system, so our FAQ retrieval system will be improving its performance along its use.

The retrieval algorithm is the central part of this study. The solution is given to the user by taking the answers associated to the retrieved Question/Answer pairs. Those answers are displayed to the user sorted by relevance. Finally, the user query could be added to the knowledge base in order to refine its performance.

The query is preprocessed by normalizing its characters (removing punctuation marks and converting all words to lower case) and performing a stemming process that represents words sharing a common meaning into one single form. Many of the NLP approaches remove stop words (non-standard set of meaningless words) in the preprocessing stage in order to reduce noise. However, since it is considered that removing stop words may cause a loss of information, we preserve them.

In this section, the retrieval algorithm will be explained in detail. Firstly, the classification criteria that will guide the algorithm search will be exposed. After that, how linguistic classifiers are obtained and how they are used to classify will be described.

[8] `http://www.ask.com/`

[9] `http://www.kiwilogic.com/`

2.3.1 Classification Criteria: Differentiability and Minimality

As pointed out before, we will take conclusions drawn in chapter II as the starting point. In that chapter, we offered an study on term relevancy. We concluded that all categories should be equally attended while searching for relevant words. In addition, positive correlation was found the most promising criterion to reflect the term relevancy to a given domain. We found that differentiation among classes could be approached even from extremely-reduced term spaces.

As we saw in the example of Table IV.1, the word "program" seemed to be relevant to the domain. However, it actually adds no useful information to the classification process. Note that words "install" and "remove" suffice to distinguish the FAQ entries (see Table IV.2). For example, the queries *Could you explain to me how to* ***install*** *new Software?* and *Can I* ***remove*** *a program if I am not logged in as Root?* would be successfully classified. However, if a new FAQ entry is considered (see Table IV.3), those classifiers are advocated to fail. In that new case, words "program" and "device" become now relevant. Therefore, this suggests that there are not isolated words (keywords) which actually contribute to classify, but the relations between words (multi-word expressions). In the example, the relation between "install" and "program" has enough classification information. These relations between words will be called *Differentiator Expressions* (DE) and could be captured by means of regular expressions. The DE extracted from the first sentence would be $\ast install \ast program \ast$, where symbol $'\ast'$ denotes "any chain of characters" . The following examples: $\ast install \ast device \ast$, $\ast install \ast new \ast device \ast$, $\ast how \ast install \ast device \ast$, or $\ast device \ast$, are also differentiator expressions. However, as the set of all DEs may be too broad, it seems reasonable attending only to the ones containing the fewest words. These expressions, here called *Minimal Differentiator Expressions* (MDEs, [MCLZ10, MNCZ12]), are therefore the simplest and most generic multi-word expressions that allow differentiating among domain entries. Moreover, there exist differentiator expressions like $\ast do \ast I \ast a \ast$ which are semantically non-related to any entry. These expressions could be differentiative from a linguistic point of view, but not from a semantic perspective. These noisy expressions might be removed, in order to avoid the system misdirects the user to unrelated entries. According to this, our algorithm will try to find out the minimal and most relevant relations between words that allow the complete differentiation among linguistic sentences.

Class	Sentence	Keywords
Installing	How do I install a program?	{install, remove}
Removing	How do I remove a program?	

Table IV.2: Initial case base.

Class	Sentence	DE	MDE
Installing	How do I install a program?	$\ast install \ast program \ast, \ldots$	$\ast install \ast program \ast$
Removing	How do I remove a program?	$\ast remove \ast program \ast, \ldots$	$\ast remove \ast$
AddHardware	How do I install a new device?	$\ast install \ast device \ast, \ldots$	$\ast device \ast$

Table IV.3: FAQ after adding a new entry.

The following comment may serve to clarify the intuition beyond MDEs: "an MDE is composed by a set of words satisfying (i) the *addition* of any word makes it be no longer *minimal*, and (ii) the *removal* of any word makes it be no longer *differentiator*"

In connection with chapter II, the Minimal Differentiator Expressions algorithm might be regarded as a Feature Selection algorithm that takes the minimal set of features to totally differentiate each FAQ reformulation. MDEs generalize not only on most cases but also in outliers. This criterion leads us to think that MDEs will get good characterizations of the sentences, because the addition of just one sentence may cause some similar linguistic interferences to be solved. Conceptually, the main difference with respect to other classical classifiers such as K-nn is that each classifier unit is weighted according to its discriminant capacity (section 2.3.2).

2.3.2 Obtaining the MDE Classifiers

Before describing the obtaining procedure, the concepts of expression, differentiator expression, and minimal differentiator expression will be formally defined.

Preliminary Definitions In this part of the study we are only considering a subclass of regular expressions we will call *plain regular expressions.* Concretely, we will deal only with regular expressions composed as a concatenation of symbols in Σ bounded by wildcards. More formally and according to the notation shown in section 1.4, plain regexes R_P will be here defined as $(*\ \Sigma)^*\ *$ —recall that $*$ is here used as a shorthand for Σ^*, that is, any chain of symbols of the alphabet Σ. Note also that $R_P \subset R_\Sigma$. We will deal with a broader set of more elaborated regexes in section 4.

Definition 1: Expression. Being S a sentence, the plain regular expression $e \in R_P$ is said to be an *Expression* of S, noted as "*e exp S*", *iff*: (i) It is composed by a subset of the words in S arranged in the same order. (ii) Not all its words are stop words.

$$e\ exp\ S \longleftrightarrow S \in L(e) \wedge \Sigma_{\mathcal{S}}^* \cap L(e) = \emptyset \tag{IV.13}$$

Where $\Sigma_{\mathcal{S}} \subset \Sigma$ is the alphabet containing all the stop words. The set of all the expressions of a given FAQ entry G is denoted as:

$$E(G) = \{e \mid \exists S \in G, e\ exp\ S\} \tag{IV.14}$$

Following examples regarding Table IV.3 may be illustrative. Note that $*how*$, $*how*install*$, $*how*install*program*$, and so on, are valid expressions of *Installing.* Note that they are composed by a subset of the words in sentence *How do I install a program?*, the words are arranged in the same order, and not all words are stop words. In contrast, the followings ones are not valid expressions: $*do*a*$ because it only contains stop words, and $*how*program*install*$ because it is not in the correct order.

Definition 2: Differentiator Expression. An expression e is a said to be a *Differentiator Expression* (DE) of sentence S with respect to a set of FAQ entries F if it unambiguously distinguishes S from all sentences in F. A DE is defined as follows:

$$e\ exp_{DE}\ S\ wrt\ F \longleftrightarrow e\ exp\ S \wedge \forall H \in F,\ e \notin E(H) \tag{IV.15}$$

Where *wrt* is "with respect to", and $e\ exp_{DE}\ G\ wrt\ F$ represents that e is a differentiator expression of the FAQ entry G. Finally, the set of all differentiator expressions of G with respect to the entry set F is denoted as:

$$DE(G,F) = \{e \mid \exists S \in G,\ e\ exp_{DE}\ S\ wrt\ F\} \tag{IV.16}$$

Table IV.3 shows examples of DEs. Note that $*install*$ and $*how*program*$, are expressions but not DEs because they do not differentiate between different entries.

Definition 3: Minimal Differentiator Expression. A *Minimal-Differentiator-Expression* (MDE) is a differentiator expression e involving the smallest set of symbols in Σ. Since each symbol imposes a restriction in the language the regular expression accepts, this definition implies MDEs are the most *generalizable* DEs.

$$e\ exp_{MDE}\ S\ wrt\ F \longleftrightarrow e\ exp_{DE}\ S\ wrt\ F \wedge \forall e' \in (DE(G,F) - \{e\}),\ L(e) \not\subset L(e') \qquad \text{(IV.17)}$$

We will denote the set of all MDEs in an entry G with respect to the entry set F as:

$$MDE(G,F) = \{e \mid \exists S \in G,\ e\ exp_{MDE}\ S\ wrt\ F\} \qquad \text{(IV.18)}$$

Some examples of minimal differentiator expressions are shown in Table IV.3. Note that, even though $*remove*program*$ is a differentiator expression for the semantic case *Removing*, it is not minimal, because it contains the MDE $*remove*$, which has fewer number of terms and also allows differentiation.

Minimal Differentiator Expressions Algorithm The algorithm calculates a *coverage* of a given FAQ F by iteratively computing the $MDE(F_i, F - \{F_i\})$ set. This algorithm implements, thus, a Learner Machine as defined in section 1.4.

The search is conceived as an exploration of the possible combinations among words in a given sentence, exploring them through a tree of combinations by levels to avoid recalculations (Algorithm iv.2). It constructs a registry R of MDEs already calculated that allows to identify whether an already obtained differentiator expression contains another one. The algorithm also handles a queue of candidate expressions. Those expressions which still are candidates for being an MDE are labelled as OPENED. Otherwise, the CLOSED label is used and the branch is pruned.

The number of words in sentence S is noted as ℓ_S and each word contains an index which informs of its position in the sentence, consulted by the function $index(\cdot)$. We will denote $S[k]$ to the kth word in S.

The branching factor of the tree may get very high. That is why the tree is pruned in line 11 by an α threshold (as an experimental value, we set $\alpha = 3$ by default) so no expressions with more than α words will be obtained.

There are two situations where a set of words to differentiate a sentence cannot be extracted. The first one is related to the depth of the search. In this case, to differentiate a sentence, it would be necessary to extract a number of words higher than α. The algorithm will not calculate those MDEs, because it would require a non-feasible computational cost. The second situation occurs when there are subsumed sentences. Sentence S is said to be subsumed into sentence T if all the words in S are also in T, respecting the order of precedence. For example, question *How do I install a program?* is subsumed into question *How do I install a new device using the device manager program?*. To solve this problem, in lines 18 to 20, *exact expressions* are created. An exact expression is an expression which concatenates all words in the sentence and does not allow wildcards $*$. The procedure that calculates the MDEs set must check if the candidate expression is already registered in R or if it is included in a sentence from a different entry —if so, the expression is not a differentiator one. $CheckMDE(e, R, A)$ is the boolean function that implements this checking. This function verifies that the expression e is not contained in a set R, and is not an expression of any of the sentences in A (Equation IV.19).

$$CheckMDE(e, R, A) \longleftrightarrow e \notin R \wedge \forall S \in A, \neg(e\ exp\ S) \tag{IV.19}$$

```
Procedure getMDEs($S, F, R$)
MDEs:=$\emptyset$
OPENED:=$\emptyset$
for $k := 1$ to $\ell_S$ do
  Push(OPENED, $\{S[k]\}$) //initialization
end for
while OPENED$\neq \emptyset$ do
  expr:=pop(OPENED)
  if CheckMDE(expr, R, $F$) then
    MDEs:=MDEs $\cup$ $expr$ //is a MDE, this branch is pruned
  else
    if $size(expr) < \alpha$ then
      for $t := index(tail(expr)) + 1$ to $\ell_S$ do
        $push(OPENED, expr \cup \{S[t]\})$ //explores subsequent combinations of words
      end for
    end if
  end if
end while
if MDEs=$\emptyset$ then
  MDEs:=$\{$exact($S$)$\}$ //creates an Exact expression of $S$
end if
R:=R $\bigcup$ MDEs
return MDEs
```

Figure iv.2: Minimal Differentiator Expressions algorithm

See Appendix 1 to access a sample trace of the algorithm.

Weights of the MDEs Once the algorithm has calculated all the MDEs, weights to the MDEs are assigned in order to measure its quality as a classifier. The weight of the expression is calculated using two criteria: Relevance into the FAQ entry (*EntryRelevance*) and Relevance into the domain (*DomainRelevance*).

The *EntryRelevance* of an expression e in the entry G is calculated as the proportion of reformulations S in G that satisfy $e\ exp\ S$, according to Equation IV.20. In this way, less important MDEs will not lead the classification process.

$$EntryRelevance(e, G) = \frac{|\{S \in G | S \in L(e)\}|}{|\{S \in G\}|} \tag{IV.20}$$

The *DomainRelevance* depends on the number of words in the expression (according to the heuristics that the closer to a complete sentence it is, the more precise it will be) and the importance λ_w of each word in the expression. The word-weight criteria *idf(w)* of the tf·idf algorithm has been used, divided by $\log|F|$ to normalize it:

$$\lambda_w = \frac{idf(w)}{log|F|} = \log_{|F|} \frac{|F|}{|\{F_i \in F|f_c(w) > 0\}|} = 1 - \log_{|F|} |\{F_i \in F|f_c(w) > 0\}| \tag{IV.21}$$

Where F is the FAQ, F_i is a FAQ entry in F, f_c is the number of times that word w appears in F_i. Equation IV.21 measures the inverse frequency of the number of entries that contain the word w and it represents its informational content normalized. Finally, the *DomainRelevance* is calculated (Equation IV.22). It is divided by α that represent the maximum length that an expression could reach:

$$DomainRelevance(e, F) = \frac{\sum_{w\in e} \lambda_w^2}{\alpha} \tag{IV.22}$$

Finally, weight ω_e of expression e extracted from F_i is calculated as a convex linear combination (Equation IV.23).

$$\omega_{e\in G\ wrt\ F} = \beta \cdot EntryRelevance(e, G) + (1 - \beta) \cdot DomainRelevance(e, F) \tag{IV.23}$$

Table IV.4 shows a very simple example of the weights assigned to the expressions in a case base with only 4 cases considering $\beta = 0.25$. "Software", "update", "linux", and "programs" are some examples of words appearing in various cases, thus, they are not differentiators by themselves. However, relations **software*linux**, and **update*programs** are differentiators. Note the differentiator expression **software*repository** is not taken because **repository** is not only differentiator by itself, but also minimal. As an example of the impact of the weights, the *Kernel* entry would be more relevant to the query "How can I install the new version of the Linux Kernel?" than *AddSoft* because the weight of **kernel** is higher than the weight of **install**.

Case	Reformulations	MDEs	ERel.	DRel.	ω_e
Kernel	What is the Linux kernel? How can I update the Linux kernel?	$*kernel*$	1.0000	0.2500	0.4375
AddSoft	How can I install programs in Linux?	$*install*$	0.6667	0.2500	0.3525
	What is the best software repository?	$*repository*$	0.3334	0.2500	0.2700
	How can I install software in Linux?	$*software * linux*$	0.3334	0.0730	0.1372
UpdSoft	Can I update programs in Linux?	$*update * programs*$	1.0000	0.1250	0.3438
DelSoft	How can I remove programs? How can I remove software?	$*remove*$	1.0000	0.2500	0.4375

Table IV.4: Weighting Assignments.

2.3.3 Retrieval with MDEs

Once the set of all MDEs have been calculated and so have its weights, the retrieval procedure is defined as follows. The most similar entries among all entries in the FAQ must be retrieved. We consider the similarity function shown in Equation IV.24:

$$Score(S, F_i)_{wrt\ F} = \sum_{e_i \in MDE(F_i, F)} m(S, e_i) \cdot \omega_{e_i} \tag{IV.24}$$

Where S is the user's question, F_i is each entry in the FAQ , $m(S, e)$ measures the inclusion degree, and ω_e is the weight of expression e. Taking into account the order relations and the semantic distance between the words in the expression and the words in the sentence, the inclusion degree $m(\cdot)$ is computed. The semantic distance between a pair of words is represented by labels *Equals*, *Different*, or *Related*. Two words are said to be *Related* if they are synonymous or directly related hierarchically through WordNet [MBF+90]. To simplify the explanation, we will consider the Boolean predicates *Order(S,e)*, true *iff* non *Different* words between S and e are in the same order; *AllEquals(S,e)*, true *iff* all non *Different* pair of words between S and e are *Equals*, and *SomeRelated(S,e)*, true *iff* at least one pair of non *Different* words in S and e are *Related*. In this way, inclusion between S and e can be *Exact* ($e \subseteq_E S$), *Relaxed* ($e \subseteq_R S$), *Disordered* ($e \subseteq_D S$), or *Relaxed-Disordered* ($e \subseteq_{R-D} S$), as is shown in Table IV.5.

	$Order(S, e)$	$\neg Order(S, e)$
$AllEquals(S, e)$	$(e \subseteq_E S)$ $e = \{*install * program*\}$	$(e \subseteq_D S)$ $e = \{*program * install*\}$
$SomeRelated(S, e)$	$(e \subseteq_R S)$ $e = \{*install * software*\}$	$(e \subseteq_{R-D} S)$ $e = \{*software * install*\}$

Table IV.5: Definition of the inclusion degrees and examples considering S="How can I **install** a **program**?"

Applying the weighting criterion *m(S,e)* (see Equation IV.25), the inclusion degree between an expression e and a sentence S is modelled[10]. *m(S,e)*=1 represents the maximum inclusion degree, while *m(S,e)*=0 represents no inclusion. Intermediate inclusion degrees satisfy $1 < \gamma_1 < \gamma_2 < 0$.

$$m(S, e) = \begin{cases} 1 & if\ (e \subseteq_E S) \\ \gamma_1 & if\ (e \subseteq_R S) \vee (e \subseteq_D S) \\ \gamma_2 & if\ (e \subseteq_{R-D} S) \\ 0 & in\ other\ case \end{cases} \tag{IV.25}$$

2.3.4 Theoretical Implications

Definition of *Expression*, *Differentiator Expression*, and *Minimal Differentiator Expressions* imply certain theoretical conditions that will be highlighted in this section. According to section 1.4, a

[10] Because *Exact* inclusion is equivalent to decide $S \in L(e)$ this check could be easily implemented by using regex pattern matching tools provided by *Java* or *Phyton* programming languages. The implementation of the remaining inclusion degrees imply managing additional data structures and undeniably entails higher computational costs.

procedure M is considered to be a Learner Machine *iff* it calculates a set of regular expressions that *discriminates* the positive sample side from the negative one (Equation IV.15), and all regular expressions are *justified* by at least one positive example (Equation IV.13).

Because a Minimal Differentiator Expressions is, by definition, an Expression of at least one sentence, then follows that the *justification* condition is directly satisfied.

Because each Minimal Differentiator Expression is, by definition, a Differentiator Expression, then follows that the *discrimination* property is also satisfied.

Because MDE algorithm is a *calculable* (Algorithm iv.2) procedure complying with the *justification* and *discrimination* properties, then follows MDE is a Learner Machine.

Then, the theorem formulated in section 1.4.2 ensures

$$\mathcal{T}_{MDE}(F) = \{T^F_{MDE}(F_1), T^F_{MDE}(F_2), \ldots, T^F_{MDE}(F_n)\} \tag{IV.26}$$

Where:

$$T^F_{MDE}(F_i) := MDE(F_i, F - \{F_i\}) \tag{IV.27}$$

Is a *correct*, *generalizable*, and *non-redundant* coverage of the FAQ F.

Note that *Minimality* criterion is only meant to force a bounding criterion. Thus, that property relates only to the efficiency of the method —both from a time and space complexity, as reflected in the next section—, rather than to theoretical implications.

Finally, *correctness*, *generalization*, and *no-redundancy* properties will allow us to define more sophisticated regular expressions aimed for optimising these features (section 4).

2.3.5 Efficiency of the Algorithm

In this section, we discuss the efficiency of the MDE algorithm in terms of space complexity —how many MDEs the algorithm obtains—, and time complexity —how long it takes for the algorithm to find the MDEs. This section presents a theoretical study on efficiency (see Section 2.4.3 to access an empirical contrast).

Space Complexity The MDE algorithm identifies the MDEs for each of the n sentences. Being S a sentence of length ℓ_S, the number of all candidate expressions corresponds to the $|\mathcal{P}(S)| = 2^{\ell_S}$ possible combinations of words (brute force exploration). However, since the α-bound restricts the length of MDEs, the number of candidate expressions is, at worst, $n \cdot \binom{\ell}{\alpha}$, where ℓ is the maximum length of sentences in the FAQ F. Assuming that α is constant, space complexity is asymptotically bounded by $O(n \cdot \ell^\alpha)$. Fortunately, minimality and differentiability criteria force the MDEs to be smaller in practice. In this way, assuming that certain combination of words is minimal and differentiator, the exploration of its subsequent combinations is pruned.

The number of words needed to differentiate a given sentence depends on the domain complexity. Further, no additional assumption on the number of MDEs could be considered. However, even considering ℓ and α to be constants in terms of efficiency, the truth is that it takes much time to process long sentences. For this reason, we consider α_S as a dynamic parameter depending on the particular length ℓ_S of each sentence S (Equation IV.28).

$$\alpha_S := max\left\{a \in N | (a \leq \ell_S) \wedge ((\ell_S)^a < 10^3)\right\} \tag{IV.28}$$

The upper bound is set to 10^3 because in our datasets, the average length of the sentences was 9.36, and $\alpha = 3$ was empirically proven to be the best balance between efficiency and accuracy[11]. Thus, α_S is automatically set by establishing the upper bound $O(n \cdot 10^3) = O(n)$.

Time Complexity As commented before, the number of MDEs is, at worst, bounded by $n \cdot \ell^\alpha$. A test of minimality and differentiability is performed for each candidate expression. $CheckMDE(\cdot)$ verifies, in constant time[12], that the expression is not already contained in the set R. It also verifies the differentiability criterion in $O(n \cdot \ell \cdot \alpha)$ by comparing the expression to the rest of sentences. Thus, time complexity of the exploration of MDEs is bounded by $O(n \cdot \ell^\alpha \cdot (n \cdot \ell \cdot \alpha + 1)) = O(n^2 \cdot \alpha \cdot \ell^{\alpha+1})$.

The time complexity to obtain the weights of the MDEs depends on the calculation of *EntryRelevance* and *DomainRelevance*. Since *EntryRelevance* is obtained in $O(n \cdot \alpha \cdot \ell)$ and *DomainRelevance* is obtained in $O(\alpha)$, time complexity is bounded by $O(n \cdot \ell^\alpha \cdot (n \cdot \alpha \cdot \ell + \alpha)) = O(n^2 \cdot \alpha \cdot \ell^{\alpha+1})$.

Finally, the efficiency in time of the algorithm is $O(2 \cdot n^2 \cdot \alpha \cdot \ell^{\alpha+1})$. Since $(\ell_S)^{\alpha_S}$ is bounded by a constant (see Equation IV.28), time complexity of the algorithm is $O(n^2)$.

2.4 Computational Results

2.4.1 Case Study

Several experiments were performed to evaluate the performance of the described approach in order to validate its accuracy and scalability comparing it with other high-performance IR systems. We apply our method to three FAQs from different domains.

The data

- *Restaurant FAQ*: This FAQ belongs to a Restaurant and it has just 39 entries. The FAQ entries inform about reservations, prizes, menus, etc. It contained a total of 400 reformulations.
- *Linux V.2.0.2 FAQ*: This FAQ is a public list of questions [13] about the operating system Linux. It contained a total of 59 entries and 450 reformulations that were obtained from users.
- *UGR FAQ*: This FAQ and the corresponding reformulations, were generously supplied to us by the University of Granada (UGR). Those reformulations were obtained from the execution logs of the Virtual Assistant on the UGR web page[14]. 5000 questions introduced by students, professors, and administrative staff in the first year of life of the system, were ramdomly selected to perform these experiments.

[11] Note that instead of using 10^3, the upper bound could be set to $\lceil average(\ell_{S_i})\rceil^3$, where $average(\ell_{S_i})$ is the average length of sentences in F.

[12] Check whether an element pertains to a given set is achieved in constant time by using hash tables

[13] `http://www.linux-es.org/Faq/Html/`

[14] `http://tueris.ugr.es/elvira/`

We construct equal proportion and size subsets for each 10-fold cross validation. The subsets were randomly selected using MATLAB 7.1. Table IV.6 shows the details of the data sets used in this experiment: columns display the number of FAQ entries, the number of reformulations, the number of training and testing set, and also the average of reformulations for each FAQ entry (Average Ref/Entries).

	FAQs	Reformulations	Density(Ref/Entries)	Training Sets	Testing Sets
Restaurant FAQ	39	400	10.26	360	40
Linux V.2.0.2 FAQ	59	450	7.63	400	45
UGR FAQ	310	5000	16.13	4500	500

Table IV.6: Details of the FAQs used in the experiment.

The original version of the FAQs, the set of reformulations, and the subsets used to carry out the 10-fold cross validation in our experiments, are accessible in[15].

Performance of the experiments The model is tested by performing a 10-fold cross-validation for each FAQ. Ten complete validations are performed in order to obtain sufficient results to confirm the results with a t-test [Leh12]. In each 10-fold cross validation, the entire data set is divided into ten mutually exclusive subsets with the same distribution. Each fold is used once to test the performance of the classifier generated from the combined data of the remaining nine folds.

In this study, we evaluate the classification performance of the models by most popular FAQ retrieval evaluation metrics *accuracy* and *Mean Reciprocal Rank* (MRR) —a different evaluation using IR metrics is later offered in section 4. Accuracy measures the proportion of correctly classified cases from among all of the classified cases (see Equation IV.29). MRR is a measure for evaluating any process that produces a list of possible responses to a query, ordered by probability of correctness. The reciprocal rank of a query response is the multiplicative inverse of the rank of the first correct answer. The MRR is the average of the reciprocal ranks of results for a sample of queries (see Equation IV.30).

$$Accuracy = \frac{|\{t \in T | rank(t) = 1\}|}{|T|} \tag{IV.29}$$

$$MRR = \frac{1}{|T|} \cdot \sum_{t \in T} \frac{1}{rank(t)} \tag{IV.30}$$

Where T is the entire testing set and *rank(t)* computed the rank of the most relevant FAQ entry given by the t query.

For the biggest FAQ (UGR FAQ), we also developed a scalability study, which shows how the system works depending on the amount on training data. To this end, 11 reformulation sets have been generated. T_0 contains just one question for each FAQ entry. The remaining reformulations have been partitioned into 10 subset with the same distribution, $T_1, \ldots, T_{10}$. The subsets were randomly selected. $T_{training}$ will be the training set that will be different and bigger in each test. In the first test, $T_{training} := T_0$, thus, no reformulations are being considered. In the remaining tests, one different subset is added to $T_{training}$. In this way, in the ith test, $T_{training} := T_0 \cup T_1 \cup$

[15] `http://decsai.ugr.es/~moreo/publico/FAQRetrieval_dataSets/MDEs_FAQ_Retrieval_Datasets.html`

$\ldots \cup T_{i-1} \cup T_i$. For the testing set ($T_{test}$), a random set of reformulations, not included in any training set, was selected. The whole process has been repeated 10 times in order to decrease the dependency of the results on a concrete partition of the reformulation set.

Comparison Algorithms The performance of MDE algorithm is contrasted with the following models:

- tf·idf [SM86]: This method measures the similarity between two word frequency vectors $\overrightarrow{U}$ and $\overrightarrow{S}$. This scoring function weights each word by its term frequency (*tf*) and its inverse document frequency (*idf*).
- Adaptive tf·idf [SM86]: It is an improved version of the classical tf·idf. The difference consists of performing a hillclimbing for each weight to bring a question and its corresponding answer "closer", raising the score function.
- Query Expansion [BCC$^+$00]: This model aims to bridge the lexical chasm between questions and answers. In the context of document retrieval, query expansion involves adding words to the query which are likely synonyms of (or at least related to) words in the original query. These words are added by calculating the mutual information between query terms and answer terms in the training set.
- FRACT [KS08b]: It is a cluster-based retrieval system. It clusters the logs into predefined FAQ entries and extracts weight scores of potentially occurring words from the clusters by using a centroid finding method based on LSA techniques. It represents FAQs and query logs in latent semantic space and uses the vector-similarity function to compute the closeness scores between FAQs and query logs. Then, the clusters are used as a form of document smoothing during retrieval.
- Co-occurrence Model [Jua10]: This method takes advantage of word co-occurrence corpus (semantic model) to improve its ability to match questions and answers through a question similarity measurement. Similarity is based on the number of relative terms and the length of the query sentences.
- Rough Set Theory [DYC08]: This algorithm combines hierarchical agglomerative clustering method with rough set theory to address the problem of FAQ retrieval. The lower/upper approximations to a given cluster are used to classify users queries.

Setting Parameters To select parameters γ_1, γ_2, and β, we carried out some preliminary experiments. Thus, those parameters were empirically verified as the most suitable for our datasets. We set $\gamma_1 = 0.5$ and $\gamma_2 = 0.25$. Since γ_1 and γ_2 weights intermediate inclusion degrees, the algorithm presents low sensitivity to variations on them —less than 1% in terms of accuracy in our experiments using other values. In contrast, the algorithm is more sensitive to variations in parameter β that models the influence of *EntryRelevance* and *DomainRelevance*. In this regard, we verified that *DomainRelevance* should dominate the weighting criteria. In our experiments, it was found that by setting $\beta = 0.25$ the algorithm obtained a better performance in all our datasets —it gained approximately 6% in accuracy in contrast to $\beta = 0.5$. However, it should be pointed out that those parameters are domain and language dependant.

Regarding the comparison algorithms, we have taken same parameters reported by their authors in their original papers. We implemented all these methods and MDE algorithm in Java: 1.6.0. In order to implement FRACT, we used *Jama*, a basic linear algebra package for Java. An Intel(R) Core(TM)2 Duo E7400 2.80GHz with 4GBytes RAM was used to carry out the tests.

2.4.2 Results of the study

Table IV.7, Table IV.8 and Table IV.9 show a summary of the accuracy and MRR measurements for each FAQs after carrying out the experiments.

Measure	1	2	3	4	5	6	7
Accuracy	MDE	Ad. tf·idf	tf·idf	Q. Expansions	Co-Model	FRACT	RoughSet
	0.8236	0.7481	0.7452	0.7364	0.6843	0.6795	0.6745
MRR	MDE	Ad. tf·idf	tf·idf	Q. Expansions	Co-Model	FRACT	RoughSet
	0.9138	0.8493	0.8470	0.8328	0.7820	0.7526	0.7454

Table IV.7: Ranked ordered accuracy of each method in Restaurant FAQ

Measure	1	2	3	4	5	6	7
Accuracy	MDE	Ad. tf·idf	tf·idf	FRACT	RoughSet	Q. Expansions	Co-Model
	0.7226	0.7087	0.7060	0.6707	0.6633	0.6515	0.6390
MRR	MDE	tf·idf	Ad. tf·idf	FRACT	Co-Model	Q. Expansions	RoughSet
	0.8661	0.8081	0.8076	0.7549	0.7548	0.7513	0.7417

Table IV.8: Ranked ordered accuracy of each method in Linux FAQ

Measure	1	2	3	4	5	6	7
Accuracy	MDE	FRACT	Co-Model	Ad. tf·idf	tf·idf	Q. Expansions	RoughSet
	0.8059	0.7547	0.5969	0.5475	0.5466	0.5371	0.5520
MRR	MDE	FRACT	Ad. tf·idf	tf·idf	Q. Expansions	Co-Model	RoughSet
	0.8853	0.8109	0.7211	0.7205	0.7087	0.6852	0.6325

Table IV.9: Ranked ordered accuracy of each method in UGR FAQ

As can be observed in these tables, our model exhibits the highest global average in terms of accuracy and MRR in the three FAQs. Query Expansion obtained low accuracy in contrast to tf·idf because of overtraining. Co-occurrence Model performs well in terms of accuracy, but other comparison methods are better in MRR. Since the size of hierarchical agglomerative clusters obtained by RoughSet model is usually too big, the MRR score obtained is low. In light of the results, in small domains such as Linux or Restaurant FAQs, MDE algorithm performs similarly to other methods. However, as the number of reformulations increases, MDE algorithm reaches higher accuracy score than the other comparison methods. As a result, MDE algorithm works better in large case bases.

In order to verify the statistical significance of these results, we applied a t-test with a confidence level of 95% obtaining p-values < 0.005 in all cases. Finally, Figure iv.3 display the scalability results in UGR domain in terms of accuracy and MRR. The regression lines show how accuracy and MRR increase as the training set size gets larger. Although Query Expansion obtained low accuracy and MRR scores in our experiments, it shows the best slope in the scalability study with regard to the other comparison methods. However, the MDE algorithm reaches the highest slope, which is 1.24 times better than the best of the others methods' slopes.

It should be pointed out that FRACT shows better performance than MDE in terms of accuracy while the training set size is lower than (approximately) 2750 in the UGR domain. However, MDE algorithm outperforms FRACT algorithm for larger training set size, thus MDE presents better scalability. As depicted in Figure iv.3 (left), scalability lines of MDE and FRACT intersect, thus FRACT performs better in the first stages of the scalability study. However, according to Tables IV.7 and IV.8 MDEs performed better in smaller domains with a similar reformulations/entries proportion to the one in earlier steps of the scalability study. This indicates FRACT is more sensible to the number of documents (FAQ entries) than to the size of these documents (reformulations). MDE has proved to work more robustly in small domains and also to take better advantage of the collection of reformulations in large domains in the long term.

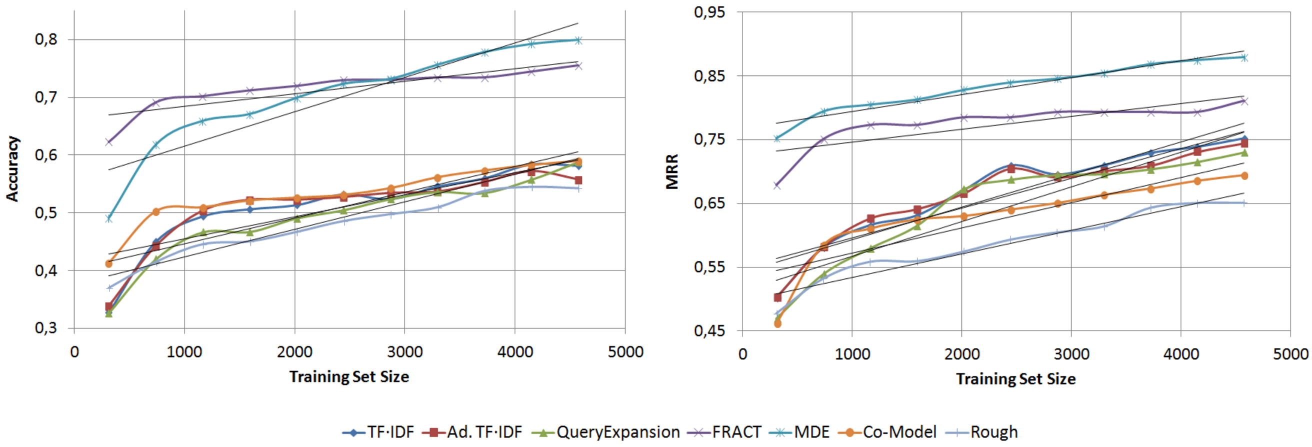


Figure iv.3: Accuracy Scalability study (left) and MRR Scalability study (right).

2.4.3 Study of the Efficiency of the Training Algorithm

Discussions on space and time complexity Theoretically, the space complexity of the MDE algorithm is asymptotically bounded by $O(n)$, and the time complexity is bounded by $O(n^2)$. This section is devoted to analyse and discuss our empirical results in this regard. To this end, we reused the incremental datasets taken from the UGR domain (see section 2.4.1). Figure iv.4 represents the efficiency of our model in terms of number of MDEs obtained (left) and time spent in the calculation (right).

As can be observed, the empirical results confirm the theoretical time complexity. However, the tendency of the number of MDEs obtained seems to fall below the linear function. The reason is that while the set of reformulations is being increased in each subsequent set of the scalability study, the number of FAQ entries remains constant. Thus, the proportion of relevant MDEs to the number or reformulations tends to decrease in the long term. In our experiments, the $|MDEs|/n$ proportion decreases from 7.53 ($|T_{training}| = 310$) to 3.59 ($|T_{training}| = 5000$).

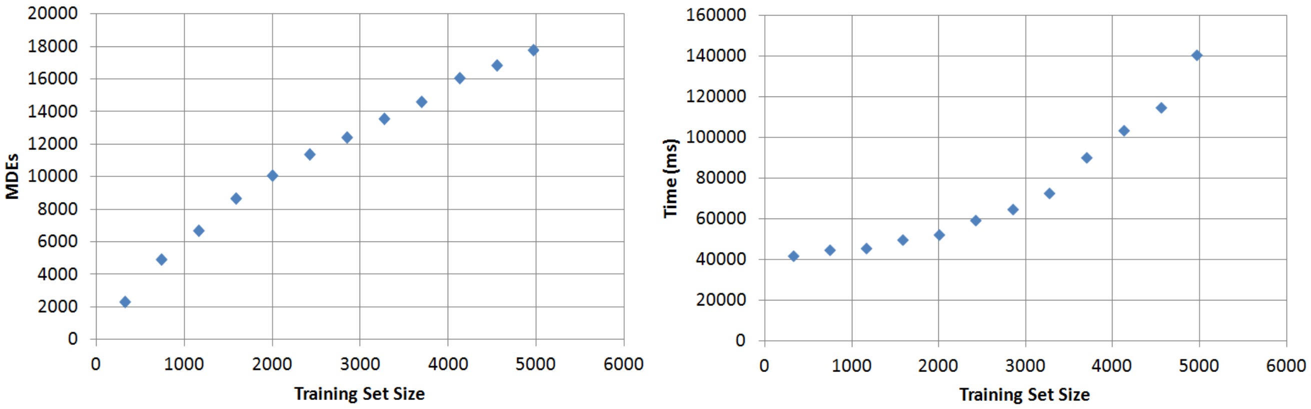


Figure iv.4: Efficiency of the MDE algorithm: Space complexity (left) and Time complexity (right).

Deep search using Dynamic α: Undeniably, α bound plays a key role in the trade-off between efficiency and accuracy. We have studied the dependency of the algorithm on this parameter by setting different constant values for α bound in the UGR dataset (Table IV.10). In light of the results, it seems that *Accuracy* and MRR measures are established for $\alpha \geqslant 3$. Indeed, we have calculated that, in our experiments, between $60 - 75\%$ of the MDEs contained only 2 words, less than 20% contained more than 2 words, and less than 4% contained only one word (this also indicates that using multi-word expressions is better than using isolated keywords to compute the similarity measure). Dynamic α allows the system to maintain a deeper exploration in concise reformulations and avoid efficiency penalty in the exploration of larger sentences. For this reason, using the dynamic version of α incurs in a faster calculations maintaining the *Accuracy* and MRR.

α	$Time$	$\lvert MDE\rvert$	Acc	MRR
1	16162	3404	0.566	0.603
2	73214	14823	0.8	0.842
3	706890	21769	0.801	0.856
4	694573	21769	0.804	0.885
dynamic	196101	18204	0.804	0.882

Table IV.10: Effect of α

Evaluation of minimality and differentiability pruning criteria In order to validate the minimality and differentiability pruning criteria, we have contrasted the impact in the growth of the candidate expressions set contained in the search space by considering different pruning criteria (Table IV.11). First column represents the number of reformulations. The second column represents the number of candidate expressions considering a brute force exploration —every combination of words is a candidate expression, so the growth progression is given by $n \cdot 2^{\ell}$. The third column shows the number of candidate expressions by applying a constant $\alpha = 4$ bound —its growth curve is $n \cdot \binom{\ell}{\alpha}$. Fourth column shows the number of expressions considered while applying the dynamic α threshold and, finally, the last column corresponds to the prune applying the minimality and differentiability criteria along with the dynamic α threshold. This result evidences the efficiency of our pruning criterion based on minimality and reflects its impact on the search space complexity.

n	brute force	constant α	dynamic α	MDE
310	317440	65100	12917	2336
733	750592	153930	30738	4953
1153	1180672	242130	49370	6700
1573	1610752	330330	68491	8665
1993	2040832	418530	87429	10093
2414	2471936	506940	106653	11406
2837	2905088	595770	125319	12464
3258	3336192	684180	142828	13572
3691	3779584	775110	162225	14619
4116	4214784	864360	181352	16073
4540	4648960	953400	198960	16876
5000	5076992	1041180	218371	17805

Table IV.11: Effect of the pruning on the search space

2.5 Conclusions and Future Work

In this research, a new question answering approach dealing with FAQ retrieval problem has been proposed. Our method has outperformed all the comparison algorithms in small domains as well as in large domains with a high number of reformulations. In this chapter, we intended to deal with some problems presented in the closed-domains NL approaches where the KB is structured and no additional KR is available. Those problems included the the scalability of the algorithm, and the efficiency in the search for incremental systems.

MDE algorithm does not require any metainformation (choosing keywords, linguistic rules, patterns, or so on) so it can be easily adapted to new domains. MDE algorithm improves with the use since its performance relies mainly on the collection of reformulations. This task is much less costly than knowledge modelling ones because it does not require any expert support.

Unlike in statistical classifiers or latent term weight models, the multi-word expressions calculated by our algorithm are directly extracted from questions and could be manually revised by an expert to achieve a more reliable system (further discussions about this claim will be later offered in section 4).

In view of the results our algorithm performs well in terms of scalability. Since its classification criterion is based on differentiation, it becomes more and more reliable as the number of reformulations and FAQ entries increases. The precision of the extracted expressions is directly related to the exhaustivity of the reformulations sets. With the addition of new reformulations, noisy syntactic interferences between words are broken. Regarding this, it can be concluded that in the limit, supposing an exhaustive (theoretical) set of reformulations, the obtained expressions would be optimal.

Finally, minimality and differentiability search criterion has proven to be an efficient prune regarding the search space. Although space complexity is theoretically linear, in practice it presented an under linear behaviour in real FAQs. The algorithm is self-adjusted by using a dynamic exploration threshold allowing the time complexity to be quadratically bounded. As a result, it could be said that our search criterion guides the exploration in an efficient manner.

In the present research, a new algorithm that takes advantage of the differentiation information between sentences has been proposed. In this research, some parameters were manually set (β, or intermediate inclusion degrees in function $m(S, e)$). Those parameters intend to model the influence in the meaning of a given sentence depending on the order of words and semantic of terms. Influence of the order of words is a language dependant issue, while the semantic variation between semantically related words is a domain dependant issue. In future research we plan to automatically obtain the optimal parameters to a given language in a given domain.

In following chapters, we will address the problem of how to automatically detect knowledge gaps by usage mining (section 3), and how those sets of regular expressions could be better presented in order to diminish the costs that an expert should undertake to manually improve the performance in critical systems (section 4).

3 FAQtory: A Framework for Collaborative FAQ Retrieval Systems

3.1 Introduction

As commented before, FAQ lists represent an interesting Customers Relationship Management (CRM) strategy to overcome the costs of call centers. However, according to [Sne99], those lists present several deficiencies. The user is forced to explore the entire list in order to find a relevant question. Moreover, the information of interest may be mixed up in several questions, or even not exist in the FAQ. Despite the amount of effort that has been devoted to investigate FAQ retrieval methods, how to create and maintain high quality FAQs has received less attention. To the best of our knowledge, the only directly related researches in this regard are [HYJ10], a semi-automatic method to assist managers in constructing a forum FAQ, and [Yan09a], a collaborative FAQ system in learning community. The first of them analyses similarity among forum questions in order to identify FAQ entries. The second one, relies on the active interaction generated by the learners community following a QA scheme. In contrast, our aim is to assist the FAQ managers by exploiting users feedback and usage information. In this chapter we investigate effective methods to improve the domain coverage in systems with structured knowledge and without any meta-knowledge resource. This study is materialized in a web application called FAQtory [MRCZ12a] —an entire framework to create, manage, and use intelligent FAQs.

FAQtory has been designed with a continuous learning cycle that allows the managing of the usage feedback to semi-automatically improve the quality of FAQs. In FAQtory there are two types of users: customers and FAQ managers. On the one side, the system allows customers to automatically find relevant information by typing NL queries. Customers are also able to report whether the information requested was not (entirely) contained in the FAQ. On the other side, FAQ managers have access to this explicit feedback to improve the FAQ quality by adding or modifying FAQ entries. Furthermore, *Usage Reports* are automatically submitted to FAQ managers in order to assist the maintenance of their FAQs. Those reports are generated by means of *Usage Mining* techniques including *usage descriptions*, *weaknesses detection*, and *knowledge gaps discovery*. *Summarization* and *Visualization* tools are used to show the reports in an interpretable and meaningful manner. Thus, the performance of FAQtory improves with use. Since we are not considering any kind of meta-knowledge resource, FAQ managers role in FAQtory consists of just adding or modifying FAQ entries. By taking advantage of unattended methods, no further modelling task is needed.

Traditional FAQ maintenance is carried out manually and depends only on expert (FAQ manager) knowledge. Our main objective consists of identifying actual users' information needs to be reflected in the FAQ. To achieve this goal, following maintenance mechanisms are proposed (see Figure iv.5): (i) Explicit Feedback: allows users to explicitly propose new questions, (ii) FAQ analysis: evaluates the current usage of the FAQ in order to assist FAQ managers in the maintenance, and (iii) Knowledge Gaps discovery: automatically analyses users questions to identify most important knowledge gaps in the FAQ. In this way, maintenance of the FAQ is no longer directed only by expert knowledge, but also by actual users' requirements in an emerging collaborative scenario.

All characteristics described incur in undeniable positive repercussions for both customers and companies. Customers are allowed to solve their queries in an intuitive and efficient manner avoiding the compulsory reading of the whole FAQ list and also avoiding a keyword-based query. At the same time, companies make a profit from the system thanks to the reduction of costs related to the maintenance of call centers while offering their clients a high-quality information access. In addition, FAQtory helps to bridge a new interaction between customers and FAQ managers by explicit feedback from customers.

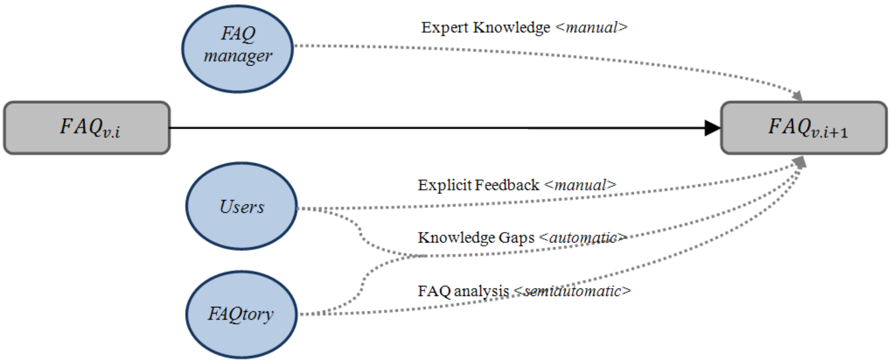

FAQ manager
Expert Knowledge <manual>
FAQv.i
FAQv.i+1
Users
Explicit Feedback <manual>
Knowledge Gaps <automatic>
FAQtory
FAQ analysis <semiautomatic>

Figure iv.5: FAQ maintenance

The rest of this chapter is organized as follows: An overview of previous work on FAQ retrieval applications is presented in section 3.2. Section 3.3 presents the architecture of the proposed system. In Section 3.4, the FAQ retrieval method used will be explained. In Section 3.5, how FAQ managers are assisted to improve their FAQs is discussed. Finally, Section 3.6 describes our experiences with the system, and Section 3.7 concludes with a discussion of the framework and future research.

3.2 Related Work

In this section, the main related work to our approach are discussed. Current approaches related to our system are discussed. This discussion covers the following areas: methods requiring knowledge modelling, methods that do not require knowledge modelling, and systems designed to create and manage FAQs.

Recent works could be classified into two types: those that require meta-knowledge modelling, and those that do not. Often, meta-knowledge is modelled through domain ontologies[16]. In this line, [Yan09a] proposes an Interface Agent based on ontology-directed and template-based models with the aim of capturing the user's intention in the query. Other works in IR dealing with user modelling architectures could be consulted in [LRM03, UGB05]. [HSCD09] focuses in expanding their ontology with the changing information providing an assistance mechanism able to create appropriate answers if none of the existing ones is relevant. Following this strategy, in [LLL10] new questions are manually annotated and added to the system each time the similarity score does not exceed the threshold. They rely on question-query patterns as well as on ontological models to address the problem of QA based on FAQs. In [CHWC05], the questions in the FAQ are classified into ten question types. The answers in the FAQ are clustered using Latent Semantic Analysis (LSA)[TLL98] and K-means algorithm. An ontology based on WordNet and HowNet is used for semantic representation of the aspects. The retrieval process is considered as a maximum likelihood estimation problem in a probabilistic mixture model.

Although those methods provide accurate answers to user queries, they imply many knowledge modelling. In order to overcome this disadvantage, statistical FAQ retrieval methods that perform without high-level knowledge bases have been proposed. FRACT [KS08a, KS06] is a clustered-based system that performs a LSA on query logs. Those query logs present the advantages that they are easy to collect and they cover a larger language. OPTRANDOC [KAE11] is a self-learning algorithm that automatically modifies the set of keyword terms using TF·IDF considerations by

[16] http://www.w3c.org

taking advantage of the feedback from users queries. It runs a genetic algorithm to implement the IR engine. Our method Minimal Differentiator Expressions algorithm [MCLZ10, MNCZ12], presented in section 2 is also a domain-independent method.

Although the FAQ retrieval problem received considerable attention, how to construct and maintain high-quality FAQs did not. [HYJ10] proposed a semi-automatic method to identify and reduce similar questions, posted in Open Source Projects forums, in order to assist forum managers in constructing the FAQ. In this way, the volume of similar questions in the forums FAQ can be reduced preventing forum members from wasting time on answering questions that were already solved. In [Yan09a], a system devoted to online community learning is proposed. This system is a knowledge share platform structured as a FAQ. This FAQ is continuously enriched through Q&A interaction generated by the community members (learners and instructors). A new question is proposed to the system each time the FAQ retrieval algorithm is not able to retrieve useful information. In a similar way, users in our system are allowed to indicate whether their information needs were not satisfactorily solved. What makes our proposal different is that FAQ managers in our system have usage information reports at their disposal. Those reports are automatically generated and provide FAQ managers with useful information to monitor the FAQ performance so as to modify it conveniently if needed.

3.3 The FAQtory Framework, an Overview

In conventional FAQ retrieval systems, customers are allowed to express their queries in a natural manner by typing questions in NL. The system uses the FAQ retrieval algorithm to search for the most related questions to the user query. Then, a ranked list of Question/Answer pairs is given to the user, according to their relevance to the question. In this way, customers are not forced to manually explore the FAQ list. However, our aim is to go beyond the functionality of traditional FAQ retrieval systems, providing a framework to create, maintain, and improve the FAQs quality. To this purpose, FAQtory exploits the feedback generated by its use to assists FAQ managers to improve their FAQs. In this section, the overview of the entire framework is discussed.

First of all, we explain the functionality of the system. Later, the main modules of the system and their interoperability are described.

3.3.1 Improving with the use

There are two clearly differentiated roles in the system: customers and FAQ managers. Customers access the system to solve their queries. The role of FAQ managers consists of creating and organizing the FAQs lists. In this section, we explain how the system monitors the FAQ managers by means of usage information.

As depicted in Figure iv.6, two Front-ends have been designed. In *Customers Front-end* Customers can type their questions in NL. This front-end allows customers that are not satisfied with the retrieved information to indicate whether their question was not successfully solved (*explicit feedback*). Simultaneously, the *Usage Mining* module generates *Usage reports* by providing statistical information related to the usage of the FAQs, for example which questions were often retrieved but rarely read (further explanation on this module shall be seen in Section 3.5). By means of *Visualization and Summarizing tools*, FAQ managers get in a meaningful manner the information contained in explicit feedback and usage reports. By analysing those data, FAQ managers are provided with enough information as to edit their FAQs and so improve their performance. Our FAQ retrieval algorithm gets retrained every time FAQs are edited and those changes are learned for further queries.

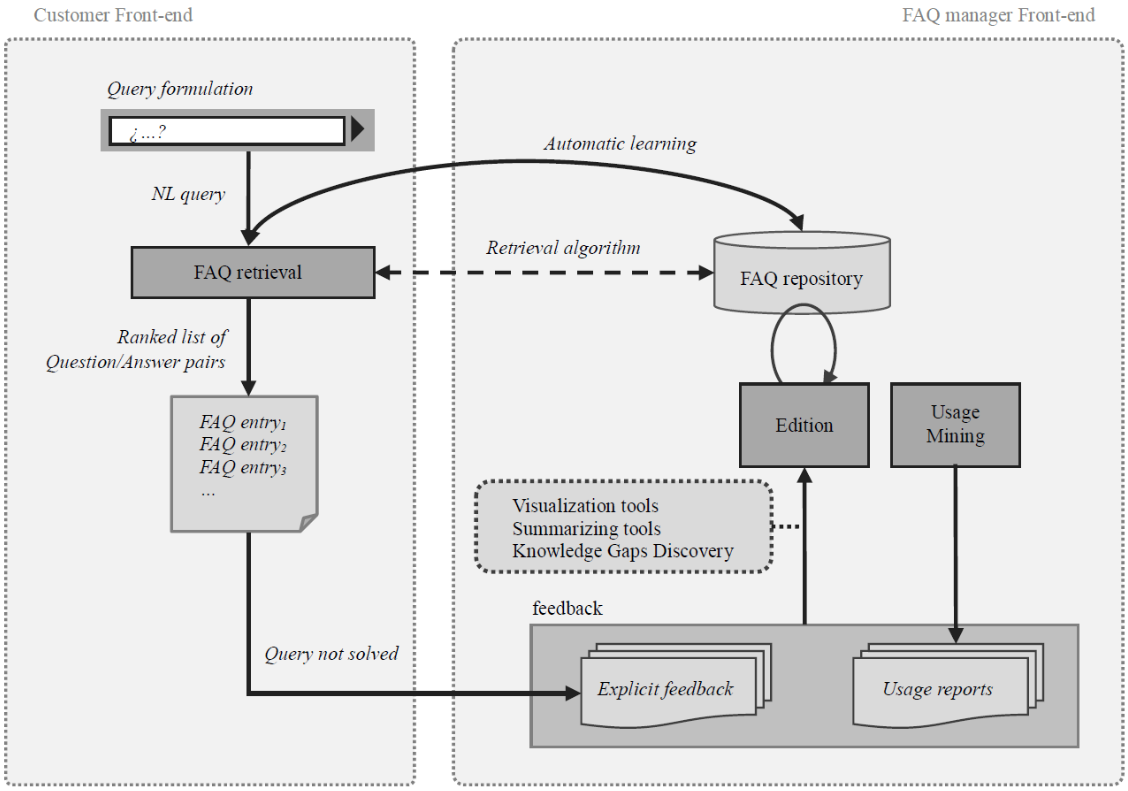


Figure iv.6: Functionality diagram.

3.3.2 Main modules

In this section we describe in detail the key modules that compose our entire system. Figure iv.7 depicts those modules and the interoperability among them. Since both FAQ retrieval algorithm and Usage Mining are the most relevant and complex modules, they will be explained in Section 3.4 and Section 3.5 respectively.

1. FAQ editor: provides access to FAQ managers to edit their FAQs. Main editing options include adding FAQ entries, modifying questions and/or answers, and setting up details such as FAQ description, release date, language, and so on. Answers can be accompanied by multimedia resources such as videos, hyperlinks, or attached files. As our FAQ retrieval engine is not supported with meta-knowledge, FAQ managers are requested to add rephrases to FAQ entries through the editor. The more rephrases added, the better the performance of the FAQ retrieval engine would be. They can also cluster related FAQ entries defining sections.

2. User Management: this module manages all kinds of users establishing the functionality for each type. Customers interact with the system in the role of registered user or non-registered user. Registered users can request access to private FAQs as well as use the notification system while non-registered users are not allowed to access this functionalities. FAQ managers can

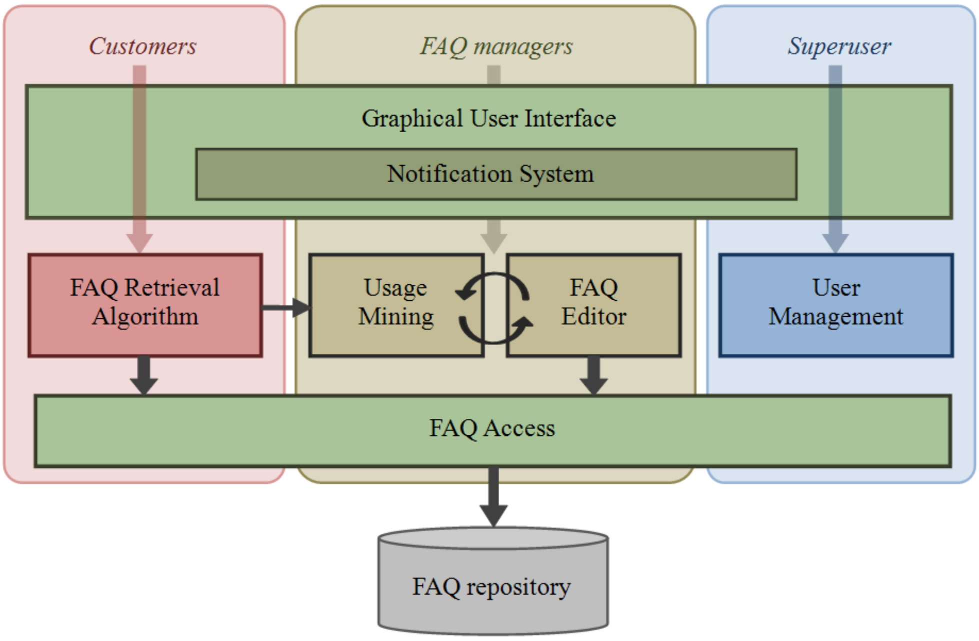


Figure iv.7: Module diagram.

access the visualization tools, edit their FAQs, and also validate access to their private FAQs. Finally, the Superuser is the system administrator that assigns the role of FAQ manager to those registered users that request it.

3. Notification System: this system brings registered users the option to communicate between them as well as a direct interaction between customers and FAQ managers. System also delivers automatically generated information to FAQ managers such as new access requests to their private FAQs.

4. Graphical User Interface (GUI): besides the common characteristics expected in this kind of web applications dealing with user-friendliness, usability, error tolerance, and so on, our interface implements additional visual functionalities. In fact, beyond classical viewing of FAQs lists, the user has the option of inspecting the FAQ clustered in sections previously defined by FAQ managers. This feature leads to a more direct and efficient exploration for users that prefer a manual search rather than delivering a question in NL. Furthermore, FAQtory can easily be embedded in any corporate website avoiding their customers to be redirected to our platform. Embedding options also permit a high level of visual customization.

5. FAQ access: implements the security restrictions while accessing the FAQ repository in the server-side. This module manages safety access to private and public FAQs.

3.4 FAQ Retrieval Algorithm: the Continuous Learning Cycle

FAQtory will contribute a great deal to the company insofar as it is useful for their customers. In such a context, finding the best related questions to user query is a very important issue. Thus, the retrieval algorithm is a fundamental part of the framework. As commented before, since FAQ managers in our framework are not intended to be experts in IR techniques, they are not requested to define keywords, templates, ontologies, or so on. Instead, they are only requested to add questions, reformulations, and answers to their FAQs. Thus, the retrieval algorithm relies only on the domain knowledge, and does not count with any meta-knowledge at its disposal.

To implement the FAQ retrieval engine, we have considered two different algorithms that do not require knowledge modelling: MDE algorithm (see section 2) and the TF·IDF method. Each of these algorithms presents a number of strengths that complement each other to compose a more robust FAQ system. MDE has proven to perform well in domains of different size while TF·IDF is extremely efficient in terms of training time. Thus, while the FAQ is being edited the system gives priority to the TF·IDF criterion. Otherwise, if the FAQ is stable, the system gives priority to MDE algorithm. Nevertheless, this mechanism is transparent to customers that raise questions through the interface. Regardless of the algorithm in charge, most related QA pairs are presented to them as a list sorted by similarity (Figure iv.8).

There are further reasons to combine those algorithms. Occasionally, users tend to formulate their queries in a simplistic manner, supposing the system performs a keyword-based search. They pose only those few words that consider to be relevant to the FAQ. Since such queries are not actually NL, MDE algorithm may not be able to retrieve any FAQ entry. For example, if the user poses only one word, and that word appears in several FAQ questions, then the differentiation criterion is not enough. In such a context, TF·IDF is more reliable than MDE algorithm. However, TF·IDF does not attend to the semantic information of the words itself. In contrast, MDE incorporates mechanisms to take semantic words relations through WordNet and co-occurrence implications into consideration. Thus, MDE is more reliable than TF·IDF if users pose NL sentences. Furthermore, none of these algorithms is restricted to any particular language. Indeed, we have validated the system in English and Spanish languages.

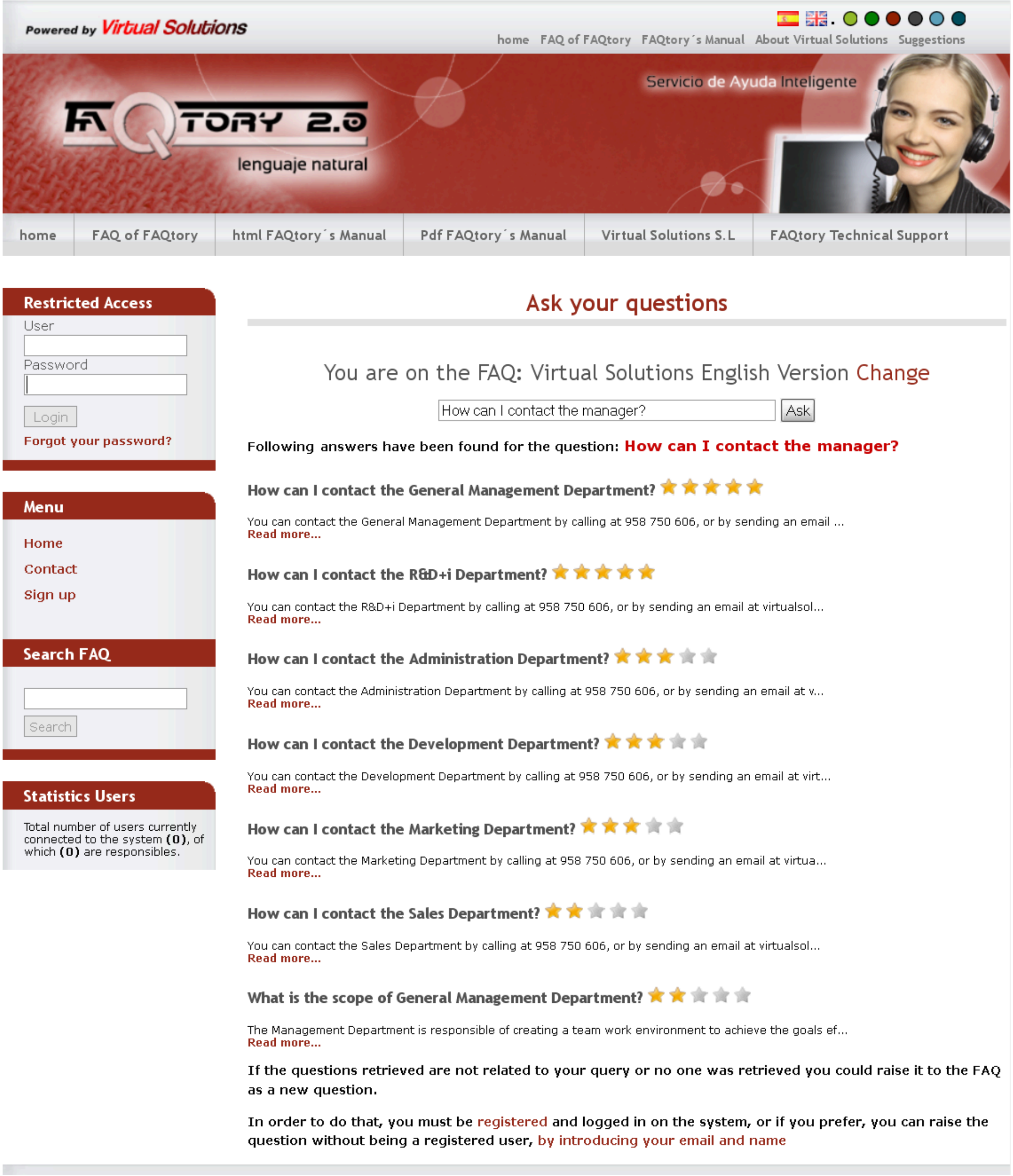
Since FAQs are dynamic, FAQtory carries out a continuous learning cycle to maintain the retrieval algorithms updated. As this process is automatic, FAQ managers remain unaware of it.

Assuming the length of sentences to be constant, MDE algorithm present a quadratic order of efficiency (see section 2.3.5). Retraining in TF·IDF consists of precalculating each word frequency in the documents. Figure iv.9 depicts execution times taken from both algorithms while training FAQs of different sizes.

It takes approximately one minute for the MDE algorithm to train FAQs containing about 2000 reformulations. Training time is significantly lower in TF·IDF than in MDE algorithm (less than a second in the same case). For this reason, TF·IDF is automatically retrained after each FAQ edition. Whenever a FAQ is modified, the MDE retrieval algorithm is temporarily blocked until its next retraining. Training algorithm in MDE is launched periodically for each modified FAQ. Finally, if FAQ manager attends a question proposed by a customer (explicit feedback) then a notification including the new answer is sent to the customer. Figure iv.10 depicts the continuous learning cycle followed by FAQtory.

3.5 Usage Mining

FAQ managers design their FAQs according to what they consider relevant. Moreover, the information contained, the FAQ structure, and the specific wording of the questions are highly biased to the knowledge of the FAQ manager. Starting from the premise that knowledge about the domain may be different for customers and FAQ managers, it is likely that: (i) relevant information to customers remains out of the domain, or (ii) the FAQ gathers a plenty of entries that are actually useless for customers. Thus, regardless of the FAQ retrieval algorithm performance, it is undeniable that



Total number of users currently connected to the system (0), of which (0) are responsibles.

Ask your questions

You are on the FAQ: Virtual Solutions English Version Change

How can I contact the manager? Ask

Following answers have been found for the question: **How can I contact the manager?**

How can I contact the General Management Department? ★★★★★

You can contact the General Management Department by calling at 958 750 606, or by sending an email ...
Read more...

How can I contact the R&D+i Department? ★★★★★

You can contact the R&D+i Department by calling at 958 750 606, or by sending an email at virtualsol...
Read more...

How can I contact the Administration Department? ★★★☆☆

You can contact the Administration Department by calling at 958 750 606, or by sending an email at v...
Read more...

How can I contact the Development Department? ★★★☆☆

You can contact the Development Department by calling at 958 750 606, or by sending an email at virt...
Read more...

How can I contact the Marketing Department? ★★★☆☆

You can contact the Marketing Department by calling at 958 750 606, or by sending an email at virtua...
Read more...

How can I contact the Sales Department? ★★☆☆☆

You can contact the Sales Department by calling at 958 750 606, or by sending an email at virtualsol...
Read more...

What is the scope of General Management Department? ★★☆☆☆

The Management Department is responsible of creating a team work environment to achieve the goals ef...
Read more...

If the questions retrieved are not related to your query or no one was retrieved you could raise it to the FAQ as a new question.

In order to do that, you must be registered and logged in on the system, or if you prefer, you can raise the question without being a registered user, by introducing your email and name

Figure iv.8: Retrieved information.

the IR process depends heavily on the FAQ quality. In this section, some techniques to help FAQ managers to improve their FAQs are discussed. Those techniques are based on the analysis of how users react after receiving answers by exploiting the information generated through click-navigation. Usage Mining techniques pursue the following two goals: (i) evaluate the actual performance of the FAQ, and (ii) discover current needs for new information.

Apart from questions explicitly proposed by customers, FAQtory automatically collects all the generated queries in order to analyse them. Furthermore, for each FAQ entry the following attributes

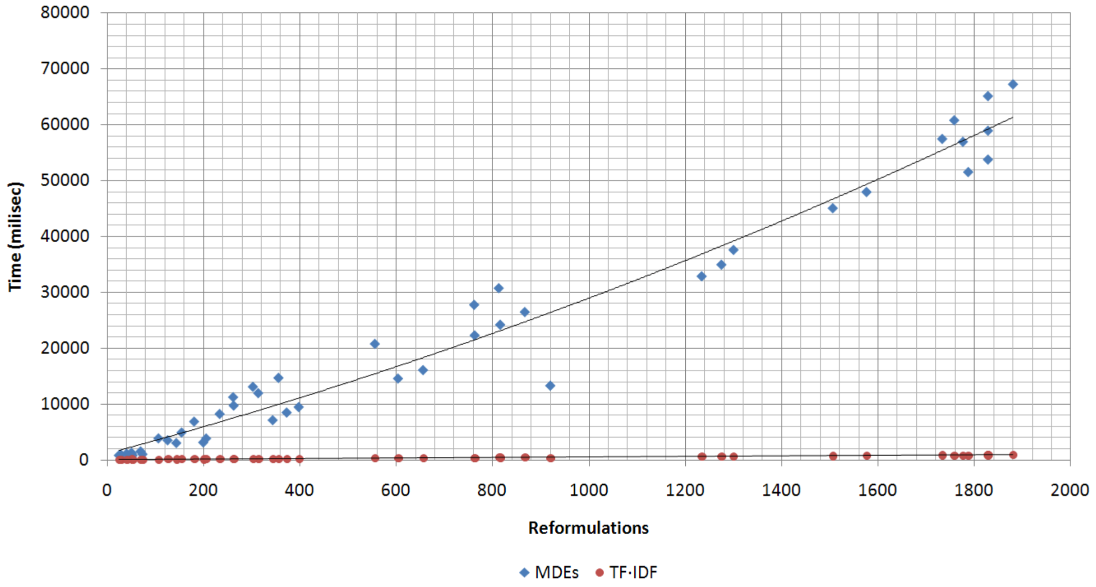


Figure iv.9: Training Time.

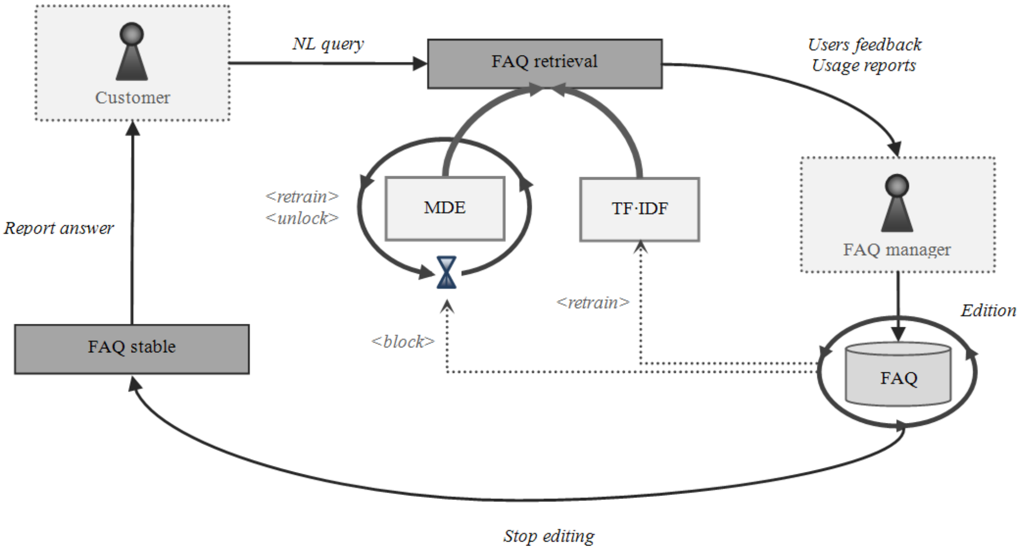


Figure iv.10: Continuous Learning Cycle.

are considered to generate reports:

- *#Retrieved*: counts the number of times a QA pair was retrieved.
- *%Retrieved*: percentage of retrieved times (*#Retrieved*) over the total number of queries posed to the FAQ.
- *#Read*: counts the number of times a given answer was read (clicked) after being retrieved.
- *%Read*: percentage of read times (*#Read*) over the total number of retrieved times (*#Retrieved*). This attribute indicates to what extent the retrieved question is useful to the user.

- *#Accessed*: total number of times the answer was accessed (clicked) either from the traditional FAQ list of after being retrieved.

Those attributes could be consulted in tabular form by FAQ managers. However, since the FAQ list could be large, tabular form may remain hardly interpretable. To override this problem, Visualization and Summarization tools have been implemented. First we describe how these tools could be used to identify FAQ weaknesses. Later, a method to discover specific knowledge gaps is discussed.

3.5.1 Visualization Tools

Visualization Tools display graphical explanations on the internal relationships between data. In order to develop these tools, we use Google Chart Api[17]. The information on accesses is grouped by FAQ sections and depicted through Pie Chart representation (Figure iv.11). This kind of charts provides an interpretable representation describing which FAQ categories contain the most interesting information to users. If FAQ sections are properly designed, FAQ managers are able to obtain useful information through this chart. For example, if a company organizes their FAQs in sections related to its products and services, they could observe which ones receive more attention from users. Analysing also the accesses to concrete FAQ entries, FAQ managers could know which aspects are relevant to customers (price, guarantee, technical service, or so on) and even take advantage of this knowledge in their marketing strategies.

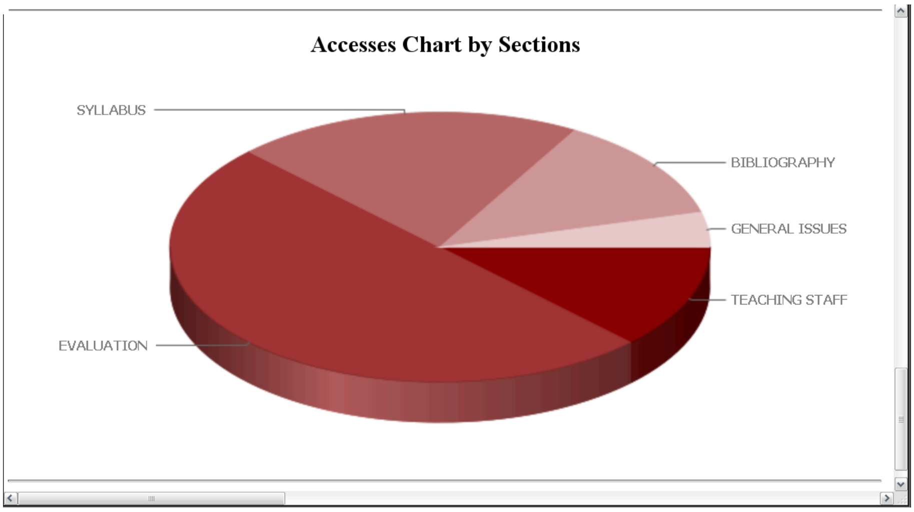
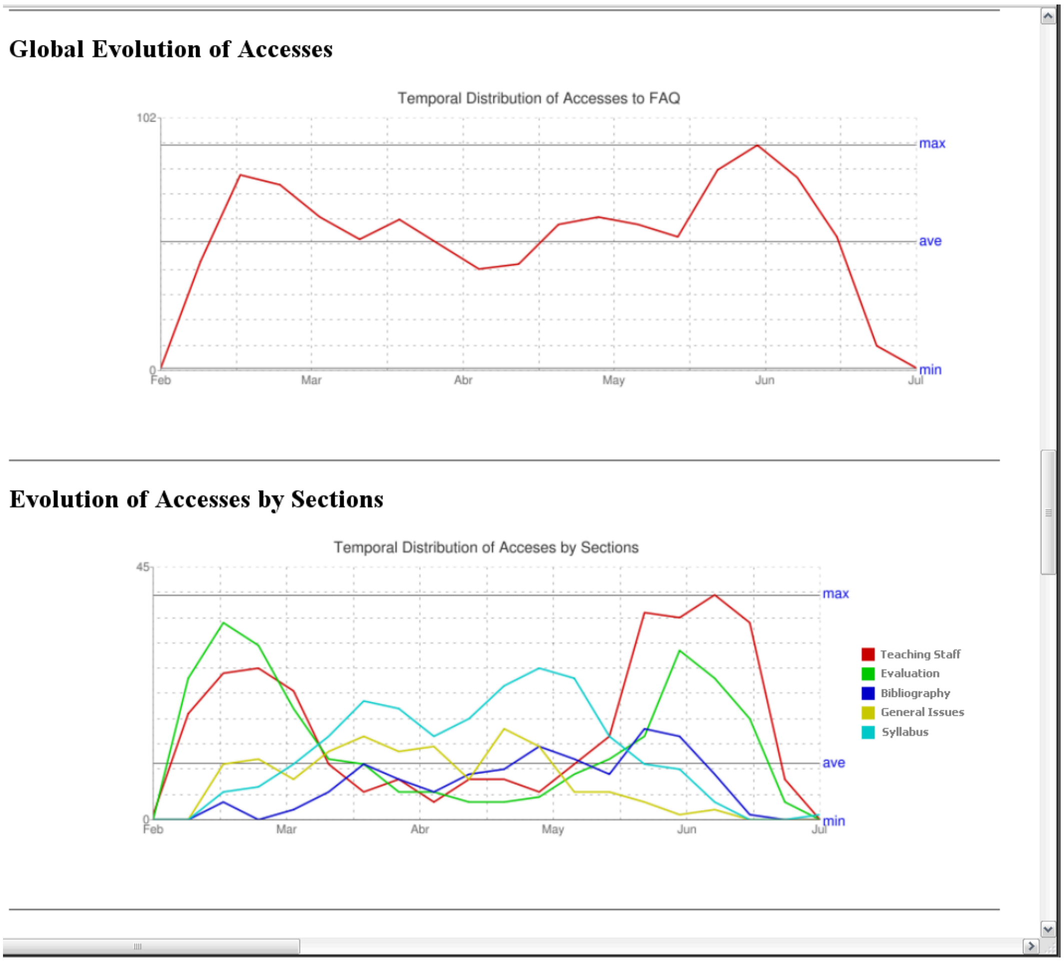


Figure iv.11: Pie Chart Graphic.

However, since customers interests may change over time, other graphical representations may be more useful. Date of access is incorporated to the analysis to capture how users interests evolve in a certain period of time. Time Chart representation (Figure iv.12) shows an example of accesses distributed over time. The FAQ concerns some issues relevant to a subject[18], including the following sections: General issues, Evaluation, Teaching staff information, Syllabus, and Bibliography. The figure on the top depicts the overall accesses to the FAQ, while the one on the bottom depicts the breakdown of the accesses by sections.

[17] `http://code.google.com/intl/en-EN/apis/chart/`

[18] *Knowledge Engineering* taught in the University of Granada from February to June



Global Evolution of Accesses

Evolution of Accesses by Sections

Figure iv.12: Temporal Chart Graphic.

3.5.2 Summarization Tools

Summarization Tools provide FAQ managers with Natural Language descriptions about analysis of accesses. The first tool lists questions that were rarely read after being retrieved. Concretely, a maximum of 10 questions whose *%Read* value do not exceed a threshold (20%) are listed sorted by *%Read*. Those FAQ entries are considered to be conflictive since achieving a low *%Read* rate could be due to (i) those FAQ entries are actually irrelevant to users, or (ii) those FAQ entries are confusingly-written and users are misdirected to them.

Section Balance Analysis is another Summarization Tool that contrasts the number of accesses to a section with section sizes. Low balanced sections (low section_accesses/section_size rate) may contain irrelevant entries to users, thus those sections could be simplified. Overbalanced sections (high section_accesses/section_size rate) indicate that information contained in the section is often useful for users but the section may contain a low number of questions. Thus, in order to improve the FAQ quality those sections could be enhanced with more fine-grained questions. Appropriate preset NL phrases are selected depending on the balance rate achieved and are displayed to FAQ manager describing each case.

The Global FAQ Quality is measured by means of the *%Read* average. As *%Read* measures the usefulness for each question, this new measure represents the usefulness central tendency of the entire FAQ. Values close to 100 indicate that the FAQ is useful for customers. If this rate is low, FAQ managers are encouraged to revise and improve their FAQs.

Finally, Structural Information is considered. This analysis aims to identify potential structural weaknesses by checking certain conditions (Table IV.12). One preset phrase is given to the FAQ manager each time a sign of structural weakness is found.

Condition	**Weakness**
FAQ entry with only one reformulation	There is no syntactic variety
FAQ entry with low number of reformulations AND low *%Read* rate	FAQ retrieval algorithm may not be able to generalize it properly
Section with low number of entries	Section should be reconsidered
Low number of sections AND sections include high number of entries	FAQ is poorly organized. Manual searches may be overly tedious
High number of sections AND sections include low number of entries	FAQ is poorly organized. Manual searches may be overly tedious
Low number of FAQ entries AND high number of proposed queries	Relevant information is not contained in the FAQ

Table IV.12: Structural weaknesses conditions.

3.5.3 Discovering Knowledge Gaps

Techniques described above are intended to identify weaknesses in the current FAQ. The following one is designed to discover which knowledge should be added to the FAQ to satisfy users information requirements.

To illustrate our aim, let us consider a FAQ based on computational models. It would be a valuable contribution to FAQ managers to have an automatic method at their disposal being capable of analysing users queries to provide information such as "In this FAQ there is a lack of knowledge regarding the following concepts: *Kohonen, Adaline*, and *ANN*". In this way, FAQ managers are encouraged to improve their FAQs according to users expectations. It is even possible that these concepts are already contained in the FAQ presenting different wording. For example, *Self Organizing Map* instead of *Kohonen* or *Artificial Neural Networks* instead of *ANN*. In this case, FAQtory discovers how users refer to FAQ concepts. It might be underlined that acting in this way causes the construction of FAQs is no longer directed by expert knowledge but also by users requirements.

To achieve this goal, FAQtory contrasts term-relevance in the FAQ document with term-relevance in all the users queries. Same word-weight criterion defined in TF·IDF ($\mu_d^w = tf_d(w) \cdot idf_d(w)$) is considered to compute term-relevance μ of term w in document d. If some term is considered highly relevant in users queries but lowly relevant in the FAQ it is assumed to be a knowledge gap.

First step consists of identifying the initial candidate list of concepts. Since unigrams (isolated terms) do hardly provide contextual information, it would be desirable detecting noun phrases in each sentence. However, syntactic parsers do not perform efficiently because users may pose incomplete queries. Thus, n-grams containing any noun are considered in this step. We set n to three in FAQtory.

Second step consists of calculating relevance of each candidate concept. In this case, the entire FAQ and the list of user queries are considered as different documents. Term frequency in each document is contrasted with term frequency in the general language [19] and term frequency in the rest of FAQs. We note μ_Q^w and μ_F^w as the relevance of w in users query list and in the FAQ respectively. If w is not contained in some of these lists its corresponding μ^w is considered as the lowest representable number ε satisfying $\varepsilon \simeq 0$ and $\varepsilon > 0$.

Finally, we contrast candidate-relevance scores in users queries with candidate-relevance scores in FAQ by means of $G_w = \mu_Q^w / \mu_F^w$ proportion. Only relevant concepts in query list (25% of the terms achieving the highest μ_w^Q score) are considered. Note that, being $G_w > 1$, the higher G_w score achieved, the more significant is the knowledge gap concerning concept w. If so, a preset sentence including top three knowledge gaps is shown to the FAQ manager.

3.6 Experiences with FAQtory

Current version of FAQtory[20] has been developed by Virtual Solutions[21] following a client-server architecture. The system's sever side operating system is CentOS (version 5.6 Final). On the client side, we use JSP (Java Server Page), AJAX (Asynchronous JavaScript And XML), and Mootools technologies. On the server side, we use Java JDK 6 (1.6.0_20) and preprocessor Lucene[22] to process natural language, and MySQL (14.12) under Apache Tomcat 6 (6.0.32) to manage the data storage.

FAQtory contains currently 15 FAQs. It is being used by the University of Granada (5 FAQs) and private companies (10 FAQs). Although our framework is capable of providing multilingual support, we have only tested our FAQ retrieval module in Spanish and English languages.

Besides the FAQ retrieval module, providing a formal validation of a system of this nature is complicated. Since FAQtory is designed to assist FAQ managers in the improvement of the FAQ, it would be desirable to evaluate the satisfaction of FAQ managers as well as the satisfaction of customers based on their experiences with the system. Nonetheless, due to data privacy policies restrictions we are not allowed to know users identities. Moreover, since those opinions could not be contrasted with other similar platforms, the formal validity of this satisfaction survey would be questionable. For these reasons, we present and discuss our own experiences with the system. Assuming regular maintenance, our aim is to monitor the proportion of failures on the total number of queries in order to examine how it evolves in the curse of time. Every time the system is not able to retrieve any relevant information for a full meaning question in the domain scope should be considered a failure, even if that information was not already contained in the FAQ. However, specifying the criteria to assume one question to be related to a certain given domain is not an easy task. Thus, every time the user does not read (click) any retrieved answer will be here considered to be a failure. According to this, we have created and monitored a FAQ about computational models. As the collection contained just 20 FAQ entries at the beginning and none reformulations, its answering ability was expected to be poor. Students of the subject "Computational Models" (about 60 students) were encouraged to use the system during the course progress (four months). If increasing the size of the FAQ incurs in a lower number of failures cases, then it would indicate that the FAQ manager is properly taking advantage of the system to maintain its FAQ. During the first half of the course progress, a total of (about) 250 questions were posed to the system and 97 queries were explicitly proposed as new candidate questions. In that period, the size of the FAQ

[19] Wikidictionary `http://en.wiktionary.org/wiki/Wiktionary:Frequency_lists` is consulted to perform this goal

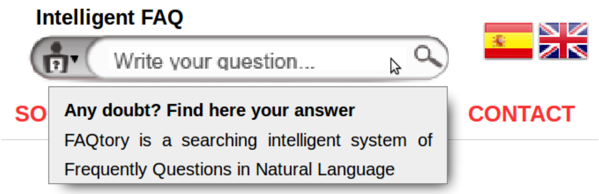
[20] `http://www.solucionesvirtuales.es:8080/FAQtory_VS/inicio.jsp`

[21] `http://www.solucionesvirtuales.es/`

[22] `http://www.dotlucene.net/`

was duplicated and the average failure rate achieved was 35%. According to our expectations, in the second half of the course progress, the number of explicitly proposed questions (just 43) decreased noticeably and so did the average failure rate (about 15%). That is an indication that the FAQ content was evolving in the direction of users requirements. In the end, the FAQ contained 93 FAQ entries and a total of 312 reformulations. Final average failure rate decreased to 9% in the last weeks. Although those experiences could not be considered as a formal validation, they lead us to think that the system's ability to support FAQ managers is actually useful in practice.

FAQtory can be applied to different contexts. Its most direct application is as a complete platform, providing customers and FAQ managers an unified collaborative framework. However, since redirecting customers to an external web may cause their distrust, FAQtory can also be embedded in an already existing web (see Figure iv.13). Thus, customers are able to access information of the company from its own website. In this case, only FAQ managers visit FAQtory to create and maintain their FAQs. Finally, FAQtory can also be applied to e-learning paradigm along the lines of [Yan09a], defining a collaborative community learning by setting all users as FAQ managers. In this way, all users are able to pose questions, propose new questions, and answer proposed questions from other users.



Any doubt? Find here your answer

FAQtory is a searching intelligent system of Frequently Questions in Natural Language

Figure iv.13: Example of FAQtory bar embedded in an already existing website

There are currently three commercial versions of FAQtory under the copyright of Virtual Solutions: the embedded version FAQtory service[23], the full version VS FAQtory[24], and VS FAQtory *Docencia*[25] devoted to teaching purposes.

3.7 Conclusions and Future Work

In order to simplify information access by using NL as query mechanism, numerous FAQ retrieval algorithms have been proposed. However, how to create and maintain high quality FAQs has received less attention. In this section, an application able to retrieve relevant information to customers is proposed. This framework also takes advantage of the usage information to assist FAQ managers in improving their FAQs.

Most recent work in FAQ retrieval rely on meta-knowledge models such as ontologies. However, the task of defining and maintaining those domain models remains restricted to IR experts —we will devote the next chapter to this issue. In this work, algorithms that do not require any knowledge modelling have been considered in order to allow non-technical people to create intelligent FAQs. The algorithm exploits the structure of its knowledge to automatically discover useful relations and knowledge gaps. We have combined high-performance MDE with classical TF·IDF to take advantage of the strengths of both methods.

How users react after receiving answers reflects their interest regarding the retrieved information.

[23] http://www.faqtory.es/faqtory/

[24] http://www.solucionesvirtuales.es/es-ES/soluciones/faqs-inteligentes/vs-faqtory

[25] http://www.solucionesvirtuales.es/es-ES/soluciones/faqs-inteligentes/vs-faqtory-docencia

By analysing those data, it is possible to identify certain FAQ weaknesses or even how users interests evolve over time. Summarization and Visualization Tools have been developed in order to present Usage information to FAQ managers in a meaningful manner. Moreover, the analysis of users queries brings out the most interesting information to users. By contrasting such information with the FAQ, it is possible to automatically detect some knowledge gaps in the FAQ. Those tools allow FAQ managers to interpret the FAQ quality and modify it, if appropriate. Applying Usage Mining techniques the task of creating a FAQ is no longer directed by expert knowledge but also by users requirements.

We had the chance to test our framework in real environments. Our platform is currently being used by different entities as well as by a plenty of users. FAQtory is a flexible platform that could be easily embedded in any website or adapted to act as an e-learning platform. There are currently three commercial versions of our framework maintained by a private company.

Since few work have been conducted in the area of creating and maintaining high-quality FAQs, there is much work ahead. In future work, we intend to incorporate ontologies support in order to give experts in IR the opportunity to get profit of meta-knowledge modelling. That is, we will put our system to test under the SK/PM configuration. In this scenario, how usage-mining techniques could be used to assist experts in identifying new concepts and relations becomes a new challenge.

4 Improving the System: Refining Regular Expressions

4.1 Introduction

Among the possible techniques to address the FAQ retrieval problem, the Template-based approaches [Yan09b, Sne09] reduce the problem to a comparison between the user question with the set of templates that cover the question/answer pairs. However, the main advantage of this model is also its main drawback: templates embody the expert's knowledge on the domain and his/her ability to understand and answer questions, but designing these templates may become a complex task since it is usually carried out manually, incurring the following problems:

1. Considering in advance all possible wordings that users could employ to express the same question is difficult.
2. Creating templates by hand is an error-prone task that entails a considerable amount of work.
3. It is difficult for a human to predict global behaviour of a set of templates.
4. Obtaining an early version of the system with a competitive performance is costly.
5. Since templates must be revised as the domain is (substantially) modified, template-based systems are restricted to stable domains.

ML approaches represent an interesting alternative insofar as no human effort is required once a suitable base of training questions is available. However those methods are usually less interpretable than template-based ones. Thus, how to modify the system to better fit one particular case is not trivial. Additionally, it may even require the learner to be rebuilt once again. Some companies may prefer to ensure full control of the system, even if this entails additional human resources.

There is evidence that template-based methods perform more robustly than other fully-automated approaches [DSS11, Sou01]. However, those results are not surprising insofar as hand-crafted templates embody the human (expert) ability to understand and answer questions. Although these methods are not automatic, companies may prefer to undertake this solution in order to offer a better service. The above mentioned problems make it evident the necessity of an effective mechanism for reducing this human effort in the task of creating consistent templates covering domain questions. [Sne10, p. 391] [DSS11, p. 238] [GLP$^+$10] are just some examples of methods requesting for such methods. So far, we have investigated how to obtain certain classifiers to support the retrieval of the most related FAQs to a given NL query (section 2). MDEs are *plain* regular expressions in the sense that they only consist of a concatenation of words and wildcards. However, although each MDE is easily interpretable, the set of MDEs obtained by this algorithm is usually too broad. Thus, it is difficult for an expert to predict the language accepted by the whole set of templates. Some closely related techniques devoted to bring interpretability and generality to a set of rules could be found in the field of Fuzzy rules [CC07, CC08a]. In this chapter, we investigate how to combine MDEs to create a reduced set of more meaningful and interpretable set of templates.

Main related methods proposed in the literature deal with the creation of templates to extract the answers in QA [RH02, ZL02, SGS06, CKC07], or to determine the most probable answer type in Question Classification [Her01, HW03]. However, neither of them solves directly our problem: answers are pre-stored in the FAQ collections, and the corresponding FAQ entry is not necessarily determined by the type of answer. For example, a FAQ entry gathering the contact information of one particular professor should be retrieved from questions such as "What is the email of Professor

$< X >$?", "Where is the office of Professor $< X >$?", or "How could I schedule a personal tutor session with $< X >$?". Note that the answer type is different in all cases, albeit the content of the FAQ entry is undeniably related to all of them. In this chapter, we propose a semiautomatic method to reduce the problem of creating templates to that of validate, and possibly modify, a list of proposed templates for the FAQ retrieval task. In this way, a better trade-off between reliability —the system is still monitored by an expert— and cost, is achieved. In addition, the effort and time required to obtain an early operative version of the system is substantially alleviated, updating or repairing templates after domain changes becomes easier, and human mistakes are reduced. Figure iv.14 illustrates our proposal.

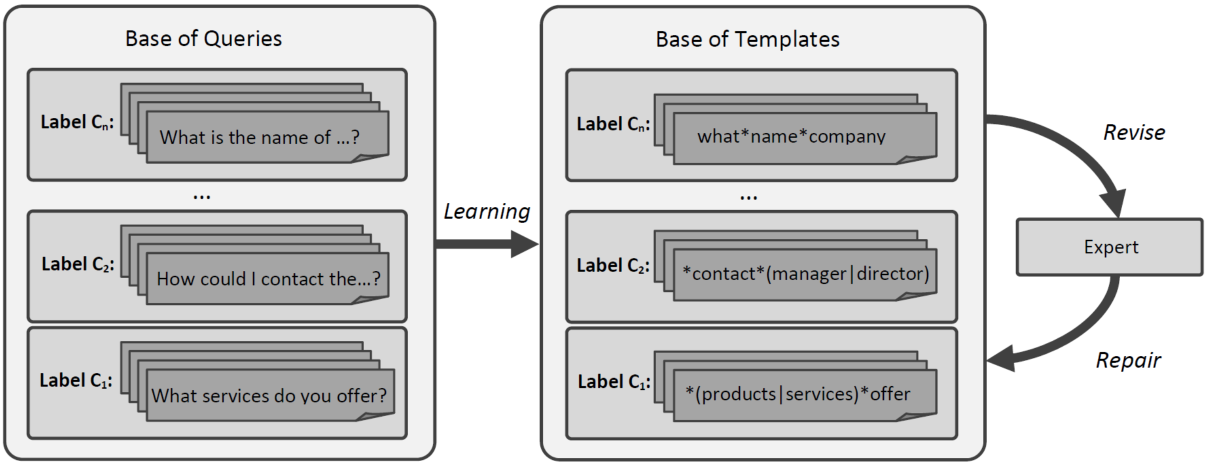


Figure iv.14: Semiautomatic Learning regular expressions to Template-based QA systems

Regular expression matching is a well-known tool broadly used in many NL applications [KCGS96, NG01]. Although there are several works in learning regular expressions given a set of positive and negative data chains (e.g. [Fer09]), most applications are devoted to obtain Document Type Definitions (DTDs) descriptions [BNST06, BGNV10] or learning templates to extract the answer in QA [RH02]. Our problem differs from these insofar as our data source consists of NL sentences. Hence, identifying loops is not helpful in our case. Instead, creating regular expressions reflecting which words are relevant, discarding irrelevant ones, and capturing different wordings would be a worthier challenge. To this purpose, the MDE algorithm seems to be a promising option.

Collecting reformulations is a low-cost task that helps to expand the lexical and syntactical variety of questions. Our proposal is based on inferring regular expressions inducing the language conveyed by a set of previously collected query reformulations. Our intention is to frame the problem as an optimisation one. Table IV.13 shows an example of the reformulations associated with the domain question "*How can I contact the manager of the company?*", and the set of templates (regular expressions) obtained by our approach. The wildcard '*' denotes any string, and '|' represent disjunction. Note that because these regexes are more elaborated than plain MDEs, the number of templates is expected to be smaller than that obtained by the MDE algorithm. Thus, the system may become more interpretable for a human that could even revise and modify them, if appropriate.

The rest of this section is organized as follows: An overview of previous related work is presented in section 4.2. Section 4.3 exposes the criteria that determine the adequacy of a set of templates with respect to the expert decision. Section 4.4 offers an explanation on how two well-known optimisation algorithms could be applied. Finally, Section 4.5 describes our experiences with the system, and Section 4.6 concludes with a discussion of the framework and future research.

Reformulations	How can I contact the manager of the company? Please tell me how I could contact the manager What is the telephone number of the manager? What is the FAX number of the director? Tell me the email of the director of this company
Regular Expressions	* how * contact * (manager|director) * * (((telephone|FAX) number)|email) * (director|manager) *

Table IV.13: Example of regular expressions obtained

4.2 Related Work

In this section, the main related work to our approach are discussed. This section covers the following areas: template-based QA approaches and applications, Question Classification, and methods for learning regular expressions.

Automated QA is a sub-area of NL understanding introduced in the late 1960s that aims to provide concise answers to questions in NL (the interested reader is referred to [AS05] for an overview on QA systems, and [MV07] for an overview of main approaches focusing on Restricted Domains). Recent research indicates QA is an useful tool for different fields including e-Commerce [TR08] and e-Learning [SVFBCN$^+$12]. The most related work within QA to our research are template-based approaches. The aim of those systems is to cover the knowledge with a set of linguistic templates that are later used for matching. START [Kat97] with Omnibase [KFY$^+$02] is an example of this approach which employs a lexical-level —handling synonymy and IS-A relations— and a syntactical-level comparator. [Sou01] presented the template-based QA that won the QA track of TREC 2001 and 2002. It consisted of obtaining a set of indicative patterns to extract the most probable answers to a certain sort of questions. [Yan09b] combines a template-based approach with a domain ontological model based on keywords. This system needs a knowledge domain model in which domain concepts are previously represented. Sneiders' systems [Sne09] perform the matching based on a set of regular expressions that encapsulate human decisions, called "Phrases", and a keyword comparison technique, called "Prioritized Keyword Matching". Keywords are classified as *required, optional*, and *irrelevant*, attending to their role in the domain. Examples of commercial template-based FAQ answering systems could be consulted in[26]. More examples of QA systems relying on hand-crafted templates could be seen in [BLB$^+$01, ZL02, GLP$^+$10]. Apart from template-based approaches in Question Answering and FAQ retrieval, this technique has also been applied to NL Interfaces to Data Bases [Sne02] and Ontology interfaces [OOMH08] where the question templates may contain entity slots that are replaced by data instances from the underlying knowledge model.

Automatic E-mail Answering is a relatively new research area that bears strong resemblance to FAQ retrieval. It aims to automatically classify and answer recurrent questions posed in mails [BSA00, LK03, Sne10, MZ09]. In [DSS11], it was found that, in this scenario, template-based methods outperforms machine-learning ones in terms of precision. However, it was also pointed out that creating those templates is much more time consuming than training ML methods. In this regard, methods for obtaining templates would be desirable.

Question Classification (QC) concerns with determining automatically the most probable category for a given question with respect to its answer type [Her01]. This field attracted increasing

[26] AskJeeves: `http://www.ask.com/`, Kiwilogic: `http://www.kiwilogic.com/`

attention from the ML community as a promising alternative to create hand-crafted rules. In this regard, some approaches [HW03] make use of supervised models such as Support Vector Machines (SVM) where words (optionally n-grams) conform plain features that could be combined with syntactic [ZL03] and semantic [BM07, LR06] information. Although FAQ retrieval seems to be intrinsically related to QC, there are some important differences that should be pointed out: different categories in QC usually correspond to different sorts of abstract concepts, while in FAQ retrieval, different categories correspond to different recurrent questions, regardless of the nature of the underlying concepts. For example, "What is the Kernel of Linux?" and "Give me a definition of Bash" could belong to the same category from the point of view of QC —both questions demand a definition—, while they may actually correspond to different FAQ entries. Equivalently, "What can you tell me about Linus Torvalds?" and "History of Linux" could be associated with the same FAQ entry, while the sort of query may be completely different for a QC approach. In any case, QC through ML seems not to be an appropriate alternative to the problem under consideration, since the trained models are less interpretable than templates, and their performance is thus hardly revisable for a human.

How to bring interpretability to a set of rules has thus far been investigated in the field of Fuzzy logic. Fuzzy Model Identification consists of identifying the set of fuzzy rules that better describe a system behaviour, based on the observation of a given set of previously labelled examples [CCZ04a, CCSZ01]. In this regard, some methods rely on optimization algorithms, such as the Ant Colony Optimization, to optimize both accuracy and interpretability of a set of fuzzy rules [CC07, CC08a]. As will be seen, our proposal bears strong resemblance to these ones, in the sense that we will also take advantage of optimisation algorithms (Simulated Annealing and Genetic Programming in our case) to improve certain desired properties, including interpretability, of a previously generated set of rules —the MDEs. A closely related issue has also been addressed in the field of fuzzy models in [CCZ04b, CCSZ99], where authors propose a method to improve both generality and interpretability of the extracted fuzzy rules, by maintaining high accuracy. Similarly, we propose a new method for creating linguistic patterns from the optimization of generality, interpretability, and correctness degrees taking a set of labelled NL questions as starting point.

There exist different paradigms for learning formal languages. The Gold's model [Gol67] represents arguably the most extended paradigm. This model aims to infer in the limit a regular language given a set of finite samples of the language (examples could be found in [Fer09, BNST06]). Angluin's model [Ang87] aims to learn languages from queries samples to a *Minimally Adequate Teacher* that is able to answer *membership queries* and evaluate *conjectures* on a regular set —"yes" if this set is equal to the unknown language, and a *counterexample* otherwise (see [Kin08] for an extension adding the concept of *corrections*). Finally, in Valiant's Probably Approximately Correct (PAC) model [Val84], the learner should select the generalization function that minimizes the generalization error in learning a distribution of samples. [SA95] include application examples in this line. However, since most of these algorithms operate with data chains (consider the example *ababb, aabb, ababa, abc* taken from [Fer09]), their primary concern is to define alignments and to identify loops and Kleene stars operators (following the example, $a^+b^+(\varepsilon \mid ab^+(\varepsilon \mid a) \mid c)$ would be obtained). Unfortunately, progresses in this line are hardly applicable to NL. To our aim, identifying relevant words and capturing irrelevant ones become a primary concern, while identifying loops in questions becomes unhelpful. Also, although our motivation is related to the so-called *minimum consistent DFA* problem [Gol78], there are some aspects in our proposal that differ from it. Minimum consistent DFA is mainly motivated by minimizing the complexity (number of states) of the DFA (Deterministic Finite Automata), whereas other criteria such as *generality* (see Figure iv.1 in chapter 1.4; this concept will be formally defined below) are equally important to our proposal. Thus, it should be remarked that those problems are not comparable.

Most of the early studies in this line do not operate directly with regular expressions but with other formal languages, such as automata [Yok95], or formal grammars [Sak88]. In any case, since there exist algorithms to transform them into a regular expression [HMU00], those formalisms could be considered similar to our purpose. Also, most of the algorithms that directly obtain regular expressions have been applied for inference of DTDs and XMLs Schemas Definitions (see [GGR$^+$00, Fer09, BNST06, BGNV10]). In the field of Information Extraction (IE), learning regular expressions was first applied to obtain patterns describing structural tagged tokens generated by other text-processing tools such as PoST [Sod99, RF08] or syntactic analysers [WZT$^+$11]. In this line, [SNSP01] proposed a learning algorithm based on the maximum entropy to disambiguate the sense of words. Other approaches dealing with IE automatically learn the surface text patterns for extracting the answer [RH02, ZL02, SGS06, CKC07], or learn regular expressions describing structural features of entities in the text [LKR$^+$08]. In [SDYH08], a different sort of templates, called surface text patterns, are learned based on a sentence alignment algorithm for Chinese QC. More details on these sort of techniques will be driven in chapter V.

Unfortunately, even if the nature of these works seem to be strongly related to ours, the truth is that creating templates to recognize questions differs noticeably from creating templates to extract answers. In this sense, there are approaches more closely related to our aim. [Li02, CKC07] are examples of pattern-based approaches that aim to overcome the limitations of hard patterns from a probabilistic point of view. In [VZPM05] the Alignment-Base Learning algorithm was applied to the problem of QC. This algorithm relies on the idea that constituents can be interchanged to deploy new valid sentences. In the training phase, the algorithm searches for sentences alignments (hypotheses) that are later used to conform regular expressions associated with the Expected Answer Type (EAT). The effectiveness of this method is based on its ability to capture the syntax variety for a given EAT. However, as stated before, it is not necessarily true that this will always be useful in FAQ retrieval. The most related FAQ entry to a question may not be decided from its EAT. In the example offered in section 4.1 (concerning the contact information of certain professor), each question has a different EAT. Rather, combinations of words “email”, “office”, “professor”, “tutor session”, or the name of the professor, seem to be more informative in this case. In [BZHZ10], authors propose strict patterns (regular expressions) where irrelevant words are substituted by wildcards, and soft patterns based on clustering. The main difference with our method concerns with how authors treat “irrelevant” words. They rely on the class of each word (*function* and *content* words), that is language-dependant. Instead, we rely on differentiability criteria —an implicit feature of the underlying dataset that is self-tuned.

Finally, the problem of inferring templates has also been faced from an optimisation point of view. In this regard, Genetic Programming (GP) algorithms play a key role. GP algorithms rely on the evolution of formal structures (or programs) representing the individuals of a population. GP algorithms have been widely used in many areas such as grammar evolution in the medical field [NLWC98], POSIX regular expressions inference [Cet07, LH08, BDDL$^+$12], stochastic regular expressions for pattern matching [Ros00], or sentence generation [LKHC04]. As shall be seen, we will also employ optimisations algorithms in this chapter. However, our algorithms differ from these in several aspects: since we use questions as training sources, we have investigated linguistically-motivated operators as well as how to deal with wildcards.

4.3 Proposal for Learning Templates

Given a base of questions, the theoretical study conducted in Section 1.4 concludes the existence of learner machines to obtain a *correct*, *generalizable*, and *non-redundant* coverage. The MDE algorithm presented in section 2 is an example of such learner machines. Since there may be several coverages presenting those properties, we aim to find the best possible. To this end, we will define some criteria that lead us estimate the quality of a coverage. Those criteria are based on the notion of *usefulness* and *maintenance*. The algorithm will search the best coverage possible in terms of these criteria by applying successively a set of operations. These operations transform successively an initial set of automatically generated templates: the *Minimal Differentiator Expressions*.

This section is structured as follows: Measures for interpretability and suitability of the templates according to experts' criteria are discussed in Section 4.3.1, and algorithms optimising these measures are later explained in Section 4.4.

4.3.1 Measuring Usefulness and Maintenance

In order to alleviate as much as possible the necessity of repairing templates, the algorithm optimizes some criteria measuring the *quality* of the template set. Since those templates should be revised by an expert, the extent of the concept *quality* becomes undeniably subjective. To this aim, we will define certain measures that will help us to establish a criterion to evaluate a set of templates. Since quality is undeniably subjective, there is not any standard criterion to determine which template is better. Even so, we consider that quality depends somehow on the degree of *correctness*, *generalization*, and *interpretability* of each regex. We define the following measures to approximate heuristically those criteria.

Correctness measures the extent of effectiveness of a given regex with respect to positive (I^+) and negative (I^-) sample sets, that is, its degree of *adequacy* on that domain. Correctness is calculated as the proportion of accepted positive examples provided that none of the negatives examples is accepted[27]:

$$correctness(r, I^+, I^-) = \begin{cases} 0 & \text{iff} \quad L(r) \cap I^- \neq \emptyset \\ \frac{|L(r) \cap I^+|}{|I^+|} & \text{in other case} \end{cases} \tag{IV.31}$$

A regex is said to be correct if its correctness degree with respect to I^+ and I^- is positive. For example, let $I^+ =${*What is the email of the director?, Could you kindly tell me the email of the director?*}, then the correctness factor of regex $r =$**what*email** is 0.5 assuming that none of the negative examples in I^- is in the language accepted by r. Finally, correctness degree of a set of templates R with respect to I^+, I^-, is calculated as the average ($ave(\cdot)$ in Equation IV.32):

$$Correctness_{I^+,I^-}(R) = ave_{r \in R}(correctness(r, I^+, I^-)) \tag{IV.32}$$

Generalization measures the capability of a regex to accept examples never seen before. This measure is related to the concept of recall. Generalization should be in accordance with the proportion of potential examples accepted. Unfortunately, since every regex containing wildcards accepts

[27] This concept is somehow related to the standard Recall. However, notice that Correctness is estimated on the training data. Furthermore, the precondition whereby none of the negative examples could be accepted mixes this concept with Precision.

infinite examples, it is not always possible to compute this measure. Nevertheless, as we are dealing with NL, it is worth considering only finite sentences (sentences containing at most n words). The problem is then translated to that of calculating the proportion (IV.33), where $L_{\Sigma^n}(r)$ represents the finite language accepted by the regular expression r in Σ^n (the set of all words with at most n symbols).

$$\frac{|L_{\Sigma^n}(r)|}{|\Sigma^n|} \tag{IV.33}$$

For example, being $\Sigma = \{a, b, c\}$, and $n = 2$, the generalization factor of regex $r = (a|b)*$ will depend somehow on the proportion of accepted words ($\{aa, ab, ac, ba, bb, bc, a, b\}$) with respect to all words in Σ^2 ($\{aa, ab, ac, ba, bb, bc, ca,\ cb, cc, a, b, c, \epsilon\}$). That is, r generalizes 8/13 of the entire language.

A brute force count is extremely inefficient, so we propose the following method to calculate $|L_{\Sigma^n}(r)|$. Firstly, the regex r is translated to its equivalent Deterministic Finite Automata (DFA). Later, the directed graph associated with that automata is obtained by considering each state as a node, and each transition as a directed arc. Labels in the graph represent the number of transitions between the corresponding states in the automata. Finally, according to graph theory, the number of accepted examples containing exactly i symbols is calculated as $\sum_{f\in F}[m]^i_{s,f}$, where m is the adjacency matrix of the graph, s is the index of the node corresponding to the starting state, F are the indexes of the nodes corresponding to final states, and $[m]^i$ is the i^{th} power of the matrix m. The number of all examples in Σ^n could be calculated as $\sum_{i=0}^{n}|\Sigma|^i = \frac{|\Sigma|^{n+1}-1}{|\Sigma|-1}$. Finally, we add 1 and we apply a logarithmic transformation to normalize and enhance the curve respectively[28] (equation IV.34). A detailed example could be found in Appendix 2.

$$generalization(r, n) = \lg_2\left(\frac{(|\Sigma|-1)\cdot\sum_{i=1}^{n}\left(\sum_{f\in F}[m]^i_{s,f}\right)}{|\Sigma|^{n+1}-1} + 1\right) \tag{IV.34}$$

Note that, *a priori*, there is not a direct correlation between the *generalization* degree and the *quality* of a regex. Based on his/her intuition, an expert could prefer more generalizable regexes, or more restrictive ones. In any case, it should be the expert criterion the only factor that determines how adequate the generalization degree of a regex is. For this reason, the expert is requested (only once) to specify what, in his/her opinion, is the ideal regex $\widehat{r}$ in terms of generalization[29]. This regex will set the maximum value for the measure. Finally, the generalization adequacy with respect to the ideal regex $\widehat{r}$ is calculated as.

$$Generalization_{\widehat{r}}(R, n) = 1 - |generalization(\widehat{r}, n) - ave_{r\in R}(generalization(r, n))| \tag{IV.35}$$

Interpretability of a template is arguably related to the complexity of its structure. Regardless of the size of the language accepted by a regex, an human being (expert) should read and process each symbol and meta-symbol of the regex. We believe that the expert interprets the regex insofar as he/she is able to mentally construct an equivalent representation. In this sense,

[28] The reader should be aware of the possible confusion that symbols used to denote *sigma* and *summation* may cause.

[29] In the limit, an expert desiring regexes as generalizable as possible, could insert the regex $\widehat{r} = *$, while an expert desiring regexes as restrictive as possible could insert the regex $\widehat{r} = \varepsilon$

we think that wildcards play a different role, being simply considered as an additional symbol. To approximate the *structural simplicity* of a regex, we first calculate the equivalent Non-Deterministic Finite Automata (ε-NFA) using the Thompson's algorithm, because it is possibly the most intuitive equivalent structure. Later the number of states (E) and transitions (T) determine the structural complexity.

$$structuralsimplicity(r) = \frac{1}{|E| + |T|} \tag{IV.36}$$

The Structural simplicity of a set of templates R is defined as the average:

$$Structuralsimplicity(R) = ave_{r \in R}(structuralsimplicity(r)) \tag{IV.37}$$

To access a detailed example, see Appendix 3.

Interpretability of a set of templates is related to the number of templates. Even if each regex is simple in terms of structural complexity, the whole system may become poorly interpretable if it is composed by a plenty of templates. In this regard, we estimate the interpretability of a set of templates R as

$$Sizesimplicity(R) = \frac{1}{|R|} \tag{IV.38}$$

4.3.2 Adjusting the Fitness Function to the Expert's Criterion

As commented before, the quality of a set of templates is subjective. Similarly to other learning paradigms, the expert could be considered as an oracle able to predict the behaviour of templates against other potential questions not previously considered. For this reason, even if this quality is arguably related to the measures exposed before, it is the expert the only one that could determine the quality of a set of templates according to his/her preferences and his/her intuition on $\overline{I^+}$ and $\overline{I^-}$. As an example, an expert could prefer to make no risk, obtaining a broader set of templates with low degree of generalization. In contrast, other expert could prefer to obtain a smaller set of templates of arbitrary structural complexity, being also as generalizable as possible.

As commented before, the interpretability should be related to the structural complexity of each template (Equation IV.37) and the total number of templates (Equation IV.38). Thus, the degree of interpretability of any set of templates R is modelled as:

$$Interpretability(R) = \frac{W_1 \cdot Structuralsimplicity(R) + W_2 \cdot Sizesimplicity(R)}{W_1 + W_2} \tag{IV.39}$$

where W_1 and W_2 are the weighting criteria specified by the expert. Similarly, the Fitness function of any set of templates R is calculated as a combination of *Correctness* (Equation IV.32), *Generalization* (Equation IV.35), and *Interpretability* (Equation IV.39) degrees.

$$Fitness(R) = \frac{W_3 \cdot Correctness_{I^+,I^-}(R) + W_4 \cdot Generalization_{\widehat{r}}(R,n) + W_5 \cdot Interpretability(R)}{W_3 + W_4 + W_5} \tag{IV.40}$$

where W_3, W_4, and W_5 reflects the importance assigned by the expert to each criterion. Note that assigning a zero value to any criterion W_i represents that this criterion is irrelevant to the expert.

4.3.3 Theory Meets Practice

Section 1.4 covered a theoretical motivation, whereas section 4.3 described the concrete metric aimed to reflect the quality of a given set of templates. This section is to highlight the most important equivalences between the theoretical and practical sides (see Table IV.14). Note that Interpretability has no equivalence in the theoretical side.

Ref	**Theoretical**	Ref	**Practical**
Eq.IV.7	Correctness	Eq.IV.32	Correctness Degree
Eq.IV.8	Generalization	Eq.IV.35	Generalization Degree
Eq.IV.9	Non-redundancy	Eq.IV.17	Minimality in MDE algorithm
Sec.1.4	$E\,discriminates(A,B)$	Eq.IV.15	Differentiability in MDE algorithm
Sec.1.4.1	Base of Questions ($\mathcal{B}$),	-	FAQ
Sec.1.3	Alphabet (Σ)	-	English words plus Domain terms
Def.1	Learner Machine	Sec.2.3.2, 4.4	e.g. MDE algorithm, SA, or GP$_{MDE}$ (below)
Eq.IV.2	The Recognizer (Π)	-	FAQ retrieval engine, i.e. regex matcher

Table IV.14: Most relevant equivalences between theoretical properties and definitions w.r.t. practical methods

4.4 Optimisation Algorithms

Once the measures to evaluate a set of templates have been defined, a searching strategy should be adopted. Since an exhaustive exploration for the best solution in the search space would be extremely inefficient, we assume that obtaining an acceptable solution in a fixed amount of time is preferable. Thus, the problem is here set up as an optimisation problem. As a result, various optimisation algorithms are suitable for being applied. On this study, we illustrate the adaptation of two well-known methods, Simulated Annealing (Section 4.4.1) and Genetic Programming (Section 4.4.2), bearing in mind that other optimisation strategies could also be suitable for the purpose. This section concludes with a discussion on the efficiency of this algorithm in terms of time complexity and space complexity (Section 4.4.3).

4.4.1 Adapting Simulated Annealing

In this section, we face the problem through a Simulated Annealing (SA) strategy, previously defined in [KGV83, Č85]. This method is motivated by the statistical background describing how the internal energy in annealing of solids evolves in the curse of time. According to a decreasing probability, worse solutions could be accepted as a mechanism to avoid the system to be stuck in local optima.

We will consider a set of templates as a candidate solution in terms of SA. Its Fitness value will determine its energy. As commented before, the initial solution is obtained though the MDE algorithm. Successive solutions (neighbours) are obtained by applying certain operations to previous solutions.

Energy The equation that simulates the internal energy $E(R)$ of a candidate set of templates R is defined as:

$$E(R) = 1 - Fitness(R) \tag{IV.41}$$

Probability of Acceptance To determine whether a new solution is accepted we use the a probability function similar[30] to that defined by Kirkpatrick.

$$P(e, e', T) = \begin{cases} 1 & if\ (e' < e) \\ exp(\sqrt[5]{e - e'}/T) & otherwhise \end{cases} \tag{IV.42}$$

where e is the energy of the current solution, e' is the energy of the new solution, and T is the simulated temperature.

Neighbours Given a set of templates R, a neighbour solution is obtained by adding or removing an element. While adding a new regex, concatenation-join or disjunctive-join operation is randomly decided and applied only to feasible operators. Those heuristics are linguistically motivated.

(i) Concatenation-Join This operation obtains a more restrictive regex based on two previous ones, that is, a regex accepting a narrower language. It is based on the idea that the more words are understood in a sentence, the likelier the meaning of the sentence is correctly caught. To this end, both input regexes are decomposed. Their components are concatenated following the order of precedence of a given sentence.

We use $w_{a..b}$ to denote the substring from index a to index b for a given string. Given a question $s = w_1 w_2 \cdots w_n$, where $w_i \in \Sigma$, and two regular expressions $r_1, r_2 \in R_\Sigma$ satisfying $s \in L(r_1)$ and $s \in L(r_2)$ such that:

$$\begin{aligned} r_1 &= \varphi_1\, \alpha\, \gamma_1,\ w_{a..b} \in L(\alpha) \\ r_2 &= \varphi_2\, \beta\, \gamma_2,\ w_{c..d} \in L(\beta) \end{aligned}$$

Where $\varphi_1, \varphi_2 \in (* \cup \varepsilon)$, following scenarios[31] are possible for $Concat(s, r_1, r_2)$.

$$Concat(s, r_1, r_2) = \begin{cases} \varphi_3\, \alpha\, Concat(w_{b+1..n},\, \gamma_1,\, \beta\, \gamma_2) & \text{iff } \alpha \text{ `precedes' } \beta \quad (b < c) \\ \varphi_3\, \alpha\, Concat(w_{b+1..n},\, \gamma_1,\, \gamma_2) & \text{iff } \beta \text{ is `included' in } \alpha \quad (a < c \wedge b > d) \\ \varphi_3\, (\alpha \,|\, \beta)\, Concat(w_{b+1..n}, \gamma_1, \gamma_2) & \text{iff overlapping} \quad (a = c \wedge b = d) \end{cases} \tag{IV.43}$$

Where $\varphi_3 = \begin{cases} * \; if\ a > 1 \\ \varepsilon \; if\ a = 1 \end{cases}$

[30] We have empirically observed that energy variations between subsequent solutions are the order of 10^{-5}. For this reason, we enhance this variations using the $5th$ root.

[31] We assume $a \leq c$, otherwise $Concat(s, r_2, r_1)$ is returned.

The recursive process ends when any of the input regex is empty or '*', returning the other regex as output. It also ends when none of the above cases are satisfied. In that case, a *null* regex is reported. Let us consider an example where

$$s = \textit{How could I contact the Marketing Department?}$$
$$r_1 = how * Marketing*$$
$$r_2 = *contact * Marketing*.$$

The following trace occurs:

$$Concat(s, r_1, r_2) =$$
$$how\,\text{Concat(could I contact the Marketing Department, } *Marketing*, contact * Marketing*) =$$
$$how * contact\,\text{Concat(the Marketing Department, } *Marketing*, *Marketing*) =$$
$$how * contact * Marketing\,\text{Concat(Department, } *, *) =$$
$$how * contact * Marketing*$$

Note that s is also in the language accepted by the resulting regex.

(ii) Disjunction-join: The aim of this operation is to obtain a new regex able to cover the sentences covered by two previous regexes. This transformation is motivated by the Harris' substitutability criterion [Har51]. Given two regex such that:

$$r_1 = \gamma_1\,\alpha\,\varphi_1$$
$$r_2 = \gamma_2\,\beta\,\varphi_2$$

Where $\gamma_1, \gamma_2, \varphi_1, \varphi_2$ are different from $'*'$ or $'\varepsilon'$. Regexes α and β are said to be interchangeable if their context (left and right regexes) is similar (for example $\gamma_1 = \gamma_2$ and $\varphi_1 = \varphi_2$). We will also consider as similar the regexes accepting subsumed languages, for example $\gamma_1 =$*(what|tell me)* and $\gamma_2 =$*what*. If the context is similar, *Disjunction-join* calculates the following regular expression:

$$Disj(r_1, r_2) := max(\gamma_1, \gamma_2)\,(\alpha\,|\,\beta)\,max(\varphi_1, \varphi_2) \tag{IV.44}$$

Where *max* is a procedure that returns the regex accepting a broader language, that is:

$$max(u, v) := \begin{cases} u\ if\ L(v) \subseteq L(u) \\ v\ if\ L(u) \subset L(v) \end{cases} \tag{IV.45}$$

Let us consider the following example. Given two regular expressions

$$r_1 = (what|tell\,me) * telephone\,number * director$$
$$r_2 = what * email * director$$

this transformation obtains the new regex

$$Disj(r_1, r_2) = (what|tell\,me) * (telephone\,number|email) * director.$$

Removing: While removing an element, the probability for a template r of being chosen is proportional to its energy $E(\{r\})$ —worse templates in terms of Fitness are more likely to be deleted.

Neighbour algorithm: It is only worthy applying concatenation-join to regexes accepting the same positive example in I^+ (see Equation IV.46). In the case of disjunctive-join, only regexes accepting different sentences in I^+ should be considered (see Equation IV.47). The *neighbour* procedure is shown in Algorithm iv.15.

$$candidatesConcat(R, I^+) := \{(s, r_i, r_j) \mid s \in I^+,\ r_i, r_j \in R,\ s \in L(r_i) \wedge s \in L(r_j)\} \tag{IV.46}$$

$$candidatesDisj(R, I^+) := \{(r_i, r_j) \mid r_i, r_j \in R, \exists s_1, s_2 \in I^+ : s_1 \in L(r_i) \wedge s_2 \in L(r_j) \wedge s_1 \notin L(r_j) \wedge s_2 \notin L(r_i)\} \tag{IV.47}$$

```
Procedure Neighbour($R, I^+$)

$rand \leftarrow rand()$ {Random decision}
if $rand < p_1$ then
    {Adding a new regex through concatenation-join}
    $(S, O_1, O_2) \leftarrow randSelection(\,candidatesConcat(R, I^+)\,)$
    $R \leftarrow R \cup \{Concat(S, O_1, O_2)\}$
else if $rand < p_2$ then
    {Adding a new regex through disjunctive-join}
    $(O_1, O_2) \leftarrow randSelection(\,candidatesDisj(R, I^+)\,)$
    $R \leftarrow R \cup \{Disj(O_1, O_2)\}$
else
    {Removing a randomly selected regex}
    $R \leftarrow randRemove(\,R\,)$
end if
return $R$
```

Figure iv.15: Obtaining neighbour solutions

Simulated Annealing Algorithm The initial solution is calculated by means of MDE algorithms. Thus, the initial energy is established as the energy of the MDEs. In order to avoid the algorithm to be stuck in bad solutions, we have added a reboot mechanism. This mechanism restores the best solution (*Rbest*) as the actual solution (*Rnew*) if no improvements were achieved in previous γ steps. Finally, the SA algorithm is adapted as follows (Algorithm iv.16).

4.4.2 Adapting Genetic Programming

Genetic Programming are biologically-inspired optimisation algorithms based on the evolution of tree-structures (computer programs). Those trees represent regular expressions in this study. The evolution is carried out by applying the evolutionary operators crossover and mutation.

```
$R \leftarrow MDE(I^+, I^-), e \leftarrow E(R)$ {Initialization, calculate MDEs and its energy}
$Rbest \leftarrow R, ebest \leftarrow e$ {Initialize best solution and its energy}
$k \leftarrow 0$ {Initialize the iteration counter}
while $k < kmax$ do
    $T \leftarrow temperature(k/kmax)$
    $Rnew \leftarrow neighbour(R, I^+)$ {Get a random new neighbour}
    $e' \leftarrow E(Rnew)$
    if $P(e, e', T) > rand()$ then
        $R \leftarrow Rnew, e \leftarrow e'$ {Acceptance of the solution according to a decreasing probability}
    else if no updates $> \gamma$ then
        reboot() {Reset $Rnew$ if the algorithm is stuck}
    end if
    if $e < ebest$ then
        $Rbest \leftarrow R, ebest \leftarrow e$ {Update the best solution if current one is better}
    end if
    $k \leftarrow k + 1$ {Next iteration}
end while
return $Rbest$
```

Figure iv.16: Simulated Annealing algorithm

Tree Representation Individuals are represented through tree-structures. Internal nodes represent disjunctions or concatenations, while leaf nodes represent symbols of $\Sigma \cup \{'*'\}$. The corresponding regex is obtained by performing a preorder traversal of the tree (examples could be found below).

Initialization of Population We apply the so-called *grow* method to generate the initial population whereby the initial population is composed by randomly generated trees, so that operations (internal nodes) and symbols (leaf nodes) are randomly selected. However since we believe that MDEs present useful information, we consider MDEs belonging to the initial population set.

Crossover This method is applied on two individuals by switching two randomly selected branches. Note that regexes created by this operator may be significantly different from their parents. Figure iv.17 shows an example of this operator applied to two given regexes: Individual 1 = *what * fax * (director | manager)*, and Individual 2 = ** email * company*

Mutation The mutation operator is applied to an individual by modifying a randomly selected node. If this node is internal, the mutation consists of switching between disjunction and concatenation operators. If it is a leaf node, the symbol is replaced by a random one (recall that '*' is considered to be a symbol).

For example, considering the regex *"what * (fax | number) of this company"*, following individuals could appear after mutating an internal node, or a leaf node, respectively:

*what * (**fax number**) of this company*

*what * (fax | number)* ***** *this company*

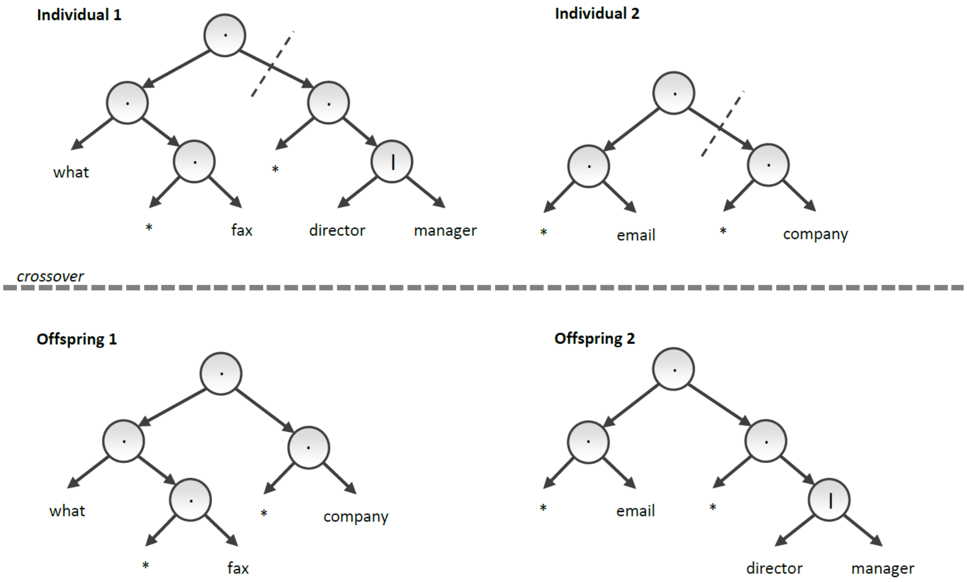


Figure iv.17: Crossover: Offspring 1 = *what * fax * company*, Offspring 2 = ** email * (director | company)*

Fitness Function We adopt the *standarized fitness* form (see Equation IV.48)—lower values represent better solutions, and 0 the best one— to calculate the fitness for individual i in generation t based on Equation IV.40.

$$f_s(i,t) := 1 - Fitness(\{pop(i)\}) \tag{IV.48}$$

Given that GP evolves an entire population, we apply the fitness function to a reduced coverage of individuals accepting I^+. This coverage is obtained by means of a greedy algorithm that subsequently selects the best individual until all examples are accepted. For each uncovered example, if any, a template accepting it and only it is added to the coverage. Thus, the solution returned is not the best individual, but the greedy coverage of the population.

4.4.3 Efficiency of the Algorithms

In this section, we discuss the theoretical efficiency of the proposed algorithm in terms of time complexity. Section 4.5.2 offers an empirical contrast.

As an optimisation problem, the algorithm is run until *maxk* budget is exhausted. Although this parameter should be set according to the domain complexity, it is a constant in terms of time complexity. Thus, the efficiency of the algorithm is determined by the time complexity of its initialization, operations, and fitness calculus.

Being n the number of examples in I, the time complexity of the initialization algorithm (MDE) is asymptotically bounded by $O(n^2)$ —differentiability should be assured by comparing each candidate expression to the rest of examples in I^-, see [MNCZ12, MRCZ12a] to access further explanations. The number of MDEs obtained —and therefore the initial size of R— is bounded by $O(n)$.

The cost of generating neighbours in SA is subordinated to the cost of selecting the candidate parameters for operations (Equations IV.46 and IV.47). Note that there is no need for exploring the entire set of candidates, but only selecting random suitable parameters. This could be achieved in $O(n^2)$ for both operations. Regarding GP algorithm, evolutionary operators (selection, crossover, mutation, replacement, and reboot) are considered to be $O(1)$.

Regarding SA algorithm, there is no need for recalculating the Fitness function for R in each iteration. Instead, each partial measure could be readjusted just by considering the regex added or removed. Since the cost of *correctness* measure (Equation IV.31) and *generalization* measure[32] (Equation IV.33) is $O(n)$, and the cost of *structural simplicity*[33] (Equation IV.36) is $O(1)$, Fitness function is firstly calculated in $O(n^2)$ and updated after each iteration in $O(n)$. This is not the case for GP algorithms. Greedy coverage is calculated in, at worst, $O(n)$ steps and evaluation of each individual is performed in $O(n)$

In this way, the efficiency of the Simulated Annealing algorithm is asymptotically bounded by $O(n^2 + n^2 + maxk \cdot (n^2 + n)) = O(n^2)$, and the efficiency of Genetic Programming algorithms is $O(n^2 + n^2 + maxk \cdot (1 + n) + n) = O(n^2)$.

4.5 Computational Results

This section offers several experiments and results obtained while evaluating our proposal. As commented before, our aim is not only to infer templates that cover a set of previously collected sentences, but also to allow the expert maintenance to become easier. Thus, since our goal is two-fold, we have conducted two set of experiments. First of all, we have formally evaluated our system as a question classification method by contrasting the performance with other methods (Section 4.5.2). Those results should be carefully interpreted since some of the strengths of our method are only reflected in a semi-automatic scenario. For this reason, we later discuss the experiences of our system on a real environment (Section 4.5.3).

4.5.1 Setting Parameters and Implementation

Parameters p_1 and p_2 that determine the probability of applying each operator in the neighbour exploration of SA were selected after designing some preliminary experiments. Parameters $p_1 = 0.4$ and $p_2 = 0.8$ were empirically verified as the most appropriate to our experiments. Thus, the probability of adding a new template is greater (0.8) than that of removing any (0.2). This configuration seems adequate due to removing templates cause the search space to be bounded. Regarding the maximum length of sentences considered, we have set $n = 100$. Finally, the maximum number of evaluations before rebooting was set to $\gamma = 100$.

Parameters that determine our proposal for genetic programming are summarized in Table IV.15.

We implemented our method and the comparison algorithms in Java: 1.6.0 0; OpenJDK 64-Bit Server VM 14.0-b08. We have used *Jama*[34] and *dk.brics.automaton* [Mø10] to implement the measures described in Section 4.3.2.

[32] According to [BP11], a regex could be converted into a DFA in $O(l \cdot \lg_2 l)$ where l is the length of the regex. Also, the calculus of the power matrix depends on the number of states of the equivalent DFA. However, we are not considering those factors into the calculus because they are not dependant on the number of examples n in I.

[33] By using the Thompson's algorithm, the time complexity depends linearly on the length of the regex. As commented before, we are not considering it because it is unrelated to n.

[34] http://math.nist.gov/javanumerics/jama/

Parameter	Value
Population Size	200
Initialization	MDEs + grow
Max initial depth	4
Cross Prob.	0.9
Mutation Prob.	0.05
Selection	tournament selection (5)
Reboot	after 100 it. without replacements
Replacement	*offspring* replaces the *worst* ind. if $Fitness(\{\text{offspring}\}) > Fitness(\{\text{worst}\})$

Table IV.15: Parameter settings of the Genetic Programming algorithms

4.5.2 Experiment 1: Formal Validation

In this section we offer a formal validation of our method. This discussion includes a 10-fold cross validation, a convergence study, and a correlation study. For the sake of simplicity, we present separately the results (Section 4.5.2) and our discussions of the results (Section 4.5.2).

The Data We have applied our method to the three datasets used for evaluating the MDE algorithm (section 2.4). We added an extension of the UGR dataset in order to test the applicability of the algorithm to larger datasets: the *UGRbig dataset*. This dataset is the evolution of the *UGR dataset* in the course of time. It consists of a plenty of query logs collected from 2010 to 2012 from the execution logs of the Virtual Assistant in the UGR web page.

We construct equal proportion and size subsets for each 10-fold cross validation. The subsets were randomly selected. Table IV.16 shows the details of the datasets used in the experiments: columns display the number of entries, reformulations, training and testing sets, and also the proportion of reformulations for each entry (Ref/Entries).

	FAQs	**Reformulations**	**Density(Ref/Entries)**	**Features**	**Training Sets**	**Testing Sets**
Restaurant FAQ	39	400	10.26	297	360	40
Linux V.2.0.2 FAQ	59	450	7.63	263	400	45
UGR FAQ	310	5000	16.13	1300	4500	500
UGRbig FAQ	1023	24853	24.29	5696	22369	2484

Table IV.16: Details of the FAQs used in the experiments.

In order to allow results to become reproducible, the datasets, the partitions used to carry out the 10-fold cross validation in our experiments, and our implementation, are accessible in [35].

Performance of the Experiments The model is tested by performing a 10-fold cross-validation for each dataset. Ten complete validations are performed in order to obtain sufficient results to confirm the results with a t-test. In each 10-fold cross validation, the entire dataset is divided

[35] http://decsai.ugr.es/~moreo/publico/LearningTemplates/Templates_Resources.html

into ten mutually exclusive subsets with the same distribution. Each fold is used once to test the performance of the classifier generated from the combined data of the remaining nine folds.

We evaluate the classification performance of the models by means of common metrics in Information Retrival: precision (P) and recall (R). Precision measures the proportion of correctly classified cases from among all classified cases (see Equation IV.49). Recall measures the proportion of correctly classified cases from among the total number of cases (see Equation IV.50). Finally, the global performance is better reflected in the *F-measure* metric (also known as $F_1 score$), the weighted harmonic mean of precision and recall (Equation IV.51).

$$Precision = \frac{\# \ of \ correct \ queries \ answered}{\# \ of \ queries \ answered} \tag{IV.49}$$

$$Recall = \frac{\# \ of \ correct \ queries \ answered}{\# \ of \ queries} \tag{IV.50}$$

$$F - measure = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{IV.51}$$

To evaluate the quality of the template sets obtained by each algorithm, we use the Correctness (C), Generalization (G), and Interpretation (I) metrics (see Section 4.3.1), and the Fitness (F) metric that combines them all (see Section 4.3.2).

Usually, retrieval algorithms present a trade-off between precision and recall. In this regard, we are interested in studying how each partial metric influences the performance. To this aim, we will use the linear regression and the sample Pearson correlation coefficient (see Equation IV.52) that measures the strength of the linear regression.

$$r_{X,Y} = \frac{\hat{s}_{XY}}{\hat{s}_X \hat{s}_Y} \tag{IV.52}$$

Where $\hat{s}$ represents the sample standard deviation.

Comparison Algorithms The following comparison algorithms have been considered as baselines:

- *Minimal Differentiator Expressions algorithm*: Since MDE algorithm implements a Learner Machine, we have considered it as a comparison algorithm.
- *tf·idf*: This classical method [BCC+00] measures the (cosine) similarity between two word frequency vectors $\overrightarrow{U}$ (the user question) and $\overrightarrow{S}$ (each entry in the domain). This scoring function considers the term frequency (*tf*) and the inverse document frequency (*idf*) of each word. It should be remarked that this method does not require training.
- *Genetic Programming*: In order to contrast the influence of MDEs in the population, we have considered a version of GP whereby the initial population does not contain the MDEs.
- *Classifier-based approaches*: Since the ultimate goal is to determine which is the most relevant FAQ entry to the user question, a natural baseline is to use the FAQ to train a classifier. Among the various classifiers in the literature, we have chosen SVMs and AdaBoost. On the one side, [ZL03] reported SVMs as the most effective method to classify questions while

compared against other well-known methods in the literature including Nearest Neighbours, Naïve Bayes, Decision Trees, and Sparse Network of Winnows. On the other side, AdaBoost [FS95, FS96] is another classifier aimed for iteratively boosting previous misclassified examples, that has been successfully applied to many text-classification problems [EFS08]. According to these works, we explored the effect of Bag of Words —(W) in tables— against Bag of Ngrams —(N) in tables— while representing the features. We used JATECS[36] to implement these methods.

Note that some of the comparison methods do not obtain templates, nor any other interpretable resource either. Thus, it should be pointed out that they will only be here considered for the purpose of performance comparison. Therefore, regardless of their performance scores, those methods are not real alternatives for the problem under consideration.

Results of the Study Table IV.17, Table IV.18, Table IV.19, and Table IV.20 present the results obtained after carrying out the experiments. In this experiment, we have weighted all parameters equally in order to show its performance. However, since the correctness degree is arguably the most influential parameter to the performance, we set double weight to it. Thus, expert criteria were set to $W_1 = 0.5$, $W_2 = 0.5$, $W_3 = 1.0$, $W_4 = 0.5$, and $W_5 = 0.5$. The ideal regex in terms of generalization was set to $\widehat{r} = which * (email|telephone|fax) * (manager|director)$. Finally, optimisation algorithms were performed over 1000 iterations.

Figures iv.18, iv.19, iv.20, and iv.21 show the traces of convergence of the proposed optimisation algorithms in one of the experiments. Note that evolutionary algorithms were performed over 2000 evaluations in these experiments in order to bring off a more detailed discussion on their convergence. To clarify the visualization, the objective function is shown in terms of *energy* or *standardized fitness*, so lower values represent better solutions. Moreover, energy (E) is broken down into correctness (C), generalization (G), and interpretability (I) partial degrees.

Table IV.21 summarizes the average proportion of templates for each FAQ entry in all domains (left-most) and exemplifies the gradual change of this proportion in the course of time for the case of Restaurant dataset (right-most).

We have investigated the influence of Correctness and Generalization on Precision and Recall by means of the sample Pearson correlation degree (see Equation IV.52) and trend lines. To study the influence of Correctness (C), we fixed parameters as reported in Section 4.5.1 and we ranged W_3 (the Correctness weight) between 0 to 5.0. Similarly, to study the influence of Generalization (G), we set W_4 (the Generalization weight) to 5.0, and we ranged $generalization(\hat{r}, n)$ between 0 (using $\hat{r} = \epsilon$) to 1 (using $\hat{r} = *$). As results in all datasets were consistent, and for the sake of simplicity, we report only the case of UGR dataset (Figures iv.22 and iv.23). Table IV.22 shows the sample Pearson correlations degrees for the UGR dataset for the proposed algorithms (SA and GP$_{MDE}$).

We have not investigated influence of Interpretability (I) on Precision or Recall because Interpretability it is not in essence related to performance, but rather to the expert's preferences —interpretability will be treated in Section 4.5.3.

Interpretation of Results MDE algorithm shows the best global classification performance among the compared template-based algorithms in terms of $F - measure$ (F_m). However it is not surprising that MDE obtains lower Fitness (F) than optimisation algorithms. As could be

[36] JATECS (JAva library for TExt Categorization, `http://hlt.isti.cnr.it/jatecs/`) were generously supplied to perform our experiments by Fabrizio Sebastiani, Andrea Esuli, and their group.

Linux	**P**	**R**	**F$_m$**	**C**	**G**	**I**	**F**
SA	89.339 ± 4.360	54.896 ± 6.146	67.862 ± 5.322	0.531 ± 0.013	0.957 ± 0.001	**0.684** ± 0.004	**0.676** ± 0.007
GP_{MDE}	90.186 ± 7.265	45.047 ± 6.065	59.819 ± 5.913	**0.631** ± 0.015	0.925 ± 0.004	0.453 ± 0.012	**0.660** ± 0.005
GP	92.580 ± 6.237	35.085 ± 7.535	50.331 ± 8.110	0.607 ± 0.019	0.832 ± 0.020	0.463 ± 0.016	0.627 ± 0.010
MDE	77.530 ± 3.600	**73.758** ± 3.352	75.579 ± 3.274	0.179 ± 0.007	**0.970** ± 0.000	0.413 ± 0.006	0.435 ± 0.005
TF IDF	75.381 ± 7.282	68.549 ± 6.491	71.758 ± 6.588	-	-	-	-
SVM$_{(W)}$	89.628 ± 3.961	67.800 ± 7.482	**77.007** ± 5.225	-	-	-	-
SVM$_{(N)}$	**95.578** ± 2.262	64.417 ± 8.208	76.685 ± 5.969	-	-	-	-
AdaBoost$_{(W)}$	76.622 ± 6.444	68.588 ± 5.488	72.329 ± 5.580	-	-	-	-
AdaBoost$_{(N)}$	85.792 ± 7.274	69.621 ± 3.846	76.780 ± 4.624	-	-	-	-

Table IV.17: Results for Linux domain

Rest.	**P**	**R**	**F$_m$**	**C**	**G**	**I**	**F**
SA	**92.574** ± 5.490	62.926 ± 5.556	74.849 ± 5.281	0.444 ± 0.015	**0.956** ± 0.001	**0.702** ± 0.007	**0.635** ± 0.010
GP_{MDE}	86.530 ± 7.231	59.250 ± 8.461	70.198 ± 7.818	0.510 ± 0.017	0.940 ± 0.006	0.526 ± 0.012	**0.621** ± 0.005
GP	77.754 ± 12.50	36.720 ± 7.739	49.683 ± 9.176	**0.568** ± 0.023	0.858 ± 0.022	0.466 ± 0.015	0.586 ± 0.014
MDE	86.428 ± 5.359	79.713 ± 6.042	82.901 ± 5.519	0.130 ± 0.010	**0.964** ± 0.000	0.429 ± 0.008	0.411 ± 0.009
TF IDF	83.941 ± 7.956	77.829 ± 8.320	80.721 ± 7.898	-	-	-	-
SVM$_{(W)}$	85.823 ± 2.931	72.804 ± 7.586	76.968 ± 6.823	-	-	-	-
SVM$_{(N)}$	84.803 ± 2.858	71.036 ± 6.186	75.631 ± 6.612	-	-	-	-
AdaBoost$_{(W)}$	87.557 ± 3.424	**82.905** ± 5.289	**85.074** ± 3.456	-	-	-	-
AdaBoost$_{(N)}$	89.130 ± 6.793	74.618 ± 5.318	81.158 ± 5.450	-	-	-	-

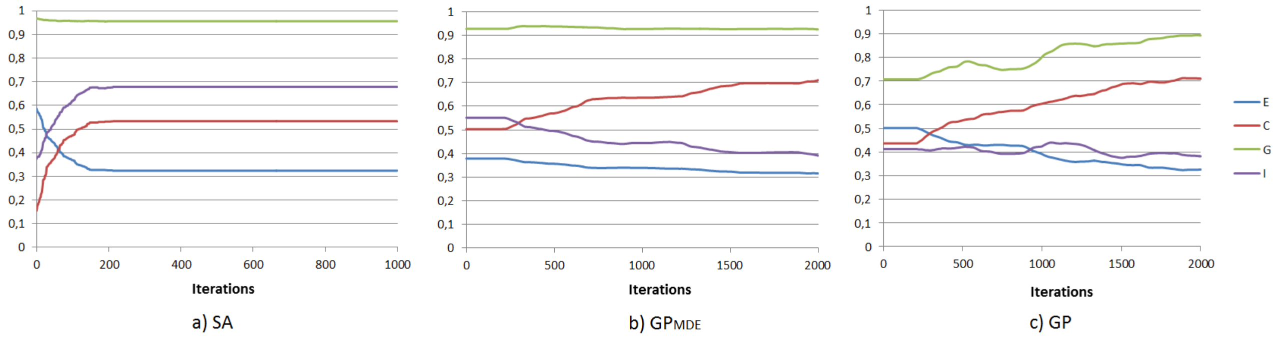
Table IV.18: Results for Restaurant domain

UGR	**P**	**R**	**F$_m$**	**C**	**G**	**I**	**F**
SA	86.018 ± 1.209	59.389 ± 3.188	70.234 ± 2.546	0.483 ± 0.004	0.902 ± 0.029	**0.567** ± 0.026	**0.607** ± 0.012
GP_{MDE}	91.184 ± 0.628	63.447 ± 2.694	74.796 ± 1.833	**0.519** ± 0.003	0.931 ± 0.001	0.509 ± 0.001	**0.620** ± 0.001
GP	87.050 ± 2.047	25.664 ± 2.957	39.567 ± 3.678	0.497 ± 0.088	0.747 ± 0.158	0.502 ± 0.042	0.546 ± 0.051
MDE	83.194 ± 2.012	**79.844** ± 2.293	81.483 ± 2.122	0.110 ± 0.003	**0.965** ± 0.000	0.358 ± 0.001	0.382s ± 0.002
TF IDF	51.432 ± 4.764	22.517 ± 1.660	31.307 ± 2.402	-	-	-	-
SVM$_{(W)}$	85.231 ± 3.446	70.273 ± 1.174	76.656 ± 2.001	-	-	-	-
SVM$_{(N)}$	86.207 ± 1.119	72.175 ± 1.497	78.585 ± 2.387	-	-	-	-
AdaBoost$_{(W)}$	88.178 ± 1.084	72.542 ± 1.701	79.590 ± 1.219	-	-	-	-
AdaBoost$_{(N)}$	**93.775** ± 1.426	75.028 ± 1.626	**83.349** ± 1.195	-	-	-	-

Table IV.19: Results for UGR domain

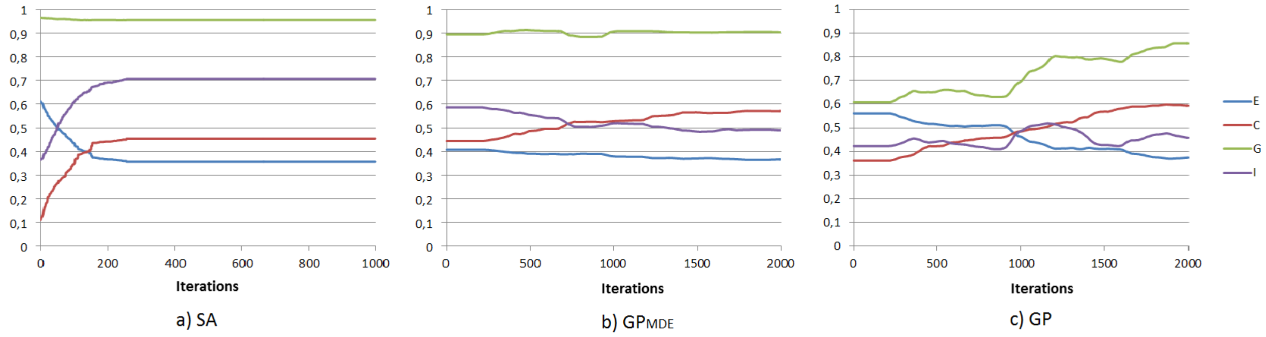
UGRbig	**P**	**R**	**F$_m$**	**C**	**G**	**I**	**F**
SA	74.011 ± 0.882	47.685 ± 4.707	57.924 ± 3.766	0.533 ± 0.078	0.882 ± 0.091	**0.538** ± 0.014	0.597 ± 0.030
GP_{MDE}	81.187 ± 8.847	50.562 ± 2.434	61.976 ± 1.992	**0.583** ± 0.004	0.838 ± 0.002	0.498 ± 0.002	**0.626** ± 0.001
GP	83.979 ± 10.333	31.442 ± 9.289	44.345 ± 4.962	0.556 ± 0.002	0.662 ± 0.004	0.513 ± 0.003	0.572 ± 0.001
MDE	74.237 ± 0.841	**70.591** ± 0.811	72.731 ± 0.816	0.215 ± 0.004	**0.962** ± 0.000	0.370 ± 0.370	0.395 ± 0.002
TF IDF	13.156 ± 0.637	8.234 ± 0.389	10.128 ± 0.477	-	-	-	-
SVM$_{(W)}$	83.739 ± 0.959	61.809 ± 0.792	70.319 ± 1.165	-	-	-	-
SVM$_{(N)}$	84.797 ± 0.605	64.631 ± 0.858	72.648 ± 1.061	-	-	-	-
AdaBoost$_{(W)}$	85.614 ± 0.663	62.095 ± 0.372	71.981 ± 0.404	-	-	-	-
AdaBoost$_{(N)}$	**86.690** ± 0.555	66.902 ± 0.566	**75.520** ± 0.457	-	-	-	-

Table IV.20: Results for UGRbig domain



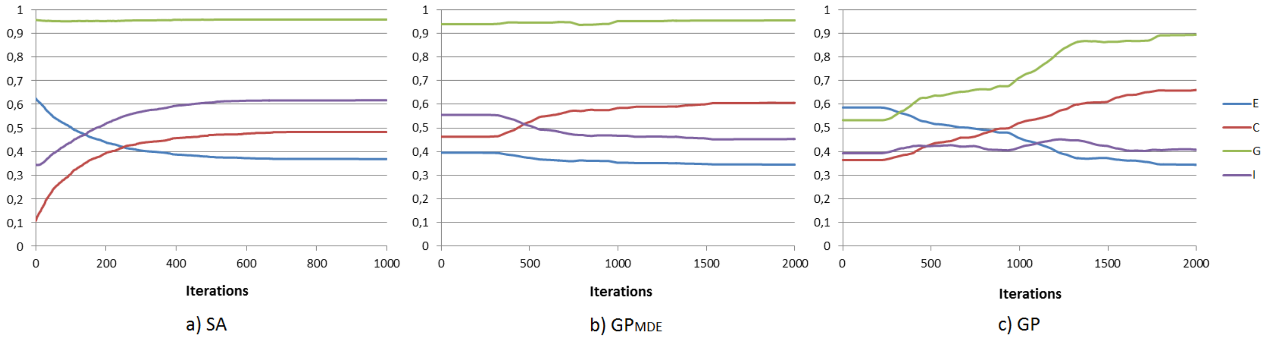
a) SA b) GPMDE c) GP

Figure iv.18: Convergence in Linux dataset



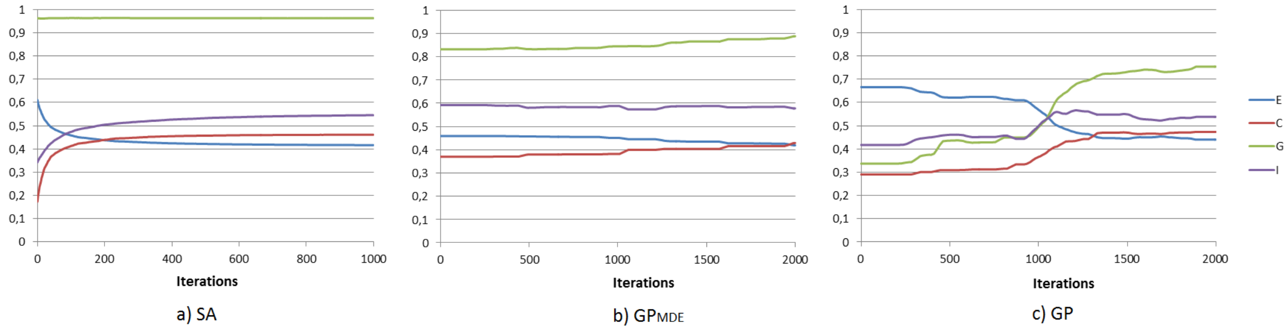
a) SA b) GPMDE c) GP

Figure iv.19: Convergence in Restaurant dataset



a) SA b) GPMDE c) GP

Figure iv.20: Convergence in UGR dataset



a) SA b) GPMDE c) GP

Figure iv.21: Convergence in UGRbig dataset

	Dataset			
Algorithm	**Linux**	**Restaurant**	**UGR**	**UGRbig**
SA	3.03	2.91	8.15	22.83
GP$_{MDE}$	2.22	2.84	3.41	8.93
GP	2.54	2.83	4.59	16.85
MDE	12.69	16.84	38.62	41.01

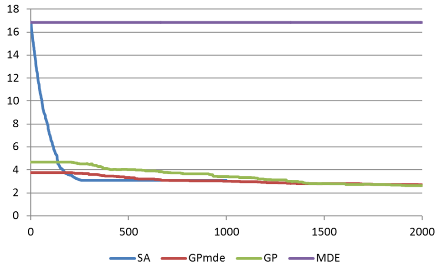


Table IV.21: Average proportion of Templates for each FAQ entry and gradual change in Rest. dataset

Pearson Corr. SA	**Pearson Corr. GP$_{MDE}$**
r_{GP} = -0.75	r_{GP} = -0.81
r_{GR} = 0.77	r_{GR} = 0.18
r_{CP} = 0.72	r_{CP} = 0.70
r_{CR} = -0.12	r_{CR} = -0.88

Table IV.22: Sample Pearson Correlation degrees [UGR dataset]

observed, correctness (C) and interpretation (I) degrees of MDEs are substantially lower than that of optimisation algorithms. Consequently, although MDE performs more robustly as a Machine Learning method, even for big domains, revising/modifying templates might be easier for the other methods. In addition, some authors may claim that in some scenarios precision is more important than recall. In this regard it should be remarked that optimisation algorithms (SA, GP$_{MDE}$, and GP) have obtained good results in precision (P). In any case, SVMs and AdaBoost exhibit the best global retrieval performance. However, because those methods do not generate any kind of template, they are not directly applicable to the problem under consideration. In any case, in light of the results, it is worth remarking that MDE's performance is comparable.

It is clear that the optimisation algorithms obtained the best results in the Fitness metric. In this regard, it was statistically contrasted that MDE-based methods (SA and GP$_{MDE}$) have outperformed significantly the GP algorithm. That evidences the validity of considering MDEs as a promising starting point. However, in light of these results it is not clear which of the MDE-based methods is the best one. Results indicate that SA performs better in smaller domains (Linux and Restaurant datasets), while GP$_{MDE}$ was preferable in bigger ones (UGR and UGRbig dataset). This claim is consistent with our observations on convergence: SA converges faster than GPs to local optima. In bigger domains, the fast convergence of SA has proven to be less effective.

As could be observed, MDEs present the highest generalization (G) degree among all comparison methods. This fact was also reflected in the generalization degrees of MDE-based methods. Similarly, MDEs show the lowest correctness degree, which also influences the correctness (C) degree of SA algorithm. GP$_{MDE}$ was not so affected in correctness degree because this algorithm counted also with additional genetic information in the initial population. Finally, interpretation (I) degree of templates obtained by SA were significantly greater than all other comparison algorithms. Thus, SA may be the most appropriate algorithm for experts preferring interpretability as the most desirable feature.

As it was shown, SA converges much faster than GP methods. In our experiments, it was found

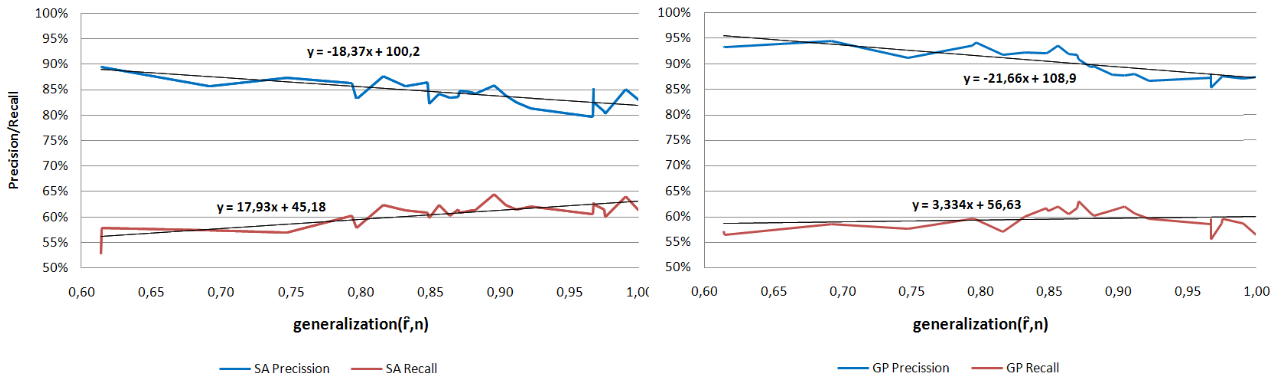


Figure iv.22: Influence of Generality [UGR dataset]

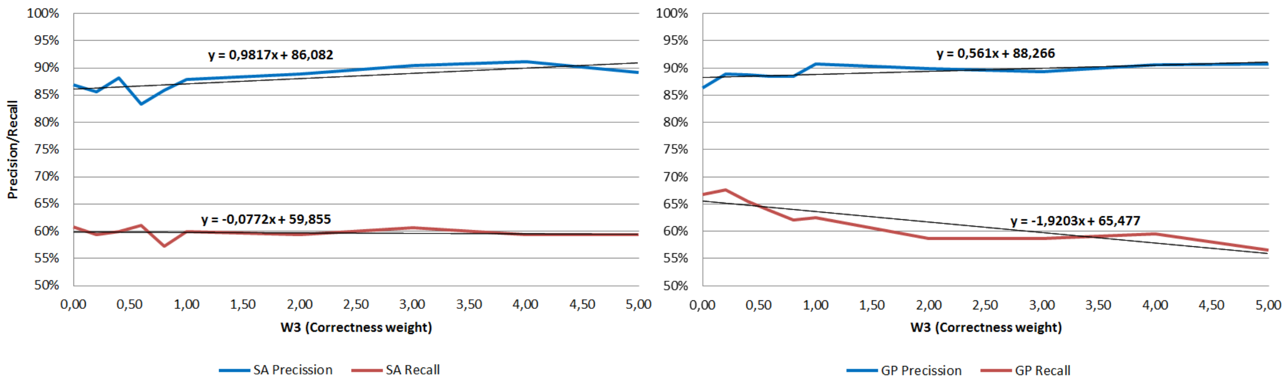


Figure iv.23: Influence of Correctness [UGR dataset]

that SA was stabilized after (approx.) 300 evaluations in small domains, 700 evaluations in the UGR dataset, and 900 evaluations in the UGRbig dataset. It takes much longer for evolutionary algorithms to be stabilized, but it seems that GP algorithms are able to obtain better results in the long term. Moreover, the benefits of considering MDE as part of the initial population is clearly exposed in the convergence traces. It takes much longer for GP to obtain comparable results to that of GP_{MDE}.

It is noticeably that interpretability (I) degree decreases in the course of time for GP_{MDE}. Interpretability depends on the number of templates and the structural complexity. Since MDEs are structurally very simple, it is clear that genetic operators increment its structural complexity. Thus, this is an indication that subsequent greedy coverages do not vary too much in terms of size while structural complexity is being augmented. In addition, correctness and interpretability tendencies seem to be paired in MDE-based methods. The fact that curves in SA evolved in parallel is striking. The reason is that each time a promising regex (in terms of correctness) is found, two old regexes could be removed. As a result, the size complexity, and therefore the interpretability, is also benefited.

SA always takes as starting point the (deterministic) set of MDEs, GP relies on randomly-generated individuals, and GP_{MDE} combines both features. As reflected in standard deviations, metrics are less affected by randomness in SA than in evolutionary algorithms, being the GP algorithm the most affected in this regard.

As it was expected, the degree of generalization influences the retrieval performance of the algorithms. The more generalizable, the better recall at expense of precision. As reflected in Figure iv.22, SA deals better with generalization —likely because, as stated before, it is more influenced by MDEs. Also, the trade-off between precision and recall is affected by the Correctness degree. As its weight increases, precision grows and recall decreases. However as reflected in the slopes of the lines, this dependency is less pronounced. The reason is that, in this case, we are ranging the *importance* (weight) of correctness —in any case, the algorithm tries to optimize the correctness measure at some extent.

Regarding the order of efficiency of the optimisation algorithms, evaluations are noticeably faster in SA than in evolutionary algorithms in small datasets, such as Restaurant and Linux. For example, it took approx. 63 seconds for SA to calculate the templates for Linux dataset, while genetic algorithms spent over 28 minutes. That is because each time a new regex is obtained in evolutionary algorithms, the algorithm should test whether any of the negative examples is accepted. In contrast, this test should only be carried out in SA after disjunctive-join operation. In any case, the hidden constants (in our implementation) took their tool in the long term, and the time spent by both algorithms did not vary significantly in UGR (around 2 hours) or UGRbig datasets (about 24 hours).

At this point, it should be remarked that the concrete optimisation strategies here proposed should only be regarded as two possible options. It was not among our aims to claim about the superiority of a particular optimisation algorithm. Rather, those examples may have served to illustrate the applicability of the theoretical study conducted in section 3.4 and the convenience of the optimization measure drawn in section 4.3.

4.5.3 Experiment 2: Real Environment

In this section we focus on validating the proposed method in a real environment. Our aim is to test whether the methodology is actually useful from the experts' viewpoint. To this purpose, we counted with four knowledge engineers from *Virtual Solutions*[37], a company with experience in commercial template-based QA methods.

The Data We chose the Linux, Restaurant, and UGR datasets from Experiment 1 (see Section 4.5.2) because they are big enough for the experiment and they could still be handled to some extent. All parameters except the expert criteria ones, were set as described in section 4.5.1.

Performance of the Experiments In order to verify which optimisation algorithm generates better templates in practice, we asked the experts to evaluate the usefulness of all the templates generated by the SA and the GP$_{MDE}$ algorithms. The experts analysed by hand all the regular expressions generated by both algorithms for the proposed datasets, labelling them as *useful* or *useless*. A template was considered to be *useful* if it could be included in the system as it was or making little modifications (to make it more general or to prevent generalization in excess). On the other hand, a template was considered to be *useless* if it did not capture the domain knowledge well.

Before performing the validation, some indications about the configuration of the system were given to the experts. According to their judgements, correctness was the most important feature, followed by size complexity. They played down the importance of structural simplicity and gener-

[37] `http://www.solucionesvirtuales.es/`

alization —however, they set $\widehat{r} = *$ to favour recall. Finally, the experts set the fitness metric as follows in order to reflect their preferences:

$$Interpretability(R) := 0.1 \cdot Structuralsimplicity(R) + 0.9 \cdot Sizesimplicity(R) \tag{IV.53}$$

$$Fitness(R) := \frac{2 \cdot Correctness_{I^+,I^-}(R) + 0.5 \cdot Generalization_{\widehat{r}}(R,n) + 1 \cdot Interpretability(R)}{3.5} \tag{IV.54}$$

Results of the Study Figure iv.24 shows how useful the templates generated by the optimisation algorithms are in practice, when experts have to deal with them in order to incorporate that knowledge into a real system. Figure iv.24 (left) depicts the total number of templates generated by each algorithm and each dataset. Figure iv.24 (right) shows the total number of useful templates labelled by the experts (vertical bars) and the percentage of useful templates with respect to the total number of templates generated (labels inside the bars).

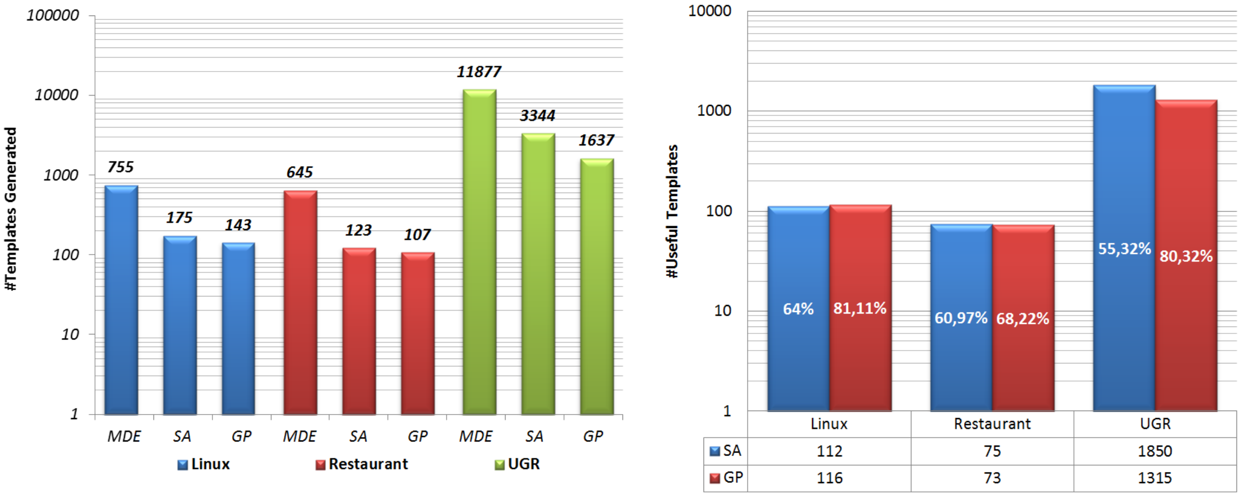


Figure iv.24: Number of templates generated and reports on usefulness

Interpretation of Results In this section, we report the opinion of the experts based on their experiences with the system. Accordingly, the reader should be aware that those results have a slightly subjective nuance. Moreover, it is worth pointing that the experts' opinions are based on their notions of usefulness, rather than on a formal validation.

As stated by the experts, the proportion of useful templates in SA is much lower than in GP$_{MDE}$. This may be caused by the faster convergence of SA with respect to GP —local solutions reached early, are not further improved. In addition, although the interpretability is more or less the same with both algorithms, GP$_{MDE}$ generates much smaller template sets. For these reasons, the experts considered that using the GP$_{MDE}$ algorithm was much more appropriate than SA for bigger domains like the UGR datasets —this indication is consistent with results in F (see Tables IV.19 and IV.20). It was also pointed out that, in smaller domains, none of these algorithms were clearly preferable. In any case, they favoured GP algorithm even for smaller domains. The experts were also asked about their impressions on the (plain) MDEs. In this regard, it was stated that the high number of templates obtained by each FAQ entry made predicting the behaviour of the system extremely complex for a human, even considering that each individual template is quite simple. Thus, manually debugging these templates could entail much effort.

Finally, the experts agreed that using these algorithms would help to reduce significantly the time spent while designing a template-based system. Although further experiments approximating how much time could be saved will be a goal for our future research, the experts estimated that this tool could reduce from 40% to 50% of the work.

They supported our hypotheses, according to which it takes less effort to validate templates, than to create them. The experts highlighted that the most important advantage is that this algorithm could provide the company with an initial set of operative templates much faster than by means of manual methods, reducing approximately about 80% of this initial effort.

4.6 Conclusions and Future Work

Although Template-based methods entail higher human effort, companies could prefer to undertake the associated costs in order to offer a better service. In this section, we have addressed the problem of how to alleviate those costs while keeping the system reliability, and we have put our method to test in a real environment.

Theoretically motivated properties (section 1.4) have been formalized as correctness, generalization, and interpretability degrees. Since resulting templates should be revised by an expert, those parameters are weighted according to his/her judgements. For this reason, templates are evaluated by means of an adjusted metric, in such a way that the problem could be addressed as an optimisation problem. Experiments indicated that our proposal is useful in real environments, providing experts with templates that need minor modifications to compose robust FAQ retrieval systems. As a result, time spent and associated costs are substantially decreased while the system is still monitored by an expert.

Regarding the objective function, two well-known optimisation methods have been adapted in this study: Simulated Annealing and Genetic Programming. While neighbour operators proposed here for SA are linguistically motivated, evolutionary mechanisms showed better trade-off between diversity and convergence. Experiments lead us not to adopt any of these algorithms as the most suitable with independence of the domain. It should be remarked that presenting an optimisation algorithm was not among our prior goals. Rather, we were interested in framing the problem as an optimisation problem, presenting a suitable formulation aimed to capture some relevant aspects. In addition, it is clear that differentiability and minimality criteria reflected in MDEs contain valuable information that could be fruitfully exploited for the purpose.

However, although results are encouraging, there is still much work ahead. For future research, we aim to automatically obtain the weights that determine the preferences for each expert. We plan to do this by mining sets of templates previously created by a specific expert. We believe that it would be a valuable contribution since the expert will no longer be expected to understand each of these parameters individually. Since it was observed that different optimisation mechanisms could be preferable depending on the domain nature, we are interested in studying automatic methods to characterize question datasets, so that we could count in advance with sufficient information to determine which of the methods is the most appropriate.

5 Final Discussions: Applications and Future Work

In this section, we offer some final remarks focusing on different applications and contributions developed by our research group with respect to the topic at hand. What should be highlighted is that the author of this dissertation is not the main author of these contributions. Rather, they are some related working projects on which the candidate collaborated actively.

First, we will discuss in section 5.1 the suitability of the regular expressions to improve the performance of a real Virtual Assistant. In section 5.2 we briefly describe a related FAQ retrieval system aimed for optimising the navigation through tag clouds. Section 5.3 is devoted to offer some remarks on a collaborative system for e-Learning. We conclude this part of the dissertation highlighting the main goals proposed and the results obtained and proposing some future research.

5.1 Template-generation for Virtual Assistance

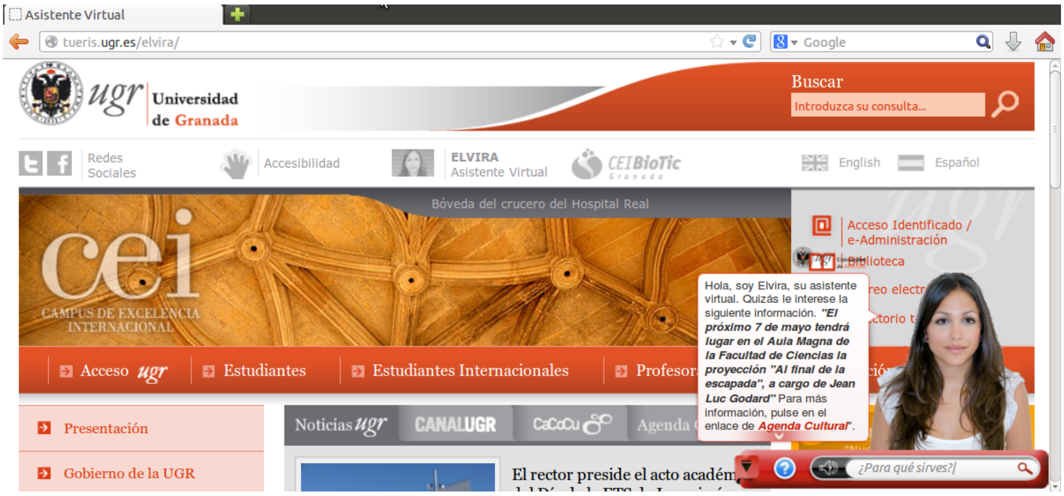
Virtual Assistants are intelligent agents devoted to answer questions about a given domain. The main goal beyond these agents is to offer a more natural interface to the information stored in a domain. As a result, the user could remain unaware of the particular structure of the web-side, that is, his/her information needs are formulated and answered through NL, avoiding traditional click-navigation searches. Additionally, Virtual Assistants could recommend related information of interest.

In [ELC12], a framework for designing closed domain virtual assistants was proposed by some members of our research group. There were two well-differentiated knowledge sources in this framework. On the one side, each possible information unit was individually covered by a set of regular expressions. On the other side, a domain ontology served to represent all meta-knowledge in the domain. Thus, according to the classification proposed in section 4 this system falls among the SK/PM approaches. However, there are reasons justifying its inclusion here. The so-called *Natural Language Understander* module was in charge of recognizing the user questions. This module relies solely on the set of regular expressions to match the user question and to decide which of the information units should be retrieved. Thus, apart from the fact that there is an ontology describing the knowledge structure and relations, the truth is that the question matcher was implemented as a template-based approach, regardless in principle of the ontology.

Templates of the early versions of the system were designed manually. We are currently working on applying the method proposed in section 4 to assist experts in designing these templates. Thus, alleviating significantly the effort needed to maintain future versions (Figure iv.25). Although there is still some work ahead to achieve this goal, preliminary progresses indicate the method is being actually useful.

5.2 High-Precision and Visual Navigation for FAQ retrieval

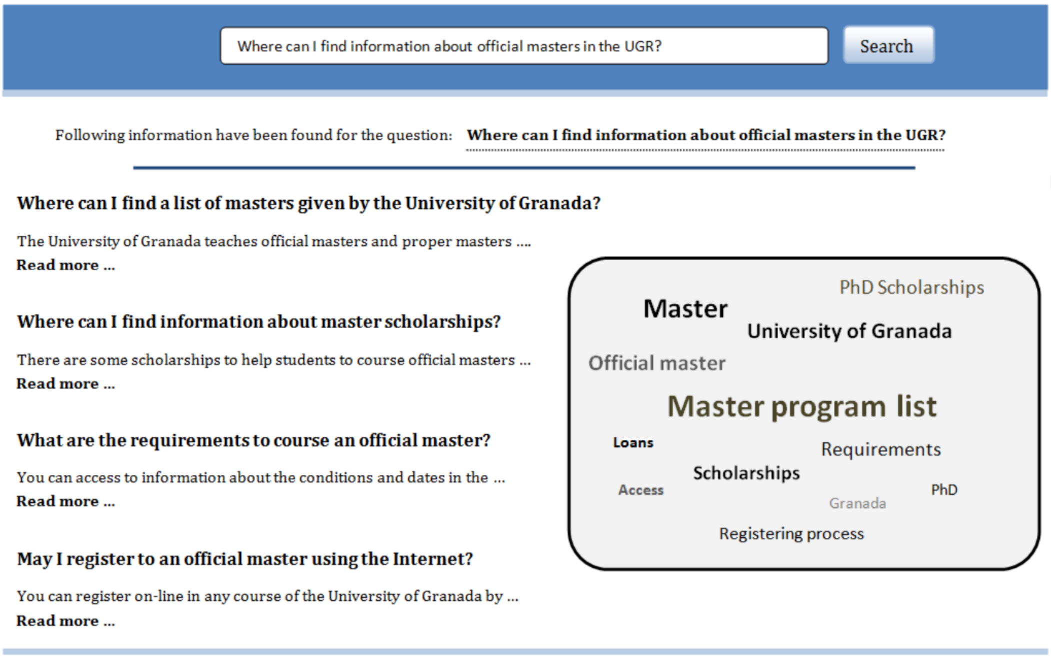
In [RMC13a] we presented a new FAQ retrieval method to improve the navigability through the domain of knowledge in order to facilitate both learning and searching. The system counted with a highly-precise FAQ retrieval engine. It was favourably compared against the MDE algorithm in terms of precision and time consumption, although it presented lower MRR score. Thus, this system could be preferable in scenarios where the training times are restrictive. In any case, the more interesting feature of this system concerned with the Tag Cloud generation module. This module was able to extract some important key concepts from the FAQ conforming a tag cloud



Hola, soy Elvira, su asistente virtual. Quizás le interese la siguiente información. *"El próximo 7 de mayo tendrá lugar en el Aula Magna de la Facultad de Ciencias la proyección "Al final de la escapada", a cargo de Jean Luc Godard"* Para más información, pulse en el enlace de *Agenda Cultural*".

Figure iv.25: Virtual Assistant of the University of Granada.

that helps the user to navigate through the retrieved information and to obtain a visual depiction of informative content in the FAQ (Figure iv.26). Those concepts were extracted as Weighted Significant Semantic Units (WSSU) by exploiting WordNet and Wikipedia resources.



Where can I find information about official masters in the UGR?

Following information have been found for the question: **Where can I find information about official masters in the UGR?**

Where can I find a list of masters given by the University of Granada?

The University of Granada teaches official masters and proper masters ....
Read more ...

Where can I find information about master scholarships?

There are some scholarships to help students to course official masters ...
Read more ...

What are the requirements to course an official master?

You can access to information about the conditions and dates in the ...
Read more ...

May I register to an official master using the Internet?

You can register on-line in any course of the University of Granada by ...
Read more ...

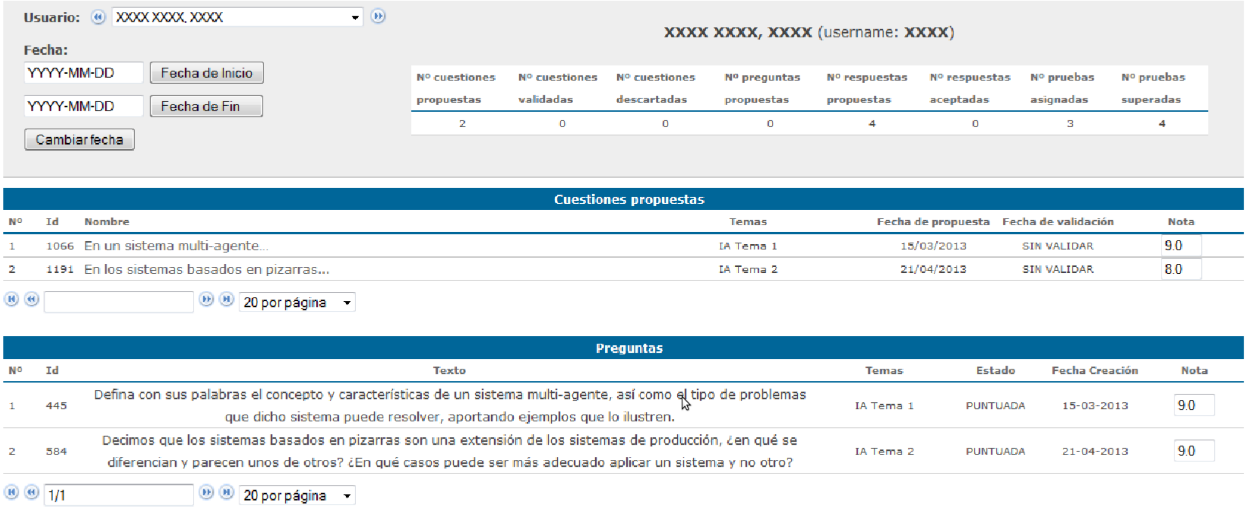
Figure iv.26: Example of a FAQ cloud

We are interested in using this new method as an alternative to the TF·IDF method in FAQtory (see section 3.4). Thus, the MDE method would perform in parallel with this algorithm. This may result in similar training times —this method is, loosely speaking, as fast as the TF·IDF to the problem at hand— while the navigation through the FAQ could be effectively improved.

5.3 Collaborative Learning Systems

Collaborative systems such as the one proposed in chapter 3 are becoming more and more popular as E-learning education tools to improve the learning and teaching activities [MRPVR07]. In this regard, a wide range of computer-based learning environments have been developed in the field of Information and Communication Technologies (ITS). Thanks to these tools, usually known as Virtual Learning Environments (VLE) or Learning Management Systems (LMS), teachers and learners can improve their learning interactivity, overcoming physical distance and time constraints. A VLE is often regarded as a platform designed to manage the learning processes by providing students with sufficient resources to reach their learning goals. VLEs are expected to provide a shared framework in which teachers are allowed to design and instantiate individual and collaborative learning activities. Following a constructivist framework, the learning resources are not only constructed and managed by teachers, but also by students that take an active part on the contribution of the information space. In order to avoid any kind of technical restriction that could diminish the collaborative experience, meta-knowledge resources, such as domain ontologies, are usually absent in these systems. In contrast, tasks, activities, problems, exams, or so on, are examples of knowledge resources presenting certain inner structure.

In this line, we are working on a Collaborative System for Learning based on Questionnaires and Tasks [RMC13b]. Because the system does not need any meta-knowledge, it becomes independent of any course structure or content. It offers functionalities to create, review, and evaluate new knowledge resources through questions and tasks. This tool allows teachers be released from the tedious task of creating all resources, while encourage students to gain the necessary background knowledge before creating any new content. It also counts with a Fuzzy controller devoted to generate exams that satisfy a customized set of pre-selected objectives. These exams could be used for final evaluation as well as for auto-evaluation purposes. We are currently using this tool in real courses of the University of Granada. Albeit it is still a very early version of the software, everything seems to indicate the tool is actually useful to improve the learning process.



XXXX XXXX, XXXX (username: XXXX)

Nº cuestiones propuestas	Nº cuestiones validadas	Nº cuestiones descartadas	Nº preguntas propuestas	Nº respuestas propuestas	Nº respuestas aceptadas	Nº pruebas asignadas	Nº pruebas superadas
2	0	0	0	4	0	3	4

Cuestiones propuestas

Nº	Id	Nombre	Temas	Fecha de propuesta	Fecha de validación	Nota
1	1066	En un sistema multi-agente...	IA Tema 1	15/03/2013	SIN VALIDAR	9.0
2	1191	En los sistemas basados en pizarras...	IA Tema 2	21/04/2013	SIN VALIDAR	8.0

Preguntas

Nº	Id	Texto	Temas	Estado	Fecha Creación	Nota
1	445	Defina con sus palabras el concepto y características de un sistema multi-agente, así como el tipo de problemas que dicho sistema puede resolver, aportando ejemplos que lo ilustren.	IA Tema 1	PUNTUADA	15-03-2013	9.0
2	584	Decimos que los sistemas basados en pizarras son una extensión de los sistemas de producción, ¿en qué se diferencian y parecen unos de otros? ¿En qué casos puede ser más adecuado aplicar un sistema y no otro?	IA Tema 2	PUNTUADA	21-04-2013	9.0

Figure iv.27: Collaborative System for Learning.

In the future, we plan to join the functionalities of this system with that proposed in FAQtory. Our aim is to bring students the possibility to use the entire knowledge base as a learning resource to be directly consulted through NL queries.

5.4 Final Remarks

Closed-domain approaches under the SK/AM configuration represent a promising scenario for collaborative systems. The present chapter of this Thesis was devoted to address different problems that arise in this kind of systems. Concretely, following topics have been considered: how to allow an incremental system retrieve efficiently the information units (section 2), how the system could be aware of its own knowledge gaps in order to assist the improvement of the knowledge resources (section 3), and how the performance could be manually refined in critical systems (section 4). All these investigations were both empirically and theoretically supported. Finally, we have presented several working projects including the maintenance of a real Virtual Assistant, a proposal for improving navigation through Tag Clouds, and a collaborative system for e-Learning, that will represent the avenues of our recent future research in SK/AM systems.

Because no meta-knowledge resource is needed, this configuration allow to design truly collaborative approaches, regardless of the technological level of their users. Thus, users should not be experts in NLP or computational linguistics either. They do only have to follow the system's policy to create new knowledge resources, such as new FAQ entries, new tasks, new exams, or so on. As a result, these systems become easily incremental.

Our study provides a means for achieving scalable FAQ systems. We believe these techniques could be extensible to the broader class of SK/AM problems by properly adapting the differentiability criterion to the specific IUs at hand. Mining users reactions while interacting with the system could reveal information on their satisfaction that could be exploited to improve the knowledge content. We think Usage Mining techniques here proposed could be potentially useful in different e-learning collaborative systems such as VLE and LMS. We have also proposed techniques for generating interpretable templates for the FAQ retrieval problem in order to alleviate the costs associated to its creation. In addition, these templates allow the system to be monitored by experts, who can revise and modify the patterns if needed. It would be interesting to our eyes investigating to what extent this theoretical framework is extensible to different SK/AM problems.

The main counterpart of SK/AM is that these systems are usually limited to superficial searches —since there is no meta-knowledge modelling, advanced inferences or reasoning become hardly achievable. In this respect, the last chapter of this dissertation will be dedicated to explore those closed-domain systems that allow reasoning on its own knowledge. This reasoning exploits both the inner structure of the knowledge and the higher level of meta-knowledge to offer more sophisticated NL solutions.

Chapter V

Natural Language Interfaces: Structured Knowledge, Presence of Meta-Knowledge

1 Introduction

So far, we have faced different problems presenting incomplete knowledge-levels from the semantic or structural point of view. These lack of knowledge restrictions could rather be regarded as a relaxation in the requirements the system imposes, in favour of flexibility. In any case, the interpretation capabilities of those systems become quite limited. Natural Language is undeniable affected by a number of complexities such as ambiguity, anaphora, abstraction, imprecision, or vagueness —to which some approaches were already presented in this dissertation. To attempt many of these problems, the system should be able to identify the intended semantic among the multiple possible interpretations.

Additionally, complex queries —the so-called *non-factual* queries— may involve reasoning or deductions that require a high-level knowledge representation. Most interesting examples among this kind of systems include *Virtual Assistants* [ELC12, ELC09], *Virtual Simulated Patients* [LEC08, SPG09], and *Natural Language Interfaces* [ART95, PRGBAL$^+$13]. As commented before (chapter I section 2) these systems provide high-level interpretations of NL to different purposes, including searching and recommending useful information in a web-domain, the diagnostic training for health care courses, or a means for querying a DataBase through NL.

Reaching a suitable meta-knowledge level may entail however a sizeable amount of effort for experts who, to cap it all, are requested to be NLP or computational linguistic experts. The increasing number of potential applications along with the considerable amount of effort it could entail made it appealing the necessity of rich approaches allowing reasoning on its own knowledge and facilitating, as much as possible, the task of creating the appropriate knowledge resources.

This part of the dissertation is arguably the most closely related to the linguistic phenomenon. Our aim is to design methods able to improve both its linguistic capabilities and its domain-knowledge representation. To this purpose, we will rely on context-free grammars and ontologies, respectively. This scenario will bring us the opportunity to investigate grammar inference methods supported by semantic knowledge representations. From a cognitive point of view, grammatical inference is related to human language acquisition processes that were widely studied from different fields including psychology, linguistic, and cognitive sciences. Language acquisition is the mechanism used by humans to acquire the ability of understand and communicate. Some of the main characteristics of this process consist of the ability to continually refine the inferred grammar and the ability of understanding sentences never seen before. How to reach an automatic interpretation of an arbitrary NL sentence is a colossal task (the so-called complete-AI) that will for sure keep busy the scientific community for quite some time. However, restricting the space of interpretations to a given closed-domain reduces significantly the problem. In such scenario, defining a suitable meta-knowledge level and a particular knowledge structure seems to become feasible to some extent.

We are interested thus in studying those phenomena from a computational point of view in a closed-domain scenario. On the one side, the system will not only parse questions, but will also be able to learn from them along the use. On the other side, the inferred grammar will be used as a structural descriptor to allow the system to conjecture about the language. As a result, some unseen expressions could be interpreted based on its previous experience.

According to the proposed methodology, we will approach the SK/PM class through a representative open-problem. It falls beyond the scope of this research to deal with human reactions/behaviour or psycholinguistic issues, both important aspects in Virtual Assistants and Virtual Simulated Patients. Rather, how full interpretations could be derived, how the system could improve with the use, and how the system could reason by analogy on its own knowledge become more interesting challenges to our objectives. Above mentioned motivations suggest Natural Language Interfaces (section 1.1) represent arguably the most convenient problem to our purpose.

The rest of this chapter is structured as follows. We will introduce first a framework for portable NLI based on grammatical parsings (section 2), and secondly a learning algorithm based on human-computer interaction aimed for alleviating the customization effort (section 3). The rest of this introductory section is structured as follows. We describe the problem of NLIs in section 1.1 highlighting the main particular difficulties and types. Later, we formally define the mechanism we will use —the formal grammars— the notation we will use, and its main subtypes. Finally, we offer a brief summary on ontologies, the most extended meta-knowledge resource to capture the conceptualizations and the semantic relations on a domain (section 1.3).

1.1 Natural Language Interfaces: Problem Statement

Natural Language Interfaces (NLI) are systems that provide an intermediate layer to deal with structured data that facilitates querying access to casual end-users. This layer relies on Natural Language techniques as the main mechanism to encapsulate the underlying formal query language (RDF, OWL, SPARQL, SQL, ...). According to [KB07, DB09], main difficulties to deal with in NLIs could be summarized in:

- Trying to capture the linguistic variability, and trying to resolve possible ambiguities could require a tremendous implementation effort.
- Systems exhibiting a good retrieval performance (in *precision* and *recall*) are often strongly tailored to a given domain or application.
- Users' acceptance on NLIs may be dramatically decremented after retrieval errors, even if this rarely occurs.

According to [KB07], among the various existing approaches to NLI, the following ones could be considered the most representative:

Keyword-based approaches: the user is expected to query the knowledge base by means of simple keywords. These systems are also called *naïve* in the sense that only a few set of limited NL processing techniques are implemented. [KBF07] is an example of this approach.

Full English entry: the system is able to deal with complete sentences, that are posed by the user through free-text forms. Thus, advanced NLP techniques should be combined to attempt the correct interpretation of the question. Examples of these systems could be found in [KBZ06, CHH07, DAC10]. In order to override the possible ambiguities, some systems rely on clarification dialogues.

Guided input search: systems like [BKK05, HPS07] allow users to compose queries using a controlled NL input mechanism. Different pop-up boxes are displayed each time the user make a choose in the previous one. Thus, only correct queries are allowed to be generated.

Graphical query interfaces: the query is composed by clicking in the graphical interface to select some elements from the ontology. Thus, a semantic representation of the knowledge is given to the user. *Semantic Crystal* is a system based on *InfoCrystal* [Spo93] that could serve as an example of this kind of approaches.

Finally, some important related concepts in the field, that would also be discussed in this chapter, are listed below:

Customization is the process whereby an expert adapts the NLI to work properly on a given domain. This task may entail the creation of a suited lexicon or ontology to map the main concepts, relations, and lexicalizations, to the existing knowledge resources [DAC10].

Tranportable or Portable systems are those that could be *easily* ported to different domains in terms of customization effort [GAMP87].

Usability refers to the *ease-to-use* property of the system [DB09]. This concept could additionally refer to the users' acceptance of the system.

Habitability concerns with how well the system is capable for interpreting the particular wordings users might employ while querying the system [OMBC06].

The 'bridge the gap' problem: refers to the disagreement between the user's expectations on the model and the actual knowledge represented and its boundaries [Hal06, WXZY07]. This problem could be similarly stated as how to successfully map linguistic expressions to domain concepts and relations during the configuration stage.

1.2 Formal Grammars: Types and Notation

In this section, we offer a brief background on formal grammars and the notation we will use.

1.2.1 Context Free Grammars

Context Free Grammars (CFG) constitute a well-known formal mechanism to parse sentences that have been widely used to face the problem of Natural Language Understanding (NLU). A CFG is a Formal Grammar defined by the quadruple $G = (T, N, R, S)$, where T is a finite set of terminal symbols, N is a finite set of non-terminal symbols, R is a finite set of relations from N to $(N \cup T)^*$

and $S \in N$ is the starting symbol. Elements of R are called *productions* or *rules of the grammar.* For simplicity, we will use here the reduced notation according to which, the symbol '|' denotes disjunction and the symbols '[' and ']' delimit optional subproductions. Productions are usually noted as $V \rightarrow p$ where the left-hand side is a non-terminal symbol $V \in N$, and the right-hand side p is a well-formed formula (*wff*): p is said to be *wff* if it is a single terminal symbol, a concatenation $p_1\ p_2\ \ldots\ p_k$, a disjunction $p_1 \mid p_2 \mid \ldots \mid p_k$, an optional subproduction $[p_1]$, or a natural grouping (p_1), being each p_i also a *wff.*

For any strings u, $v \in (N \cup T)^*$, we say that u derives in v, noted as $u \Rightarrow v$, if there exists some production $\alpha \rightarrow \beta \in R$ and $u_1, u_2 \in (N \cup T)^*$, such that $u = u_1 \alpha u_2$ and $v = u_1 \beta u_2$. Symbol $\Rightarrow^*$ is used to denote subsequent derivations.

Finally, the Language $\mathcal{L}$ accepted by a grammar $\mathcal{G}$ is defined as all words that could be reached by applying repeatedly its productions, that is $\mathcal{L}(\mathcal{G}) = \{w \in T^* : S \Rightarrow^* w\}$.

1.2.2 Featured Grammars

Featured Grammars represent an extension of the formal mechanism that allows constituents to have features. Features handle agreement restrictions in the parsing. For example, the noun phrase *a men* is incorrect in English because the number agreement restriction is not satisfied. In this case, the production $NP \rightarrow Art\ Noun$ (a *noun phrase* is an *article* followed by a *noun*), could be extended with the agreement restriction "number of Art must agree with number of $Noun$".

According to this, agreement restrictions are described with logical predicates on feature structures. A feature structure is a mapping from features, such as *number* in the example, to values, such as *plural* or *singular.* This example and a further discussion on Features and Augmented Grammars could be found in [All95, Chapter 4].

We will use $X \rightarrow_{[C]} x$ to denote the agreement restriction C in the production $X \rightarrow x$.

1.2.3 Semantic Grammars

Semantic grammars are formal grammars whose non-terminal symbols are representative of domain aspects. Note that this nuance refers only to an interpretation of the grammar from a semantic point of view, but does not entail specific formal restrictions. As stated in [GAMP87], since the grammar reflects the conceptual structure of the KB, parsing and interpretations become simpler.

1.3 Domain Ontologies

Common characteristics of the current Web include the use of huge quantities of different multimedia resources (text, images, video, or so on) that could additionally be distributed among various computers. These resources should be univocally identified before being remotely accessed. Traditionally, the Web was syntactically defined, delimiting the sort of possible queries to be performed to syntactical ones. The Ontologies emerged as a means to reach a shared semantic representation of the resources from the so-called *Semantic Web* perspective. The goal is to support automatic management of data and to favour reusability of components.

An ontology is a formal conceptualization of the knowledge and relations in a given domain. This conceptualization supports reasoning through formal logics. Ontologies provide a shared representation of the domain *vocabulary*, including *objects*, *properties*, and *relations.* Those concepts are usually hierarchically defined by means of taxonomies of classes. Most common components in an ontology also include individuals, axioms, or formal restrictions. Usually, term *domain-ontology* is preferred to emphasize the dependence on the domain. That is, an ontology models the *part of the world* regarding a specific domain. Thus, among the various meanings of a given term, only that related to the model is specified in the ontology.

Ontologies are usually described in OWL[1] (Ontology Web Language), a language based on DAML-OIL which, in turn, is the combination of the DAML (DARPA Agent Markip Language) project with the OIL (Ontology Inference Layer). This language is based on SPARQL triples. The following is just part of an ontology described in OWL about academic computer science community[2].

```
<owl:ObjectProperty rdf:ID="has-author">
<rdfs:domain>
<owl:Class>
<owl:unionOf rdf:parseType="Collection">
<owl:Class rdf:about="#Information-Bearing-Object"/>
<owl:Class rdf:about="#Publication-Reference"/>
<owl:Class rdf:about="#Technology"/>
<owl:Class rdf:about="#Method"/>
</owl:unionOf>
</owl:Class>
</rdfs:domain>
<rdfs:range rdf:resource="#Generic-Agent"/>
<rdfs:isDefinedBy rdf:resource="&base;"/>
</owl:ObjectProperty>
```

Protégé[3] us unarguably one of the most popular ontology editors. In this work, we used Jena[4], a Java API for handling OWL ontologies, in our implementations.

[1] http://www.w3.org/2004/OWL/

[2] http://www.daml.org/ontologies/322

[3] http://protege.stanford.edu/

[4] http://jena.apache.org/

2 HITS: A Grammar-based Framework for Natural Language Interfaces

2.1 Introduction

Natural Language Interfaces (NLI) allow non-technical users to access information stored in Knowledge Bases (KB) in a familiar manner. Alternatives such as menu-based interfaces [BKK05, TPT05], graphical interfaces [Spo93], keyword-based systems [KBF07], or free-input queries [KBZ06], help to overcome the complexities of formal query languages while freeing users from being aware of the particular knowledge model. The increasing number of potential applications made it appealing the necessity of developing rich NLI systems, a field that received a big deal of attention, especially since the mid 1980's.

Although earlier NLI focused mainly on improving the performance for specific Databases (NLIDB), adapting the system to new domains usually entailed much effort [GAMP87]. Therefore, this tendency changed gradually in favour of designing more *easily portable* NLI systems. *Portable* or *Transportable* systems are those that can be adapted easily to new domains (see the TEAM system, [GAMP87]). Loosely speaking, the basic idea beyond portability consists of separating the domain conceptualizations from the lexical model. To this aim, ontologies[5] and lexicalizations [MSC11] have proven to be useful tools to represent the semantic and lexical model, respectively. As stated by [SJ05], one of the main issues to deal with in NLI concerns with how to *bridge the gap* between user's expectations on the model and the actual knowledge base (see TEAM, or PANTO [WXZY07], for examples of portable NLI explicitly motivated by this issue) and "to trap user's unwitting excursions outside the database boundaries" (see [Hal06] for an example). In this regard, successfully mapping linguistic expressions to domain concepts and relations becomes paramount. These problems are usually faced during the configuration stage of the system, which in turn means designing a suited meta-knowledge resource.

In this section, we present the HITS (Human-computer Interactions for Transportable Systems), an entire framework for NLI focusing on the notion of Query Models (QM), a mechanism to encapsulate and delimit abstract query types and resolutions that makes the system become portable. We will devote this section to present the main components and mechanisms from a functional point of view. The next section focuses on customization, portability, and reasoning.

Our framework relies on the approximation of *tractable* questions as the *language* accepted by a formal semantic grammar (section 1.2). A question is said to be *tractable* if it could be understood by the system to some extent, which in turn means an interpretation could be assigned to it. Usually, interpretations on structured resources are naturally equated to parse-trees obtained by a suited formal grammar. In this regard, semantic parsings compound interpretations that constitute bridges between NL and formal conceptualizations. Since inner non-terminal symbols in semantic grammars represent domain concepts, these kind of grammars are inherently tailored to the domain. Thus, systems relying on grammar parsings do often prefer syntactic grammars (section 1.2). In this regard and according to [UHC10], although our grammar will be instantiated to a particular domain, the mechanism that generates it is completely independent of the underlying ontology.

The rest of this section is structured as follows. An overview of previous work on NLI and grammar parsing is presented in section 2.2. In section 2.3, an overview of our framework and its components is described. Finally, section 2.4 offers a discussion on the experimental validation of our study, and section 2.5 concludes with a discussion of results and future research.

[5] http://www.w3.org/standards/semanticweb/ontology

2.2 Related Work

In this section, the main related work to our approach are discussed. First, a brief introduction of earlier NLI is depicted. Later, we review the state-of-the-art focusing on performance, habitability, and usability.

2.2.1 Natural Language Interfaces to DataBases

The very first attempts in NLI where designed to deal with DataBases (NLIDB) and appeared in the late sixties. One of the earlier systems in this line was LUNAR [WKW72], first seen in the Second Annual Lunar Science Conference. LUNAR answered questions about chemical analyses of rock samples brought back by Apollo 11 from the Moon. It was specifically designed to deal with a particular domain and it could not be easily ported to other domains. LIFER/LADDER parser [HSSS78] was proposed in 1978 as an interface to the US Navy ships domain. It was based on Woods's Augmented Transition Networks (ATN) formalism to parse questions in a distributed database. Other example of system using ATN is [Wal75], which later derived in PLANES [Wal78]. PLANES is a NLIDB that answers questions about maintenance on flight stories of airplanes. Further examples among early NLIDBs include [Cod74, BB75, TT75, Bur76, Har77]. To access a vaster introduction to early NLIDBs techniques and methodologies, [ART95] is highly recommended. A more recent review of NLIDBs could be found on [PRGBAL$^+$13].

2.2.2 Performance, Habitability, and Usability

Earlier systems in the line of high-precision preservation were done in the field of restricted sublanguages [Kit82b], or adding semantic constraints to a specific grammar [Kit82a]. Although firsts approaches relied on hand-crafted domain specific lexicons, more recent work have focused on detecting automatically those restrictions [Sat97]. PRECISE [PEK03] is a domain independent graph-based approach that takes advantage of some constraints to ensure high-precision performance. It relies on the concept of *tractability* and *intractability*. PRECISE preserves precision at the expense of recall. That is, every time the system answers a question, the interpretation is likely to be correct. As a counterpart, the language coverage is restricted to unambiguous tokens explicitly appearing in the database. Other methods such as [Hal06, HPS07, BKK05] were aimed to override the effort that *free text queries* may cause to the interpretation process. These systems use a menu-based interface to assist the user in the composition of (only) tractable queries. Questions were managed by means of the so-called *Query Frames* —frequent query patterns. In [Hal06], logical representations were governed by a predefined ontology. The underlying idea beyond our *Query Models* is very close in essence to that of *Query Frames*. The main difference is that our patterns are learned from examples. Also, *Query Models* are meant to deal with free-text inputs. Moreover, *query patterns* in [HPS07] are manually constructed trying to generalize a given set of expert questions. The author stated that after some training, users were able to pose fluent and complex queries that the system was able to answer correctly. Thus, even if the system was promising in terms of *usability* —how ease-to-use the system is—, its main drawback concerned with the concept of *habitability* (see [Wat68, HLP97]). Habitability concerns with how well a system supports the language people could use to interact with it [OB96, OMBC06].

In order to override the limitations imposed by the knowledge that explicitly appears in the KB, the so-called FrameMapper were added in ORAKEL [CHH07] as a mechanism to learn new words and concepts from user interaction. However, this system is limited to factoid questions, starting with *wh*-pronouns. Also in this line, we intend to equip our method with language acquisition techniques allowing the system to incrementally learn new sort of questions. This could be done during execution time, as proposed in other approaches such as ASK [TT85] or PARLANCE [Bat89]. Similarly to ORAKEL, our grammar is strongly tied to the domain, but it is automatically constructed through a fully domain-independent procedure. In our case, the grammar is exploited to generate hypotheses and deductions to acquire new knowledge. A second case of limitation is due to the query language used. For example, methods relying on SPARQL can usually not tackle "how-many" questions [WXZY07]. For this reason, we decided to use SQL as the underlying query language.

In QTAL [FKX+07], the interpretation of the question is conducted by means of a Robust Minimal Recursion Semantics (RMRS) model, that implements the mapping to FrameNet query types. The most relevant aspect to our approach is the definition of the so-called *proto-queries*. We will also use abstract query patterns that are quite similar in essence to the concept of proto-queries of QTAL or the Query Models as defined in CLEF [Hal06]. As will be seen, one advantage of our model is that those patterns could be learned by examples. As a result, the time needed to specify mappings from lexical concepts to domain concepts is reduced.

As stated by [DAC10]: "NLI with a good performance require some customization". In this regard, PANTO is a system dealing with the trade-off between *portability*, and the effort required to bridge the gap between *real-world users* and the *logic-based semantic web*, as defined in [BKGK05]. Authors clearly differentiate the (mandatory) General Dictionaries, automatically filled and enhance with thesauri such as WordNet, from the (optional) User-Defined Synonymous. In the opposite side, there are some approaches that does not rely on any sort of customization. This is the case of NLP-Reduce [KBF07], Querix [KBZ06], or AquaLog [LPM05]. Since NLP-Reduce does not use any complex NLP technique, it is completely portable across domains. This system tries to match the user free text query (including keywords-based queries or full NL queries) against the *subject-property-object triples* extracted from the KB. Querix operates in a similar way. It deals with the problem of ambiguities in full NL queries by asking the user for clarifications. In this line, FREyA [DAC12] engages the user with disambiguation dialogues if necessary, but in a different manner: FREyA is able to learn from user's choices in order to improve its Recall over time. AquaLog [LPM05] also learns from user disambiguations, and also considers the context of application in order to resolve future ambiguities. Our system uses similar clarification dialogues to solve certain ambiguities and to acquire new knowledge based on accepting or refusing hypotheses. In contrast to FREyA and according to ORAKEL, we will only rely on the expert criteria to introduce new knowledge on the system. We use similar techniques to that used on FREyA, in order to generate and rank suggestions. However, instead of applying *string similarity* techniques, we search beyond the domain boundaries performing IR techniques on the Web. In contrast to the AquaLog's learning mechanism, ours allows the system not only to learn new word or phrases, the mapping to corresponding concepts/relations, and the context of application, but also new sort of questions. The architecture of HITS is more closely related to that of PANTO. We also distinguish between mandatory and optional linguistic resources. The main difference with respect to PANTO is that optional synonymous could be learned by examples and disambiguation dialogues in our system. Furthermore, our system is not limited by the sort of questions imposed by the underlying triple-based analysis.

2.3 Method Overview

This section offers an overview of HITS. Bearing in mind that the system evolves with the use, this section should be regarded as a snapshot of the system at a given moment. A discussion about learning and how the system improves by examples will be offered in the following chapter. Main components described in this section are briefly depicted in Figure v.1, which might be useful to guide the reading.

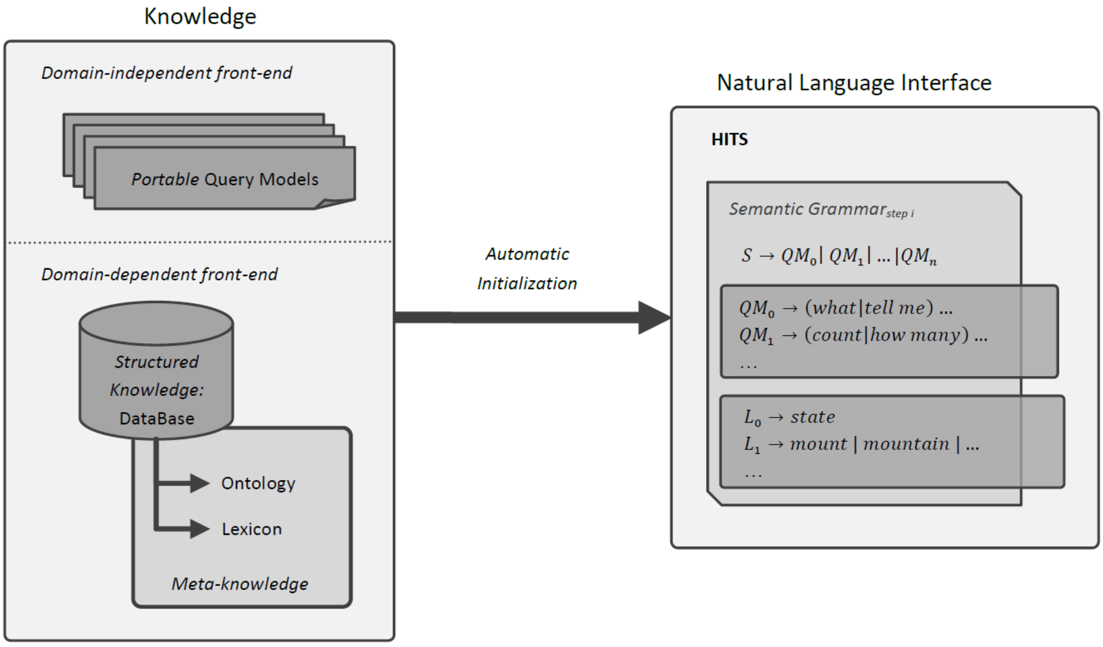

Knowledge
Domain-independent front-end
Portable Query Models
Domain-dependent front-end
Structured Knowledge: DataBase
Ontology
Lexicon
Meta-knowledge
Automatic Initialization
Natural Language Interface
HITS
Semantic Grammar$_{step\ i}$
$S \rightarrow QM_0 | QM_1 | \ldots | QM_n$
$QM_0 \rightarrow (what|tell\ me) \ldots$
$QM_1 \rightarrow (count|how\ many) \ldots$
...
$L_0 \rightarrow state$
$L_1 \rightarrow mount\ |\ mountain\ | \ldots$
...

Figure v.1: System Overview Diagram

The reader should be aware that most of the examples along this chapter are based on the datasets from Mooney's research [TM01]. The Entity-Relationship diagrams of these domains could be found in 4.

2.3.1 Components of the System

This section describes the main components of HITS (see Figure v.1). On the one side, the structured knowledge consists mainly of a DataBase containing the data instances (section 2.3.1), the meta-knowledge is mainly composed by an ontology describing the conceptual model (section 2.3.1) and a lexicon describing the domain-dependant linguistic knowledge (section 2.3.1). There is an additional knowledge resource called the *Base of Questions* —a compilation of typical questions on the domain (section 3.3.1). Since this resource is required for the learning algorithm, it will be rather described in the following chapter. On the other side, the general knowledge consists of a set of previously defined Query Models —portable abstractions of query mechanisms, that will be explained in section 2.3.2. Finally, the system counts with a semantic grammar (section 1.2) that is automatically instantiated from the knowledge and used for interpreting questions. This grammar is iteratively improved by examples, but this explanation will remain for the next chapter.

The DataBase In this chapter, the structured KB will be considered to be a relational DataBase (DB) [PC08]. The database contains all factual knowledge in the domain. It is organized as a set of tables, relational tables, features (columns), values (rows), and metadata such as primary keys, foreign keys, or so on. Databases are queried by means of DataBase Management System (DBMS). In this work, we will consider the SQL query language[6].

Ontology The ontology represents the semantic conceptualization of the KB that defines the elements and their relations. Although the ontology could be manually defined by the expert, it could rather be automatically obtained from the DB using techniques such as [GC07] or [CGY07]. In this work, we have automatically obtained the ontologies by querying the metadata tables in our DBMS. This procedure is explained in 5.

According to usual concepts in the formalization of a database, we organize the elements in the following categories:

Entities: Represent any concept of the real world. Each entity corresponds to a table in the DB —relational tables are not considered as entities. State and River are some examples of entities in the Geobase domain.

Attributes: Represent any characteristic of an entity. An attribute corresponds to the concept of feature in a table. Information about its parent entity or its datatype (numeric, string, boolean, etc.) are also available in the ontology. Hight is an example of a *numerical* attribute of the entity Mountain in Geobase domain.

Values: Refer to the value taken by an element for a particular attribute in a table. Recall that values do only exist as concepts in the ontology. Thus, the actual data values are not stored in the ontology.

Relations: Correspond to each relational table in the DB. Also information about the origin and destination entities are stored. Through is an example of relation between entities River (origin) and State (destination). The inverse relation order is also represented in the ontology, for example inverse Through relates State to River in the passive-form.

Operations: Represent certain calculations in a formal query system. Examples of operations are OrderBy, Greatest, GreaterThan, Not, etc. Each operation is linked to a procedure that calculates it. Some operations are implicitly linked to numerical attributes. For example, numerical attributes Length or Age do implicitly count with the Greatest operation, that represents *the largest* and *the oldest* calculations, respectively.

Virtual Attributes: define a certain sort of domain-dependent calculations —in contrast to *virtual relations* as defined in [GAMP87] that define join procedures between entities. Virtual Attributes (VA) are calculated taking other real attributes as parameters. For example, one could define PopulationDensity in Geobase domain as Population / Area. Note that VA are not calculated in advance, and results are not stored in the model. Rather, every time the system reaches an interpretation involving a VA, this VA is computed on demand. Defining VA is not mandatory, but contributes to improve the domain coverage.

[6] Our method is not restricted to any particular DB or DBMS. In any case, it may be worth mentioning that it was validated on MySQL

Lexicon According to the *lemon* perspective [MSC11, BCM$^+$11], suitable lexical resources linked to ontology semantic representations are required for NLP applications. Lexicalizations were proven useful in different NL technologies [MCZ12, ES06b], and how to construct them automatically [ALZ04] or semiautomatically [LMS02] gained increasing interest. The Lexicon is here considered to be a repository of all the linguistic links to each element in the conceptual model explained by the ontology. We will denote it as $L = \{(L_i, l_i)\}$ where L_i is the label of a lexicon entry associated to an ontology element, and l_i the list of its lexicalizations, or a regular expression. Although the lexicon is undeniably domain-dependant, some heuristics could be considered to reduce the effort required while creating it. Even if the lexicon is generated by the user, some heuristics allow it to be constructed through interactions. In this regard, the knowledge stored in the lexicon could be categorized in the following levels, depending on the effort required to obtain it:

Automatic: All linguistic values are automatically mapped to their content types in the ontology (for example, 'Texas' is automatically attached to value V_STATE_NAME that relates entity STATE and attribute NAME). Also names (labels) of entities, features, and relations are directly stored in the lexicon. However, some labels may be affected by prefix notation or abbreviations that may not be representative enough (for example DES_EXP referring to *desired experience* in jobdomain). Thus, not all knowledge could be filled in using only automatic procedures.

Semiautomatic: Most of the linguistic knowledge needed to adapt the system is acquired semiautomatically. Hypotheses and deductions provide the expert with possible linguistic rephrases of existing concepts, reducing the task of customization to that of validating candidates. Additionally, the system refines its performance by learning examples. In this way, the system is able to learn new sort of questions with the use. Also, WordNet is used as a tool to deploy synonymous of previous terms. According to [PPG$^+$05], prepositions play a key role in the interpretation of queries. For this reason, prepositions, such as "of", "at", "on", or so on, are presented as candidates for the relations. For example, *of* or *in* could be mapped to relations THROUGH, CITYINSTATE, or PASSES in Geobase.

Manual: According to [WXZY07], users may employ colloquial terms, jargons, or abbreviations while referring to certain entities. In this regard, we rely on user-defined synonymous as defined in PANTO. Manually modifying the lexicon entails effort, but it also allows directly improving the quality of the system [CHH07]. For this reason, the expert is allowed to provide rephrases to concepts at any step of the customization process. For example, the expert could manually correct the lexicalization of some labels, such as DES_EXP or passive-form relations, or could simply supply some synonyms for already existing concepts.

Reusable: General purpose concepts, such as numeric operations (maximum, minimum, average, ...), formatted fields (numbers, dates, passports numbers, ...), or query components (*group by*, *list all*, *how many*, ...) may be useful across many domains. Those elements are defined only once, and reused in new domains. Formatted fields are rather defined by means of regular expressions.

The Learner The learner is the system in charge of interpreting NL questions and learning from them when no interpretation is reachable. It retrieves information form the KB if the question is understood, or demands explanations to improve its performance otherwise. The learner uses and refines two main subcomponents (that will be later explained in detail):

QueryModels abstract query types and their resolutions mechanisms (section 2.3.2). We differentiate portable QMs and domain-dependent ones. Portable QMs abstract resolution mechanisms that may be useful across domains, while domain-dependent ones abstract resolutions schemas that are tied to the domain.

The Grammar formally defines the language coverage of the system, that is, the extent of *tractable* questions (section 3.4), and allows interpretations by means of parse trees. The grammar will be denoted as $\mathcal{G}$, and the set of tractable questions as $\mathcal{L}(\mathcal{G})$ (see section 1.2 for further details).

Once the ontological model and the initial lexicon are available, the grammar is automatically instantiated as a combination of portable QMs and the lexicon (see Figure v.1). Thus, notice that the system is started automatically. In any case, according to [DAC10], the answering ability of the system should be improved by means of some customization.

2.3.2 Bridging the Gap: the Query Models

So far, it has been shown how the Ontology represents the information of the system, and how the linguistic knowledge is stored in the Lexicon. Query Models (QM) abstract query mechanisms in an attempt to *bridge the gap* between formal knowledge and NL queries. Intuitively, each QM defines a particular map between NL questions and a resolution. As will be seen, a QM also includes the necessary conditions to ensure correctness in the resolution.

The following example may serve to illustrate the idea beyond the QMs. Let us consider the class of all NL questions requesting for an attribute of an element, that we will call AofV (as an acronym of *attribute of an entity named through a value*). An example of NL query belonging to this class could be "What is the capital of Texas?", taken from Geobase domain. This particular question could be resolved through the following SQL procedure:

SELECT Capital FROM State WHERE Name=‘Texas’

However, our intention is to be able to solve different instances of this class at once, such as also "Tell me the height of the Mount Whitney". This could be achieved by abstracting the SQL procedure (i) and by imposing some logical constraints (ii):

i) SELECT Attribute$_0$ FROM *fromEntity*(Attribute$_0$) WHERE *fromAttribute*(Value$_0$)=’Value$_0$’

ii) assert: *fromEntity*(Attribute$_0$) = *fromEntity*(Value$_0$)

Where Attribute$_0$ and Value$_0$ are variables containing the linguistic instances ‘height’ and ‘Mount Whitney’ respectively (analogously, ‘capital’ and ‘Texas’ in the previous example). Indexes identify univocally each sort of variable, and *fromEntity*($\cdot$) and *fromAttribute*($\cdot$) are the functions that obtain the associated entity-name and attribute-name respectively from the ontological model. We will call *Slots* and *Inspectors* to these variables and these functions respectively. Other examples of *Inspectors* are *getOriginOfRelation(·)*, *getDestinationOfRelation(·)*, and so on.

Now, the problem could be stated as determining whether a given question belongs to this class. Before addressing this issue, we will consider the set of *all* the possible questions for a given domain. As this set is unknown, we will attempt to approximate it from examples. To this aim, we propose to induce a formal grammar explaining a finite set of sample queries of the domain (section 3.4.1). In this regard, the set of *tractable* questions is formally defined as the language accepted by this grammar. Particularly, we will focus on *semantic* grammars (section 1.2). In contrast to syntax-grammars, non-leaf nodes of semantic grammars relate to semantic elements in the context of the application. In our case, we will take advantage of this feature by relating the non-leaf nodes to Query Models and *Slots*. The benefit is two-fold:

1. Deciding whether a question belongs to a QM is reduced to check whether the question is *derived* by its corresponding non-leaf variable.

2. Since *Slots* discern and encapsulate all the domain-dependant terms in the lexicon, productions associated to QMs become portable.

As will be seen, the semantic structure of the grammar is exploited by the learner to generate the interpretations, and to generate conjectures to acquire new knowledge by analogy. Picking up our previous example, the following could be some productions of the grammar:

AofV → (tell me | (what | who) [is]) [the] Attribute [of] [the] Value
Attribute → capital | height | ...
Value → Texas | Mount Whitney | ...

More formally, a Query Model is defined as a quadruple (I, P, C, S) where I is an identifier of the query model, P is a pattern describing a set of questions, C is a set of constrains, and S is an abstract resolution mechanism. Figure v.2 shows examples of QMs corresponding to AofV and ErelatedToV[7]. To improve the readability of the example, patterns shown are quite simple.

I:	AofV
P:	[tell me] (what | who | which) [is] [the] Attribute [of] [the] Value
C:	*fromEntity*(Attribute$_0$) = *fromEntity*(Value$_0$)
S:	SELECT Attribute$_0$ FROM *fromEntity*(Attribute$_0$) WHERE *fromAttribute*(Value$_0$)='Value$_0$'
I:	ErelatedToV
P:	(show | tell) me [the] Entity [that] Relation [the] Value
C:	Entity$_0$=*getOriginOfRelation*(Relation$_0$) & *getDestinationOfRelation*(Relation$_0$)=*fromEntity*(Value$_0$)
S:	SELECT *preferredName*(Entity$_0$) FROM *joinTables*(Relation$_0$) WHERE *join*(Relation$_0$) AND *fromAttribute*(Value$_0$)='Value$_0$'

Figure v.2: Query Models AofV and ErelatedToV

Since *Slots* encapsulate the linguistic references to concepts in the KB, the Query Models become domain-independent. For this reason, some QMs could be reused across domains in favour of portability. For example, AofV and ErelatedToV may be useful in practically every domain. Depending on the structure of the KB, the constraints allow to discriminate automatically the correct QM —if any— to be instantiated. For example, let us consider an user poses the query "tell me who wrote *War and Peace*". Let us consider the following possible structures (Figure v.3) of a KB about books.

[7] ErelatedToV is the QM devoted to retrieve all elements related to another element specified by a value. For example "show all rivers in Alaska".

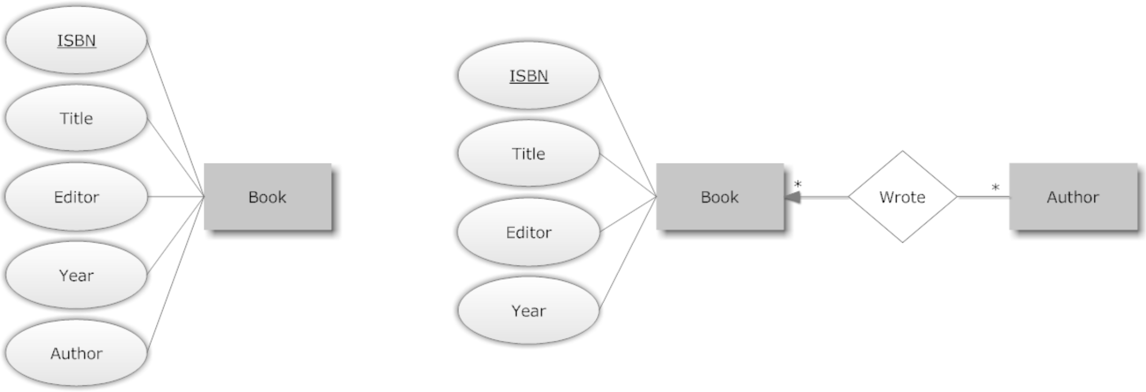


Figure v.3: Decision of the resolution based on the particular structure of the KB

Considering the Lexicon was previously filled[8], the query could be resolved through AofV in the first case, or through ErelatedToV in the second case. Semantic constrictions decide which of them automatically. This behaviour becomes more interesting as the number of involved QMs increases. For example, if the system was properly customized, it could be able to parse "List the titles of books written by Tolstoy" by combining AofV with EfeaturedV[9] in the first model, or AofV plus ErelatedToV in the second case. In both cases, the subquery "List the titles of <value>" is resolved through AofV. The only difference remains in the QM that will handle the subquery "...books written by Tolstoy", EfeaturedV or ErelatedToV respectively (Interpretations will be later explained in section 2.3.4). Note that semantic constrictions decide the resolution mechanism to be applied, keeping the user unaware of the particular structure of the KB.

The grammar, and therefore the productions that constitute each QM, are learned by examples semiautomatically. However, the constraints and the resolution mechanism should be manually validated (section 3.4.2). Although this entail certain effort, also present a main advantage: since the expert could add new QMs at any time, the system is not limited to specific question types. Section 3.5.2 offers a broader discussion on portability from the point of view of the Query Models.

2.3.3 Grammar Instantiation and Interpretations

The grammar is built from the query models, the ontology, and the lexicon. This grammar should not be regarded as a static resource, but rather as an initialization. Later, the grammar is iteratively refined by means of the learning algorithm and the interaction with the expert (section 3).

Formally, given a set of query models $QMs = \{(I_i, P_i, C_i, S_i), i = 1..n\}$, a set of ontology entries $O = \{o_i, i = 1..m\}$ that is partitioned in the categories defined by Categories= {Entity, Attribute, Relation, Operation, VirtualAttribute, Value}, and a set of lexicon entries $L = \{(L_i, l_i), i = 1..k\}$, where the identifier of each lexicon entry L_i corresponds to one ontology entry in O, we define our Semantic Featured Context-Free Grammar as $\mathcal{G} = (T, N, R, S)$, where:

[8] In the first case, the lexicon would contain entries such as (Book.Author, 'author') (Book.Author, 'write'). In the second case, the Lexicon would contain the entries (Author, 'author'), (Author, 'who'), (Wrote, 'write'). Recall the stemming process helps unify 'write', 'writes', 'wrote', ...

[9] EfeaturedV retrieves all elements containing an attribute with an specific value. For example "show me all the jobs recruited by Phil Smith"

T is a set of NL words (English in our experiments) and all terms l_i appearing in the lexicon

N $= \{S\} \cup \{I_i, i = 1..n\} \cup$ Categories $\cup \{L_i, i = 1..k\}$

R is the set of productions defined by Table V.1.

$S \rightarrow I_1 \vert I_2 \vert \cdots \vert I_n$	
$I_i \rightarrow_{[C_i]} P_i$	Featured productions of QMs
$\forall Cat \in$ Categories, $\forall o \in O$: $Cat \rightarrow o$ *iff* $o \in Cat$	Slots according to categories of the Ontology
$L_i \rightarrow l_i$	Derivation of lexical terms in Slots
Value$\rightarrow I_1 \vert I_2 \vert \cdots \vert I_n$	Recursion to parse compound queries

Table V.1: Instantiation of the Grammar

Figure v.4 exemplifies a fragment of a grammar for only three query models: AofV and ErelatedToV from previous examples; and EmostRelated[10] . The horizontal line discern the domain-independent productions (above) from the domain-dependent ones (down). It should be remarked that both types of productions are instantiated automatically (and refined semiautomatically later, see section 3.4).

	$S \rightarrow$ AofV | ErelatedToV | EmostRelated
	AofV $\rightarrow$ (tell me | what (is|are)) the Attribute of [the] Value
	[fromEntity(Attribute$_0$)=fromEntity(Value$_0$)]
Portable	ErelatedToV $\rightarrow$ (name|(give|show|tell)[me])[[all]the] Entity [which [are there]] Relation [the] Value
	[Entity$_0$=getOriginOfRelation(Relation$_0$) & getDestinationOfRelation(Relation$_0$)=fromEntity(Value$_0$)]
	EmostRelated $\rightarrow$ [tell me [the]] Entity [that | which] Relation [the] Operation [number of] Entity
	[Entity$_0$=getOriginOfRelation(Relation$_0$) & getDestinationOfRelation(Relation$_0$)=Entity$_1$]
	Operation $\rightarrow$ Greatest | Lowest | $\cdots$
	Attribute $\rightarrow$ StateCapital | StateName | $\cdots$
Domain dependent	Entity $\rightarrow$ State | River | $\cdots$
	State $\rightarrow$ state | states
	Greatest $\rightarrow$ (greatest|biggest|major|most|largest) [number of]
	$\cdots$
	Value $\rightarrow$ AofV | ErelatedToV | EmostRelated

Figure v.4: An example of Grammar

One may argue that claiming about portability through a semantic grammar is counter-intuitive [UHC10]. In this regard, it should be highlighted that in our semantic grammar there is a upper level corresponding to the Query Models. Because some Query Models are portable, these productions could also be considered to be portable. In any case, even if the grammar is tailored to domain-specific concepts, the procedure that generates it, and part of the knowledge to instantiate it, are actually portable [UHC10].

2.3.4 Interpretations and Resolutions

We will consider an interpretation of a tractable question $q \in T^*$ as a parsing tree by a given grammar $\mathcal{G}$. Because of the ambiguity of NL, more than one interpretation could be reached. In

[10] EmostRelated retrieves the element in an entity that is most related to another entity type. For example “which river traverses most states?”

that case, a disambiguation dialogue is generated. Nodes in the parsing tree corresponding to QMs indicate the resolution mechanism to be applied. This process may be clearer explained with an example. Let us consider $\mathcal{G}$ is the grammar from Figure v.4 and q is the following sentence: “What are the capitals of states bordering the state with most rivers?”

The parse tree of q by $\mathcal{G}$ —its interpretation— is depicted in Figure v.5. The SQL queries generated to solve this interpretation could be seen in Table V.2. Note that productions Value $\rightarrow I_i$ allow some compound queries to be parsed. Note also that the resolution mechanism of AofV is applied to each propagated value in the resolution of ErelatedToV.

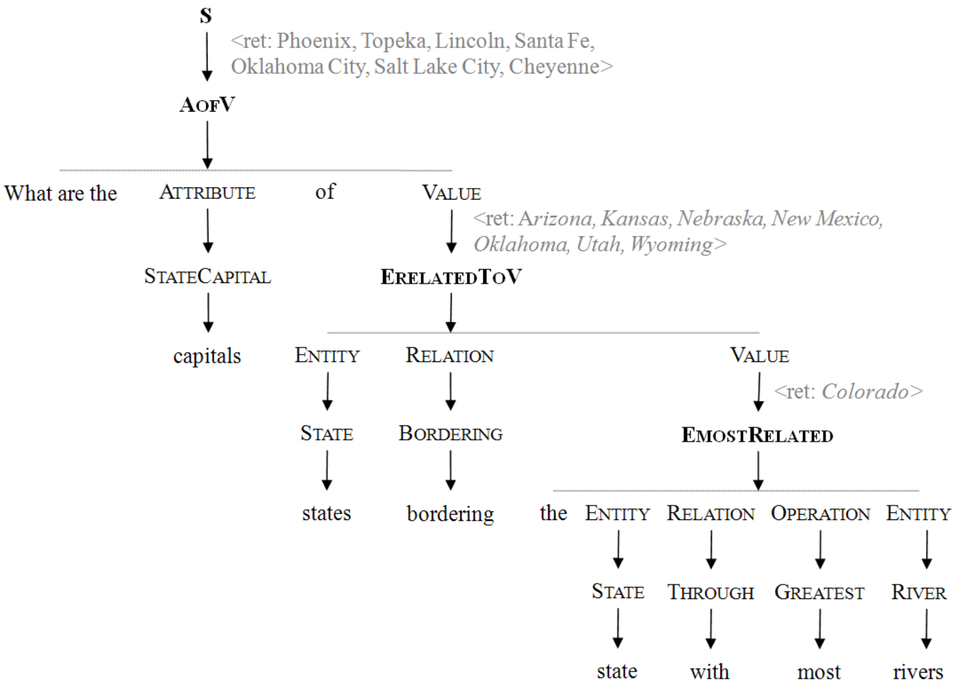


Figure v.5: Full parse of the sentence.

Query Model	SQL instance	Result
EmostRelated	SELECT through.state, COUNT(*) AS counting FROM through GROUP BY through.state ORDER BY counting DESC LIMIT 1	[colorado]
ErelatedToV	SELECT * FROM state, border WHERE state.name = border.state1 AND border.state2 = ’colorado’	[arizona, kansas, . . . , wyoming]
AofV	SELECT state.capital FROM state WHERE state.name=’arizona’	[phoenix]
	SELECT state.capital FROM state WHERE state.name=’kansas’	[topeka]
	. . .	. . .
	SELECT state.capital FROM state WHERE state.name=’wyoming’	[cheyenne]

Table V.2: SQL queries generated

As commented before, semantic conditions distinguish the correct interpretation to be considered when more than one is possible. In this example, *with* is a linguistic expression attached to various inverse relations. Thus, the parsing could also have been completed using other derivations such as Relation→CityInState→*with*. However, the semantic restrictions of EmostRelated assure that the entities involved —State and River in this case— are linked through the relation. In this case, the only relation satisfying this condition is Through (in passive-form).

In this study we deal with the Mooney Dataset [TM01], where most of the questions are factual. Thus, it is still up to future research investigating how to incorporate interpretations to Quantified Sentences [DCFSSV00] in a general manner.

Parse trees represent a reliable mechanism to define interpretations. As will be seen (section 2.4), each time an interpretation is reached, it is likely that its resolution is correct. In any case, the language coverage of the system is explicitly tied to the grammar. In this regard, agrammatical questions —such as keyword-based ones— may fall outside the language coverage. Trying to customize the language coverage to capture also these sort of questions may entail several effort. For this reason, in order to offer a better trade-off between *reliability* (precision), *customization effort*, and *habitability*, we propose an hybrid technique, combining deep analysis with *shallow* analysis[11]. The combination of these techniques have already proven to be useful to NLI in the QTAL system (see [FKX+07]).

Our shallow parsing approach is quite similar to that of NLP-Reduce [KBF07]. It consists of identifying named entities, features, relation, values, and negations. Later, the system tries to compose a formal query involving them as conditions in the Where clause. If any element is ambiguous (e.g. *Mississippi* may refer to the river, or to the state), then the system tries to disambiguate it by means of other related elements matched in the sentence. The main difference with NLP-Reduce is only on the query formalism —SPARQL in their case.

Shallow interpretations are undeniably less reliable than parse trees, but also more *habitable*. The system is aware of this and shows a warning dialogue before answering a question through a shallow interpretation. Table V.3 shows an example of the SQL query generated by the shallow parsing analysis for the question “Jobs for programmer in Austin has salary 50000 uses C++ not related with AI?”, from Jobdata domain. It is worth mentioning that the necessary information to automatically join tables is stored in the ontology. Note also that negations involving relations are resolved by combining sub-queries with the *NOT IN* operator. Finally, the answer is accompanied by a comprehensive relation of the elements recognized.

2.4 Performance Evaluation

Several experiments were performed in order to evaluate HITS and to compare it against other state-of-the-art NLI. We have evaluated our proposal on the Mooney's dataset (section 2.4.1) in terms of retrieval performance (section 2.4.2). What should be highlighted is that Query Models used in this part of the evaluation were manually designed. Thus, no learning, customization, or advanced grammar inference was involved in this experiment. Our goal is to test the performance of the framework —it goes for the next chapter to test portability in basis of learning by interaction.

2.4.1 The data

The system has been tested on three well-known datasets in the domain of *jobs*, *restaurant*, and *geography*, that were originally created by Ray Mooney and his researching group [TM01]. Those datasets have been broadly used to evaluate previous research in the NLI literature, including several

[11] *Shallow parsing* or *shallow analysis* usually stands for light parsers based on the identification of constituents, usually using named-entity techniques, and regardless of further syntactic or semantic considerations. They are considered to be less reliable but faster than other techniques such as formal grammar parsers.

```
SELECT *
FROM area,has_language,job,language,salary,has_area
WHERE job.id=has_area.job_id
    and has_area.area_id=area.id
    and job.id=has_language.job_id
    and has_language.language_id=language.id
    and job.id=salary.id
    and job.title='programmer'
    and job.city='austin'
    and salary.salary='50000'
    and language.name='cpp'
    and job.id NOT IN (
        SELECT has_area.job_id
        FROM has_area,area
        WHERE has_area.area_id=area.id
            and area.name='ai'
    )
```

-System: *I think I have not completely understood your question... but this information may be useful to you:*
<Recognized: 'Austin', 'salary=50000', 'not AI', 'programmer', 'C++'>
<SQL result>

Table V.3: SQL query computed with shallow parsing

recent ones. For the purpose, we converted the original Prolog databases into relational databases[12]. Table V.4 gathers some details about the structure of each dataset. Columns in Table V.4 represent the number of NL queries, number of entity-tables, number or relation-tables, number of attributes, and the number of registers, respectively. Further details could be found in 4.

	NL queries	**Tables**	**Relations**	**Attributes**	**Registers**
Jobdata	640	11	5	27	16548
Restbase	250	3	1	9	19345
Geobase	880	14	7	23	1652

Table V.4: Details of the Datasets

To evaluate the performance of our method, we have manually created the necessary knowledge resources. For the Geobase dataset, we took the same training (880 queries) and testing (162 queries) partition proposed by Mooney. Analogous partitions were not defined in the original datasets for Restbase and Jobdata. Thus, we randomly selected half of the questions (125 queries) from Restbase for training, and two thirds (480 queries) from Jobdata. The remaining questions (125 in Restbase, and 162 in Jobdata) were used as testing questions in their respective experiments.

[12] Available in `http://decsai.ugr.es/~moreo/publico/NLIDB_dataSets/Datasets_NLIDB.html`

2.4.2 Retrieval Performance

Also in this case, the retrieval performance will be evaluated in terms of standard precision and recall. Precision measures the number of correctly answered questions, divided by the total number of answered questions (Equation V.1). Recall measures the number of questions correctly answered, divided by the total number of test questions (Equation V.2). Finally, F-measure or F$_1$ (Equation V.3) represents the weighted harmonic mean of Precision and Recall.

$$Precision = \frac{\#\ of\ correct\ queries\ produced}{\#\ of\ successful\ parses} \tag{V.1}$$

$$Recall = \frac{\#\ of\ correct\ queries\ produced}{\#\ of\ sentences} \tag{V.2}$$

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{V.3}$$

We have contrasted the performance of HITS with the following models:

- Microsoft English Query[13] is a commercial application to allow developers create NLI by means of different programming languages.
- Cocktail is an inductive logic programming method that learns how to construct logical parse trees from training data [TM01].
- Precise relies on the notion of semantically *tractable* queries to ensure a precise mapping [PEK03].
- SILT learns deterministic rules to transform sentences or parse trees to meaning structures [KWM05].
- WASP is a system motivated by statistical machine translation techniques that learns a log-linear model to weight parses [WM06].
- KRISP constructs hierarchical representations from natural language sentences using SVM*struct* with string kernels [KM06].
- Lu's Model III+R uses dynamic programming techniques for training a learner that maps natural language sentences to hierarchical structures [LNLZ08] .
- FREyA is a system that takes advantage of the clarification dialogues to improve the retrieval performance [DAC10].
- PANTO adopts a triple-based data model to interpret the parse trees output by a parser, to generate SPARQL queries [WXZY07].
- NLP-Reduce is a system that does not require any sort of customization, and does not rely on any complex NLP technique [KBF07].
- Querix is an ontology-based question answering system which relies on clarification dialogues in case of ambiguities [KBZ06].

[13] http://msdn.microsoft.com/en-us/library/aa198281(v=sql.80).aspx

Unfortunately, not all authors agreed while measuring precision and recall. For example, Precise dealt only with tractable questions [PEK03], FREyA reported results after clarifying dialogues [DAC12], in PANTO duplicate queries were removed [WXZY07], and evaluation in Querix was performed with a preselection of 215 questions [KBZ06]. Thus, results reported here should not be regarded as a direct comparison, but rather as an indication on the retrieval performance of other models. The interested reader is referred to [LUSM11, pp. 139-142] for a more exhaustive comparison, and to [DB09] for a comprehensive explanation of the main problems regarding the evaluation of NLI.

Tables V.5, V.6, and V.7 show the results obtained by each algorithm in each domain. Scores of the compared methods were directly taken from the experimental results reported in their respective papers[14]. Unfortunately, experimental results in job and restaurant domains are not available for all the compared methods.

Jobdata	**Precision**	**Recall**	**F-measure**
EQ	75.0	47.5	58.16
Cocktail	93.25	79.84	86.03
Precise	**100**	87.5	**93.33**
NLP-Reduce	81.14	29.84	43.63
PANTO	86.12	89.17	87.61
HITS	95.7	**94.84**	**95.26**

Table V.5: Results in Job domain

Restbase	**Precision**	**Recall**	**F-measure**
EQ	52.5	37.5	43.75
Cocktail	97.5	**97.5**	**97.5**
Precise	**100**	95	**97.44**
NLP-Reduce	69.6	67.7	68.63
PANTO	90.87	96.64	93.66
HITS	**100**	**98.56**	**99.27**

Table V.6: Results in Restaurant domain

As can be observed, our model obtained comparable results regarding to state-of-the-art methods. Precise performs detecting whether a question is *semantically tractable* or not. If the question is semantically tractable, then Precise is able to answer it successfully, achieving a 100% precision score. Otherwise, Precise does not take risk, and a paraphrase is reported. The authors reported that, in Geobase domain, 22.5% of the questions were intractable and were removed before testing. Although the highlight of Precise consists of maintaining a 100% precision without the need for customizing the system, tractability restrictions diminish the *habitability* of the system. Albeit we cannot ensure 100% precision in all cases, the semantic conditions increase the reliability of the parses, allowing our system to reach 100% of precision in two domains. Our Precision error reported for Jobdata is due to the Shallow Parsing interpretations.

[14] Results for Microsoft English Query were taken from the experimental validation reported in [PEK03]

Geobase	Precision	Recall	F-measure
EQ	82.5	57.5	67.77
Cocktail	89.92	79.4	84.33
Precise	**100**	77.5	87.32
SILT	89.0	54.1	67.3
WASP	87.2	74.8	80.5
KRISP	93.3	71.7	81.1
Model III+R	89.3	81.5	85.2
FREyA	92.4	**92.4**	**92.4**
PANTO	88.05	**85.86**	86.94
NLP-Reduce	70.7	76.4	73.43
Querix	86.08	**87.11**	86.59
HITS	**100**	78.91	88.21

Table V.7: Results in Geographic domain

Recall that FREyA system could deliver some clarification dialogues to the user before answering. Although this policy is different than the one used in the remaining comparison algorithms, FREyA obtained the best results in Geobase. In this line, it is fair to mention that the evaluation of our method was also affected by one particularity: each time an unseen word appeared in a training question (only in Jobdata), it was added to the lexicon. Thus, this mechanism could be regarded as similar to that of FREyA's dialogues, but answered in advance.

Virtual Attributes allowed the system to deal with factual information that was not directly accessible in the database. We found that 33 of the queries in the Geobase domain involved information about *population density*. Although Virtual Attributes are defined by hand, they could effectively expand the language coverage of the system.

The complexity of the Geobase domain remains in the relationships among entities. Several relations and entities are involved in most of the questions. However, as the structure beyond some queries seems complex, most of them could be solved by addressing different subqueries recursively. In this domain, only 11 Query Models were employed. 65,52% of the cases could be solved directly with a Query Model. To solve the remaining 34,48% more than one QM was needed. Table V.8 shows the number of Query Models employed in each domain and the percentage of questions that were solved indirectly (involving more than one Query Model).

Domain	# Query Models	% Recursions
Jobdata	7	15.24
Restbase	10	19.12
Geobase	11	34.48

Table V.8: Query Models in each domain and question structure information

Maybe Restbase domain could be considered the simplest domain: it contains the lowest number of entities and relations. Furthermore, the underlying structure beyond the query samples is quite similar and the grammar was able to accept most of the questions. In contrast, although Jobdata domain also seems to be simple, the training questions use a long variety of syntactical structures. Moreover, there are several references to values that do not exist in the database. The shallow parsing heuristic played a key role in this domain. In our experiments, we found that 76.8% of the Jobdata queries could be resolved with the Shallow parsing heuristic.

2.5 Conclusions and Future Work

NLI represents a promising field aimed for overcoming the barrier that technical requirements impose to inexperienced users. In this section, we have introduced HITS, a framework for NLIs. The system is based on the notion of Query Model, an abstract representation mechanism that encapsulates resolution of queries. Since QMs are domain-independent, they could be reused across domains. QMs include mechanisms —the query pattern, semantic constrictions, and the abstract query resolution— that help to *bridge the gap* between NL queries and the particular KB.

The presented framework is able to separately manage different knowledge resources in order to bring domain-independence. However, this has not been tested in this section. Indeed, manually customizing the system to each domain entailed a considerable amount of work. Thus, how to alleviate as much as possible this effort becomes paramount in order to claim about portability. This issue will therefore be the main goal of our next section. Concretely, we will define a learning algorithm to iteratively refine the grammar by examples. The customization follows an interaction protocol whereby the system acts as a learner, and the expert in charge of the customization acts as a teacher. Thus, each time a non-tractable question is given to the system, it will try to reason by analogy on its previous knowledge in order to find a likely interpretation. If needed, the system could ask the teacher for a suited explanation. Thus, the customization process is no longer manual or based on the expert intuition, but rather directed by the system in a semisupervised perspective.

Even if the QMs were presented as independent resolution mechanisms, we have only validated here the case of SQL. It is still up to future research to show how effective this mechanism is while considering different querying languages such as SPARQL or SeRQL. Additionally, we are interested in studying the extent to which Query Models could be adapted to deal with vagueness, imprecision, and temporal queries in more sophisticated DataBase models such as Object-Oriented [MR01] or Fuzzy models [MRPCVMCT96, BPM$^+$11]. One of the most impressive abilities human beings present consists of the capability to reason in environments of imprecision and uncertainty. How humans reasons in a natural manner in such scenarios without performing measurements or computations is studied in the field of Computing With Words (CWW) [MZT$^+$10]. We are thus interested in approximating our Query Model representation in the direction of Fuzzy Logic [TRG06, TMG$^+$07] to allow the system deal with Fuzzy Databases. To this aim, how to properly adapt the current version of the Query Models to deal with Quantified Sentences [DCFSSV00] should be investigated.

3 Reasoning about the domain in Natural Language Interfaces

3.1 Introduction

Some approaches in the literature (i.e. NLP-Reduce [KBF07], or Querix [KBZ06]) can be considered fully portable in the sense that they do not require any sort of customization at all. However, as stated by [KB07], gaining in domain-independence usually incurs in lower retrieval performance. Moreover, the distrust created in the user when the system misinterprets his/her query, may cause a loss of *usability* [DB09], even if this rarely occurs. In this line, great research efforts have been dedicated to preserve high-precision. This is the case of PRECISE [PEK03], a system that reported 100% of precision by dealing only with *tractable* questions[15]. As a counterpart, restringing the language coverage causes a loss of *habitability* —how well a system supports the language people use to interact with it [OMBC06].

In order to improve both *usability* and *habitability* criteria on NLI, some customization is required [DAC10]. Consequently, how to alleviate the customization effort required while porting the system has attracted a big deal of attention (see the ORAKEL system, in [CHH07]). Also in this line, human-computer interaction has been proposed as an alternative to customize the system by means of learning new words or expressions through *disambiguation dialogues* (see the FREyA system, in [DAC12]).

To do this, we propose an inductive learning algorithm along the lines of Angluin's paradigm [Ang87]. Our system learns by examples a context-free semantic grammar from an underlying KB. According to our learning protocol (section 3.3.3), the *system* (aka the *learner* in the Angluin's notation) engages the *expert* (aka the *teacher* in the Angluin's notation) with *conjectures* about the language. As will be seen, interaction focuses on language acquisition in our case. Acting in this way, the customization process is directed by the system (learner), incurring in lower effort. This principle was already proven useful to improve the knowledge in FAQtory, the FAQ retrieval framework proposed in chapter IV section 3.

Above mentioned problems have motivated the approach presented in this section. The main contribution of this research consists of investigating how human-computer interactions could serve to language acquisition and portability in SK/PM. Interactions are mainly devoted to validate system's reasoning based on previous knowledge. In this section we incorporate some mechanism to allow HITS iteratively refine its retrieval performance by learning from examples (section 3.4). This approach bears strong resemblance to other methods coming from the Machine Learning (ML) community, such as [TM01, WM07]. The main difference lies on the nature of the input data. Mooney's approaches usually rely on labelled input sentences, according to typical supervised learning. We rely on a learning protocol according to which, only certain inputs should be labelled (explained) in a semi-supervised scenario. In addition, conjectures in our model aim to prevent the expert from explicitly explaining all the input sentences in the training stage.

Conjectures include hypotheses (*a priori* assumptions) and deductions (*a posteriori* assumptions). Although these issues will later be tackled in detail, a previous example here may be illustrative for the reader[16]. Let us consider that, after some point, one of the productions inferred by the *learner* looks like:

$$\text{QUERY} \rightarrow \text{(what is|tell me) the capital of Texas}$$

[15] A question is considered to be *tractable* in PRECISE if none of its tokens is ambiguous in the Knowledge Base.

[16] Just for the simplicity purpose, these examples are shown as a human-like dialogue. Formal representations will be later exposed.

Now, consider the following interaction between the learner and the teacher (expert):

-Teacher: learn that "what is the capital of Utah", is another valid QUERY

-Learner: [deduction] Then, is that possible that 'Texas' and 'Utah' are actually within a new sort of concept?

-Teacher: Yes! They are two examples of (the concept) STATE.

The above example is meant to illustrate how the learner deduces language structures by analogy. Let us now suppose the inferred grammar remains as follows

$$\text{QUERY} \rightarrow \text{(what is|tell me) the capital of STATE}$$
$$\text{STATE} \rightarrow \text{Texas} \mid \text{Utah}$$

And assume that the learner is trying to answer the question "Tell me the capital of Nevada". Note that the learner is not able to parse the question because of the bias imposed by tokens derived from STATE. Then the following dialogue occurs:

-Learner: [hypothesis] May the term 'Nevada' actually refers to the existing concept STATE?

The hypothesis produced by the learner states that, if the unknown word 'Nevada' is actually an state, then it would be able to interpret the question. After expert validation, unseen expressions are acquired and mapped to previous concepts. As will be seen, there are two main differences with respect to other related approaches such as the FREyA's acquisition language procedure [DAC10]. The first one is that our hypotheses are based on structural analogies, while FREyA is based on *string similarity*. The second one is that our method contrasts these inferences with information beyond the KB boundaries to improve the effectiveness of the validation dialogue. These issues will be later described in more detail.

The rest of this chapter is structured as follows. An overview of previous work on portability and customization is presented in section 3.2. We present the modifications needed to allow customization through interaction in section 3.3. Section 3.4 provides a detailed explanation of our learning algorithm and the language acquisition techniques. Finally, section 3.5 offers a discussion on the experimental validation focusing on portability and customization, and section 3.6 concludes with a discussion of results and future research.

3.2 Related Work

In this section, we briefly review the main related approaches in the literature. This review covers two main topics: portable systems (section 3.2.1), and grammar inference (section 3.2.2).

3.2.1 Towards Domain-independence: Ontologies and Portability

Since firsts NLI were often domain-tailored, they required considerable effort to be adapted from one domain to another. Thus, in the early 80s the research community get interested in designing more easily adaptable systems. Domain-independence was first understood as the capability of a system to be adapted without *code changes*. A vaster discussion of these approaches is available in [Kap84]. The main idea beyond this attempt consists of collecting all the domain-dependent knowledge in a lexicon, and differentiating it from a generic linguistic front-end. Although the lexicon must be rewritten or set up for other domains, the definition of front-end allows interpretations to be independent of the underlying database. Some examples of domain independent systems in this line include Loqui [BDSV91], SystemX [CMP+93], Masque/SQL [ART93], Clare [ACC+92], Tamic [BBMS96], CoBase [CYC+96], Edite [RMM97], InBase [BS03], Step [Min04], and Randezvous [Min07], and SQL-HAL[17]. Examples of commercial NLIDB systems could be found in BBN's Parlance [Bat89], IBM's Language Access [Ott92], Microsoft's English Query[18], Powerset[19], START[20], Wolfram Alpha[21], or True Knowledge[22].

However, as stated in [GAMP87], "adaptation to new domains required a sizeable new effort, almost equal in magnitude to the original one". Arguably, one of the most prominent contributions in this line was TEAM. The authors focused on *Transportability* to face the problem of bridging the gap between users' expectation about the domain and the way the knowledge is actually structured. In an attempt to alleviate the expert requirements and also the time required to adapt the system from one domain to another, TEAM provides the expert with simple mechanisms for acquiring domain-specific information. To achieve this, TEAM engages the expert with NL questions requesting for adding reformulations or synonyms. In this way, the expert was only expected to know about the database structure, but he/she was not meant to know how TEAM worked, neither being an expert in NLP. Thus, the problem of avoiding *code changes* is considered to be already solved in 1987. Notwithstanding, the truth is that the task of customization to adapt the linguistic front-end was still costly. This caused the reorientation of the problem in favour of *easy adaptability*, that is, minimizing as much as possible the configuration effort to port the system. To achieve this goal, layered architectures separating syntactic parsing from semantic interpretation and concrete resolutions were proved useful. In pursuing this purpose, the idea beyond *conceptual schemas* of TEAM evolved in the direction of *ontologies*. Furthermore, recent approaches such as PowerAqua [LNS+10] deals with various ontologies as knowledge resources at the same time. Our interaction protocol bears some resemblance to TEAM in the sense that it also engages the expert with NL dialogue. However, the main difference is that some of the questions HITS proposes come from the inference of analogies found on its previous knowledge. Thus, these questions are aimed to validate conjectures.

According to the classification proposed in [PRGBAL+13], HITS falls between *Pattern Matching*

[17] www.csse.monash.edu.au/hons/projects/2000/Supun.Ruwanpura/

[18] http://msdn.microsoft.com/en-us/library/aa198281(v=sql.80).aspx

[19] http://www.powerset.com/

[20] http://start.csail.mit.edu/

[21] http://www.wolframalpha.com/index.html

[22] http://www.trueknowledge.com/

Systems and *Semantic Grammar-based* ones. It could seem contradictory, however, claiming about domain-independence or portability taking these approaches as starting point. The main limitation in pattern matching approaches, is that the answering ability of these systems depends directly on the set of pattern rules added. However, as will be seen, our system learns and refines iteratively its patterns. Also, it could be argued that semantic grammars are undeniably tied to domain-dependent concepts. Recall however that the upper levels of our grammar are rather tied to the usual conceptualization of any ontology (the concepts of *entity*, *relation*, *value*, *attribute*...), and the procedure that instantiates the grammar, is independent of the domain.

3.2.2 Grammar Inference

Inductive Grammar Inference usually starts from sets of samples containing regularities to induce a grammar explaining the data. Learning the underlying structure of a set of samples is a well-known problem that has received a great deal of attention from a variety of fields as reflected in [AV02, Sak05, MOC$^+$08].

In order to infer semantic parses automatically, several machine learning techniques have been applied to given sentences paired with logical representations [TM02, LNLZ08, KZGS10]. The main goal of this approach, known as Inductive Logic Programming (ILP), is to avoid designing parses manually. Silt [KWM05] learns deterministic rules to transform sentences or their syntactic parse trees to meaningful structures. Wasp [WM06] is a system motivated by statistical machine translation techniques that learns a log-linear model to weight parses. Krisp [KM06] is a discriminative approach where meaning representation structures are hierarchically constructed from the natural language strings. [LNLZ08] proposed a system based on a generative model that builds hybrid trees instead of using explicit parsing grammars. More examples of ILP systems could be found in [GM06, WM07, ZC07].

Our approach is closely related to the work of [UHC10] based on Lexicalized Tree Adjoining Grammars (LTAG). Both methods consists of deriving a semantic grammar connecting lexicalizations with ontology conceptualizations to achieve a question answering system. Similarly, although the grammars inferred by both models are tailored to one specific domain, the mechanisms that obtain them are domain-independent. In contrast, our grammar inference criterion is based on the notion of *learnability* while the LTAG-based approach is motivated by the *composition* of logical structures. Moreover, we also exploit the inferred grammar in order to (semi-automatically) acquire new linguistic knowledge, reducing therefore the effort required in customization.

Our grammar learning algorithm is motivated by the Gold's learning paradigm [Gol67], but more concretely with the Angluin's paradigm [Ang87]. In Angluin's model, the *learner* aims to infer formal languages from queries samples by asking a *minimally adequate teacher* about *membership queries* ("does x belongs to L?") and *conjectures* (descriptions) on a regular set. The teacher answers *yes* if the conjecture represents a valid description of the unknown language, or a *counterexample* otherwise (or even a *correction* in [Kin08]). Similarly, our inductive learning algorithm engages the teacher (the person in charge of customization) with interaction dialogue.

Semantic grammars constitute a well-known mechanism in NLP to interpret sentences in delimited domains [Bur76]. Many variations and new approaches are continuously being developed [Aka97]. As constructing a grammar by hand requires much effort, how to automatically learn grammars from a set of linguistic samples has been widely studied. The most relevant approaches to our model include learning grammars from structured sentences [Sak90], and the Alignment Based Learning (ABL) algorithm [Zaa01]. Learning from structured sentences gets the grammar that generates a language $\mathcal{L}$ described by some sentences and their skeletal definitions, i.e. production trees with unlabelled inner nodes. The resulting grammar generates structurally equivalent sentences to the ones used for learning. One interesting point in this approach concerns with the structure of the input sentences. This reflects the natural idea whereby the structure of the examples should be coherent with the production rules of the grammar. Its main drawback however is that it needs a skeletal definition tree for each sentence in the training set. On the other side, the ABL approach only needs plain sentences as input. The structure is obtained iterating a twofold process. First, the alignment learning phase identifies partial alignments of pairs of sentences producing *hypotheses.* Zaanen explains that, even if there are several ways to align sentences, in the end the hypotheses should be identified as possible constituents linked to non-terminal symbols of the grammar. Secondly, the alignment selection phase resolves the overlapping that could have been produced between the hypotheses generated. Finally, the algorithm extracts the grammar from the alignments produced. Although this approach gets a structure of a corpus of sentences from a syntactic point of view, the obtained structure could not be considered semantically relevant to the domain. For example if the following set of questions is considered:

1. What is the [best Chinese food's]$_X$ restaurant in [Granada]$_Y$?
2. What is the [distance to the John's]$_X$ restaurant in [kilometres]$_Y$?

ABL should produce the hypotheses X, Y, following the alignments *What is the* and *restaurant in.* However, because different concepts are mixed, some hypotheses could not be considered representative to the domain from a semantic point of view. Our work is strongly motivated by similar principles. The main difference is that we combine the Harris' *sustitability* criterion [Har51] with expert validation in order to obtain grammar variables that are semantically representative of the KB.

In CQL/NL system [Owe00] a semantic grammar that is similar to slot grammars [McC90] is used. Those grammars rely on fewer and simpler rules containing slots that are only instantiated with semantic concepts derived from the particular domain. However, CQL/NL only uses the semantic parse trees as an intermediate step to fill those slots with actual data values. The grammar in HITS is quite similar. As shall be seen below, we use conceptualizations of the database as slots, and a set of samples to learn the semantic grammar.

3.3 Customization through Human-computer Interaction

According to the framework proposed in section 2, we show some modifications to allow the customization through human-computer interaction (see Figure v.6). Those modifications involve the addition of a new resource, the Base of Questions (section 3.3.1), the definition of the roles in the system (section 3.3.2), and the interaction protocol dictating the customization process (section 3.3.3).

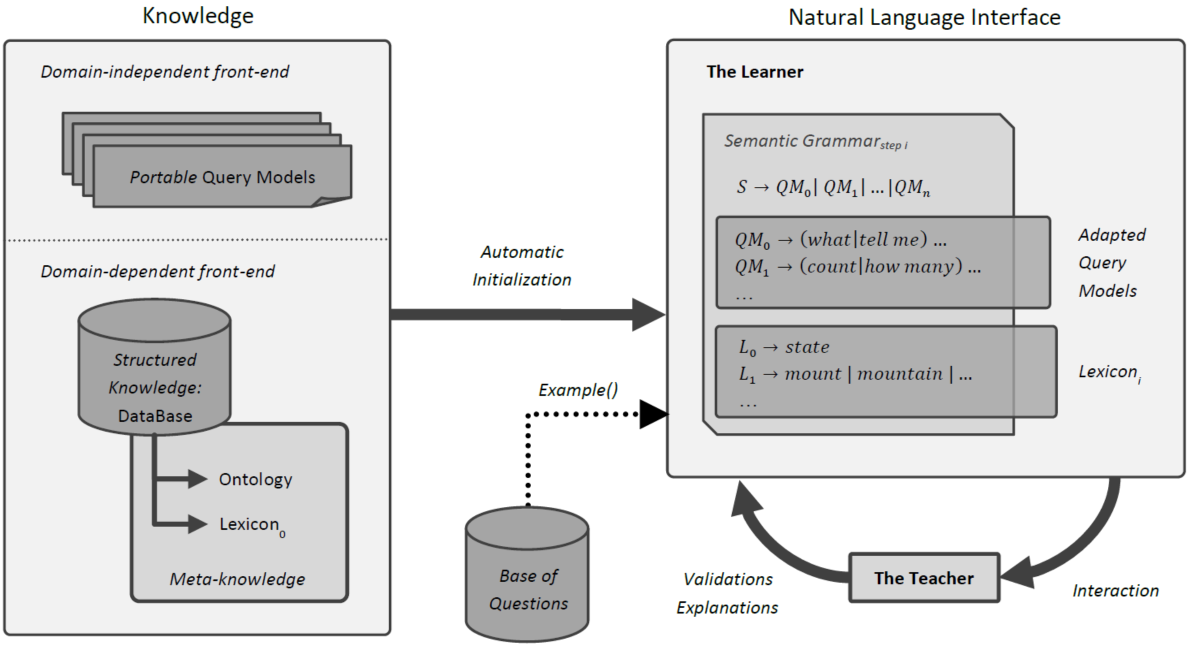

Knowledge
Domain-independent front-end
Portable Query Models
Domain-dependent front-end
Structured Knowledge: DataBase
Ontology
Lexicon$_0$
Meta-knowledge
Automatic Initialization
Example()
Base of Questions
Natural Language Interface
The Learner
Semantic Grammar$_{step\ i}$
$S \rightarrow QM_0 | QM_1 | \ldots | QM_n$
$QM_0 \rightarrow (what|tell\ me) \ldots$
$QM_1 \rightarrow (count|how\ many) \ldots$
...
Adapted Query Models
$L_0 \rightarrow state$
$L_1 \rightarrow mount \mid mountain \mid \ldots$
...
Lexicon$_i$
Validations Explanations
The Teacher
Interaction

Figure v.6: System Overview Diagram: Modifications to support customization through interaction

3.3.1 The Base of Questions

Represents a compilation of questions related to the domain. The following question is returned by the *Example()* procedure (as defined by Angluin). Following the idea that collecting questions is an easy task, these questions could be collected before customizing the system, and then used by the expert while debugging it. The learner will pick one question at a time, and will ask the teacher for an explanation if it cannot parse it. Furthermore, users questions that could not be parsed are automatically stored in the base of questions for future debugging. In this way, the system counts with a mechanism to improve its performance by use. Learning from iterative cycles based on the questions that the system previously failed was already proposed in some other approaches (see [CHH07]). Also, storing user's questions is considered to be a mechanism to overcome the bias imposed in the manner an expert tends to write.

3.3.2 Roles in the System

There are three kinds of roles in HITS: regular users, database expert(s), and the learner.

Regular User Regular users access information about the domain by means of NL questions. They are not expected to be aware of how information is structured, what are the boundaries of the domain, or what sort of questions they could pose.

Also, the system may engage the user with disambiguation and clarification dialogues. Every time the system reaches more than one possible interpretations of a given question, the user is requested to specify the correct one. For example:

- User: What is the population of New York?

- Learner: With 'New York' you refer to the CITY, or to the STATE?

Clarification dialogues aim to validate conjectures about unknown words or phrases. For example:

- User: Show me the jobs using JavaScript in Austin.

- Learner: Does 'JavaScript' refer to a LANGUAGE?

Note that the difference is that disambiguation dialogues are based on already stored knowledge, and clarification dialogues are based on conjectures about unknown words (as will be seen in section 3.4.3). Finally, as the user enters queries, the system collects them. Unparseable questions are later presented to the expert who is requested to offer an explanation —if any.

The Expert (Teacher) The expert is the person in charge of customizing the system. The role of the expert consists of supplying the necessary linguistic information to adapt the system to a new domain or to improve its language comprehension in a specific domain. To this purpose, we have adopted a teacher-learner model based on dialogue interaction. Interactions include: (i) debugging the system by querying it in NL and (ii) offering explanations if necessary; (iii) validating conjectures of the learner, and (iv) adding knowledge by hand (optional). The following are some examples illustrating these kinds of interactions, respectively:

(i) Teacher: How many rivers pass through Austin?

(ii) Teacher: Learn that 'pass through' is a rephrase of (relation) THROUGH.

(iii) Learner: Does term 'rivers' refers to the entity RIVER?

(iv) Teacher: Add rephrases to THROUGH: 'cross', 'traverse', 'in', and 'run through'

Actually, explanations (ii) may also include definitions of new sorts of questions (section 3.4.2). In order to reduce the expert effort while validating conjectures (iii), the system counts with various heuristics, that will be discussed in section 3.4.3. Finally, WordNet [MBF+90] is used in (iv) to alleviate the task of adding rephrases to existing terms. Notice that expertise in the formal query language, and knowledge of the structure of the KB, are demanded as the only requirements for the expert. He/she is not expected to be an specialist on linguistics or NLP either.

The Learner The learner is the Conversational Agent devoted to acquire the necessary linguistic knowledge to interpret queries. The term *learner* is here adopted for two main reasons: firstly, to preserve coherence with the notation of the Angluin's learning paradigm, and secondly, to empathise the fact that the system *studies* both its grammar and the Base of Questions to optimize the acquisition process. Furthermore, it retrieves information from the KB if the question is understood, or demand explanations to improve its performance otherwise.

The learner fulfils the following tasks:

Learning from examples and explanations offered by the teacher to acquire new knowledge and to refine its current linguistic abilities (section 3.4).

Interpreting users NL queries. An interpretation represents a mapping from NL expressions to conceptual elements in the KB. Interpretations are reached through grammar parsings (section 2.3.4).

Answering users NL queries. Once an interpretation is reached, the KB is formally queried and the information retrieved is finally offered to the user. This process is carried out by means of the Query Models resolutions (section 2.3.2).

Interaction The learner interacts with the user through disambiguation dialogues when various interpretations are reached. Also, the learner asks the teacher for explanations when no interpretation was reached. Furthermore, dialogues are also used to validate conjectures (section 3.4.3).

3.3.3 Interaction Protocol

To empathise the notion of learning, [Ang87] used *the teacher* to call the source of trusted knowledge, and *the learner* to call the learning algorithm. We will use the same notation here. Moreover, the author refers to the desirable features of *minimally adequate teacher* as his/her ability to present 'helpful examples' (regarding the convergence of the learning process) and to correctly answer the learner conjectures. Although it is not our intention to deal in deep with computational complexity issues in this chapter, it is worth mentioning that presenting 'helpful examples' will undeniably diminish the customization time. Rather, we are interested in language acquisition through human-computer interaction. In this section we describe the interaction protocol. Figure v.7 describes the three main interactions with the system.

Answering describes the main protocol whereby the system interprets a given question and answers it, or bifurcates in other protocols of action. The question corresponds to the next question in the Base of Questions or to an specific query posed by the teacher. The learner should decide whether the question is tractable, that is, whether $q \in \mathcal{L}(\mathcal{G})$. If so, we consider two possible scenarios depending on whether there are various possible interpretations, or only one. In the former case, the learner engages the teacher with a disambiguation dialogue. After disambiguating, the system performs the formal query and answers. There is a third case whereby $q \notin \mathcal{L}(\mathcal{G})$. In this case, process *Interpreting* calls protocol Assuming. Finally, if the interpretation is not correct, the teacher could offer an explanation.

Explaining describes the sequence of interactions whereby the learner acquires new knowledge explicitly added by the teacher (see section 3.4.3). After acquiring new information, the learner could formulate *Deductions*. In that case, the teacher is asked for validating them).

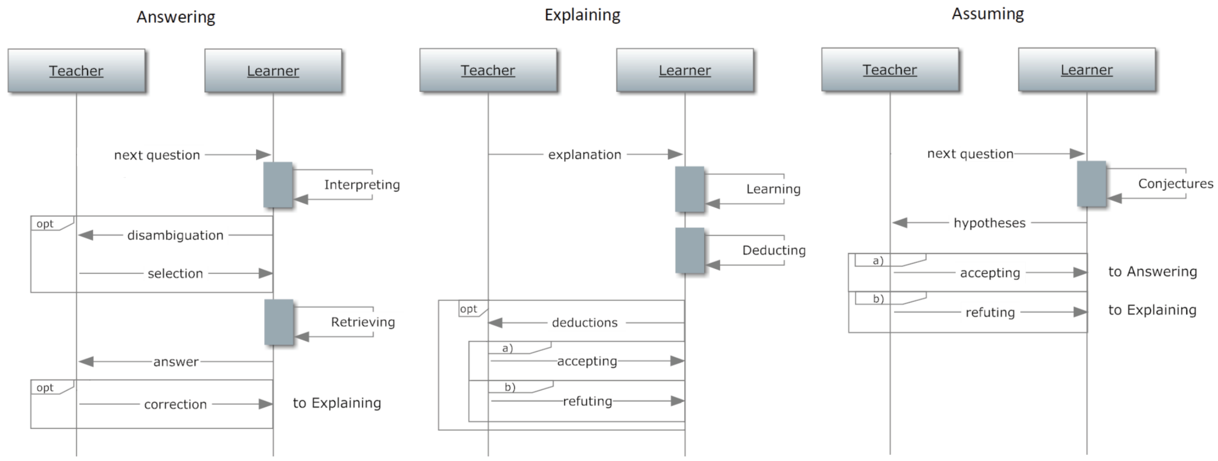


Figure v.7: Interaction Protocol

Assuming describes the process whereby the learner tries to reach an interpretation assuming that certain unknown words are actually related to the domain. Thus, the system is aware of the incompleteness of its own knowledge, and tries to fill this gap with conjectures (hypotheses in this case). Each time an hypothesis is accepted by the teacher, the learner learns from it. If no hypothesis is plausible, or all them were refused, then the learner asks the teacher for an explanation.

3.4 Learning: Language Acquisition, Hypotheses, and Deductions

So far, the entire framework for NLI system has been described, paying special attention to the interaction protocol, the query models, and the grammar. According to [GAMP87], providing a means for acquiring domain-specific information easily is crucial for transportable systems. Thus, the main goal is to enhance recall by learning from unseen terms appearing in a new question [DAC10].

In this section we describe how the system learns from examples refining its productions (section 3.4.1), creating new ones (section 3.4.2), or helping to customize the system in basis of conjectures about the language (section 3.4.3). The teacher plays a key role in the learning process, since he/she provides the system with the necessary knowledge to do this. Furthermore, it should be remarked that this process is semiautomatic, and is monitored by the system, reducing the effort required in customization.

3.4.1 Inductive and Recursive Join Algorithm

This section covers how already existing productions are refined to improve their habitability. Productions are evolved by examples by means of our *Inductive and Recursive Join Algorithm.* This algorithm is motivated by the Harris' substitutability criterion [Har51], based on the idea that "constituents of the same type can be replaced by each other".

The algorithm is inductive: the first example $q \in T^*$ presented to the system is derived by a grammar production initialized as a concatenation of the terms of q arranged in the same order. It is assumed that after the i^{th} step, productions in R cover all the previous i examples. If the $(i+1)^{th}$ example $q_{i+1} \in T^*$ could not be derived, then an explanation of the form $< q_{i+1}\, is\, L >$, where

$L \in N$ is the semantic label representing the correct class for q_{i+1}, is requested —the case $L \notin N$ corresponds to the addition of a new query model, and will be later exposed in section 3.4.2. The procedure join($\cdot$) modifies the right-hand size part of the production in R to ensure the derivation of q.

The following notation is used in the algorithm. P will denote the right-hand side part of the production being modified. Arrow '$\Rrightarrow$' is used to denote the calculation returned by the join($\cdot$) procedure. The sentence q is a list of terminal symbols, $q = w_1\, w_2 \cdots w_m$, where $w_i \in T$. We use $w_{i..j}$ as an abbreviation of the substring $w_i\, w_{i+1} \cdots w_j$. Recall that symbol '|' denotes disjunction and '[' and ']' delimit optional subproductions. Finally, according to the usual notation, the empty production and the empty string will both be denoted as ε. Steps 1 and 2 represent the recursive closure, Steps 3 and 4 are recursive calls for the concatenation and disjunction cases respectively, and Step 5 is the default case.

Step 1: If $P = \varepsilon$ and $q \neq \varepsilon$
  $join(P,q) \Rrightarrow [w_{1..m}]$
Step 2: If $P \neq \varepsilon$ and $q = \varepsilon$,
  $join(P,q) \Rrightarrow [P]$
Step 3: If $P = p_1\, p_2\, \cdots\, p_k$ and exists a derivation $p_i \rightarrow^* w_{j..k}$
  $join(P,q) \Rrightarrow join(p_{1 \cdots i-1},\, w_{1..j-1})\; p_i\; join(p_{i+1 \cdots k},\, w_{k+1 \cdots m})^a$.
Step 4: If $P = p_1\,|\,p_2\,|\cdots|\,p_k$, and exists a derivation $p_i \rightarrow^* w_{j..k}$
  $join(P,q) \Rrightarrow p_1 \mid \cdots \mid p_{i-1} \mid join(p_i,\, s) \mid p_{i+1} \mid \cdots \mid p_k$
Step 5: Default
  $join(P,q) \Rrightarrow\; P\,|\,w_{1..m}$

[a] In this case, p_i acts as a *pivot*. When more than one pivots are found, the one that produces the highest number of alignments is taken. I.e. *join*($a\ b\ c\ a$, $b\ d\ a$), pivots first on b instead of on a

Example of trace: Assuming that one of the calculated productions after the i^{th} step is ListEntity → *((enumerate|show me) all the* Entity*)*, and that the explanation $< q_{i+1}\, is$ ListEntity>, with q_{i+1} = *could you show me what are the states?*, is presented in the $(i+1)^{th}$ step, then the following trace occurs:

1. ListEntity → **join**((*enumerate*|*show me*) *all the* Entity, *could you show me what are the states*)
2. ListEntity → **join**(ε, *could you*) (*enumerate*|*show me*) **join**(*all the* Entity, *what are the states*)
3. ListEntity → [*could you*] (*enumerate*|*show me*) **join**(*all*, *what are*) *the* **join**(Entity, *states*)
4. ListEntity → [*could you*] (*enumerate*|*show me*) (*all* | *what are*) the Entity

Note that the resulting production can now derive the same language that was initially derived, the new example, and some new sentences[23] such as "Could you enumerate all the states?". Thus, the join algorithm increases the generalization of the pattern. More formally:

$$\mathcal{L}(\mathcal{G}_i) \cup \{q_{i+1}\} \subseteq \mathcal{L}(\mathcal{G}_{i+1}) \tag{V.4}$$

[23] Even if the improvement is quite limited in this example, note that the coverage could be significantly benefited after various iterations.

3.4.2 Learning New Query Models

In the previous section, it was shown how the system refines an existing production. This section deals with how the system learns new Query Models[24].

According to the definition of a QM (see section 2.3.2), the addition of a new QM entails the definition of a new identifier (I), a recognizer patter (P), a set of constraints (C), and an resolution procedure (S). This could be done directly (manually), or in an assisted way (semiautomatically). Manually defining each constituent of a QM is restricted to teachers familiarized with NLP. Since this case needs no further explanation, we will examine the semiautomatic case in greater detail.

To this purpose we will consider, once again, the example of the QM AofV (the label is manually set), and one of its instances: q=“Show me the area of the lake Iliamna”. The expert offers the explanation q *is* AofV and the SQL procedure that resolves it. Let us consider firstly a baseline method whereby the example could be directly stored (see Table V.9).

QM$_{new}$	**I** := AofV
	P := Show me the area of the lake Iliamna
	C := ε
	S := SELECT Area FROM Lake WHERE Name=‘Iliamna’
Grammar	**N** := N $\cup$ {AofV}
	T := T $\cup$ q
	R := $R \cup$ {AofV→ show me the area of the lake iliamna }

Table V.9: Baseline method for acquiring new QMs

This QM is very limited since it is only able to parse and resolve the given question. The semiautomatic method tries to replace domain-dependent substrings in q (the largest possible) with non-terminal symbols N of the grammar, in order to abstract it.

We will denote (A, a), where $a \in N, A \in T^*$, to the replacement whereby the string a is replaced by the non-terminal symbol A. For simplicity, we will use $sub(q)$ to denote all substrings of q. Finally, this procedure follows these steps:

1. Select all replacements defined by $Replaze_N(\mathcal{G})$ (Equation V.5). In the example[25] (Attribute$_0$, ‘area’) and (Value$_0$, ‘lake Iliamna’).
2. Apply replacements in q. In the example: q' :=“Show me the Attribute$_0$ of the Value$_0$”.
3. Apply replacements in the resolution. In the example: SELECT Attribute$_0$ FROM Lake WHERE Name=Value$_0$

$$Replaze_N(\mathcal{G}) := \{(A, a) \in (N, sub(q)) \mid (A \Rightarrow^* a) \wedge \nexists (B, b) \in (N, sub(q)) : (B \Rightarrow^* A) \wedge (a \subseteq sub(b))\} \tag{V.5}$$

[24] Note that, although not all the productions in the grammar correspond necessarily to QMs, it is only worth considering this case, due to the fact that the knowledge structure of a DB is considered to be known in advance. For this reason, productions representing conceptual elements or relations could be rather automatically instantiated from the ontological model. It could be interesting investigating how the system learns the structure of the knowledge by analogy.

[25] Indexes are used to univocally identify each non-terminal symbol, if they are repeated.

After that, the expert is requested to correct possible errors in the replacements (for example, if (Through, of) was also applied). At this point, the expert is requested (i) to add the constraints that ensure the correct application of the SQL procedure, or (ii) to abstract the QM to make it portable. In the first case, constraints *fromEntity*(Attribute$_0$)=Lake and *fromAttribute*(Value$_0$)=Name should be added. After that, the domain-dependent QM is now able to retrieve any attribute value from a named lake. The second case is perhaps the most interesting, since it helps the QM to become portable. The expert is requested to replace each original non-terminal symbol in the SQL query (Lake, and Name in this example) with conceptual elements in basis of the rest of elements (Attribute$_0$, and Value$_0$ in this example), and to add some semantic constraints if needed (see Table V.10). Note that a QM is considered to be portable if and only if all the non-terminal symbols involved refer only to generic concepts of a DB, that is, there is not any domain-dependent entity, relation, or operation appearing on it.

QM$_{new}$	**I** := AofV
	P := Show me the Attribute of the Value
	C := *fromEntity*(Attribute$_0$) = *fromEntity*(Value$_0$)
	S := SELECT Attribute$_0$ FROM *fromEntity*(Attribute$_0$)
	WHERE *fromAttribute*(Value$_0$)='Value$_0$'
Grammar	**N** := N ∪{AofV}
	T := T ∪ q
	R := R ∪{AofV→ Show me the Attribute of the Value}

Table V.10: Aided method for acquiring new QMs

Although this second case is more tedious, it is worth defining portable QM because they could be defined only once and be reused across various domains. Moreover, as the set of portable QMs grows, the necessity of defining new ones decreases (see our discussion in section 3.5.4). Furthermore, the system becomes incremental in the sense that it is not limited in advance to certain queries types.

Finally, realize that Step 1 may simplify noticeably the effort while dealing with compound questions. For example, assuming the QM EmostRelated is being reused, if q_2 ="What is the capital of the state with most rivers?", then *Replaze*$_N(\mathcal{G})$={(Attribute$_0$, capital), (Value$_0$, the state with most rivers)}, because Value ⇒ EmostRelated ⇒* *the state with most rivers.*

3.4.3 Conjectures about the Language

Conjectures attempt to simulate how a human reasons based on his previous knowledge of the language. This method attempts to recognize new concepts by analogy. Our proposal consists of using the grammar as an structural descriptor instead of as an static resource. In this way we try to overcome the bias imposed by a non-exhaustive set of training sentences.

We will consider two types of assumptions: hypotheses (*a priori*, that is, before an explanation is given —section 3.4.3) and deductions (*a posteriori*, that is, after an explanation is given —section 3.4.3). We will use $x \in^? X$ to denote the conjecture whereby string x may belong to the semantic class $X \in N$. Analogously, $X \to^? x$ is the same conjecture expressed in form of a production. Finally, section 3.4.3 focuses on how these conjectures are validated semiautomatically.

A priori: Hypotheses To introduce the hypotheses, let us consider the following sentence: "I live in Granada". Even if you had not heard *Granada* before, you may recognize it as a *location.* Our proposal is based on the idea that some terminal productions (productions deriving only terminal symbols) are actually incomplete. Hypotheses inference consists of completing terminal productions with new hypothetical productions so that the question becomes tractable —i.e. at least one interpretation is reachable. Now, let us come back to our problem again with the more closely related question "Where is Utah?", and the following productions:

$$\textsc{Location} \rightarrow \text{where is } \textsc{Value}$$
$$\textsc{Value} \rightarrow \textsc{CityNameValue}|\ \textsc{StateNameValue}\ | \cdots$$

It is likely that *Utah* is actually a Value, even if there is no derivation satisfying Value$\Rightarrow^*$ *Utah.* The searching algorithm will add, at some point, the hypothetical productions CityNameValue $\rightarrow$ *Utah*, and StateNameValue $\rightarrow$ *Utah.* In this way, hypotheses such as *Utah* $\in^?$ CityNameValue and *Utah* $\in^?$ StateNameValue will be proposed. A second, probably more simple heuristic to the purpose, consists of proposing all unseen words and n-grams as candidate terms for hypotheses. As will be seen (section 3.4.3), hypotheses are not only useful to identify missing values, but also to detect synonyms of already existing values.

More formally, given a semantic grammar $\mathcal{G} = (T, N, R, S)$, and a question q satisfying $q \notin \mathcal{L}(\mathcal{G})$, we will say that $h_\mathcal{G} = \{x_1 \in^? X_1, x_2 \in^? X_2, ..., x_n \in^? X_n\}$, with $x_i \subset q$ and $X_i \in N$, is an hypothesis[26] by $\mathcal{G}$ about q, *iff* there is a new shallow parse, or a full parse $q \in \mathcal{L}(\mathcal{G}')$, being $\mathcal{G}' = (T', N', R', S')$, where $T' := T \cup \{x_i, 1 \leq i \leq n\}$, $N' = N$, $R' := R \cup \{X_i \rightarrow x_i, 1 \leq i \leq n\}$, and $S' = S$, such that $\forall i, S' \Rightarrow^* \alpha X_i \beta \Rightarrow \alpha x_i \beta \Rightarrow^* q$.

A posteriori: Deductions Deductions are carried out after learning a new example. Suppose the production Location $\rightarrow$ *where is* Value, and that <"Where is Utah?" *is* Location> is the next explanation offered by the teacher. Since the grammar cannot parse it, the join algorithm will modify the production, that will result in Location $\rightarrow$ *where is* (Value|*Utah*). The structure explained by this production shows that substring *where is* is followed by a Value derivation or by the *Utah* terminal symbol. This could be an indication that *Utah* is actually a Value derivation. This is the first case of *a posteriori* deduction.

Formally, first type of *a posteriori* deduction is a set $\{s \in^? V_i, 1 \leq i \leq k\}$ for each production in the grammar of the form $V_x \rightarrow \alpha(V_1|V_2|\cdots|V_k|s)\ \beta \in R$, being $\alpha, \beta \in (N \cup T)^*$ and $s \in T^*$. Following with the example trace, the system deduces *Utah* $\in^?$ Value, that states that *Utah* could be a Value. If the deduction is finally accepted, new deductions could be reached. In this case, *Utah*$\in^?$CityNameVale, *Utah*$\in^?$StateNameVale, etc. will be deducted in cascade.

The second type of *a posteriori* deduction aims to identify new concepts —semantic levels— from the observation of analogies. It is based on the idea that *sustitutable* terms may be instances of a more general concept. For example, consider that following explanations are provided to the learner —just for the purpose of the example, we assume that the learner has no prior knowledge about the domain structure yet.

Teacher: <"What is the population of Utah" *is* AofV>

[26] Although each hypothesis could gather more than one condition, the truth is that, in most cases, hypotheses contain only one conjecture. Also, allowing various conjectures in a hypothesis requires performing a costly search on the space of combinations. To this purpose, we have implemented a backtracking algorithm. Our experiences indicate that it is worth considering at most two conjectures.

Teacher: <"What is the capital of Nevada" *is* AofV>

Teacher: <"What is the length of Yellowstone" *is* AofV>

Once the learner has acquired them, the production may lead as:

AofV → what is the (population | capital | length) of (Utah | Nevada | Yellowstone)

By observing this production, one may intuit that *population*, *capital*, and *length* (similarly *Utah, Nevada*, and *Yellowstone*) are instances of some hidden concept. Indeed, hidden concepts in this example correspond to Attribute and Value, respectively. Thus, the second type of *a posteriori* deductions are productions $V_{k+1} \rightarrow^{?} (t_1|\cdots|t_m)$ (with $V_{k+1} \notin N$), proposed after observing a production $V_i \rightarrow \alpha(t_1|\cdots|t_m)\beta$ or $V_i \rightarrow (t_1|\cdots|t_m)\alpha$, with $\alpha \in (N \cup T)^+$ and $\beta \in (N \cup T)^*$. If this deduction is finally accepted, then the initial production is modified as follows: $V_i \rightarrow \alpha V_{k+1}\beta$ or $V_i \rightarrow V_{k+1}\alpha$, respectively.

The most important characteristic of second type of *a posteriori* deductions is that they help to raise domain-dependant productions to the level of domain-independence, by abstracting particular instances to concepts, in favour of portability. That is, after expert validation, the production will be:

AofV → what is the Attribute of Value

Which is now portable to other domains.

One may argue that this kind of deduction is somehow related to Term Classification [ALSZ06]. The main difference is that conjectures may involve subordinate clauses. For example, in "What is the capital of the state with most rivers", the following deduction could be reached: *the state with most rivers* $\in^?$ Value. Note that this deduction serves to evidence the lack of a suited Query Model.

Accepting or refusing conjectures So far, we have explained how the system conjectures about the grammar attempting to improve its ability to understand the language. However, since these conjectures are heuristically motivated, expert validation is required in order to accept or to refuse them.

This section covers how the learner conjectures are presented to the teacher, in such a way that the effort required to validate them tends to be minimized. To this end, we propose two methods: automatic contrast against the database (section 3.4.3), and semiautomatic contrast against the external sources (section 3.4.3).

Contrasting the assumptions against the DB Since the data stored in a database may not be static, some data could be added or modified[27]. This heuristic is based on reviewing the data stored in the database to accept new linguistic references automatically.

This mechanism assumes that an hypothesis $x \in^? X$ is correct if the value x pertains to the active domain of attribute X in the DB. This is contrasted by means of a simple SQL query for each hypothesis, as follows:

[27] Deletions do not actually represent a problem, since interpretations regarding deleted data will operate properly, returning no data.

$$\text{SELECT * FROM } \textit{fromEntity}(X) \text{ WHERE } X = \text{`}x\text{'}$$

Recall that *fromEntity(·)* is resolved through the ontology. If some value is returned, the hypothesis is accepted, and the grammar and lexicon are modified accordingly.

Contrasting the assumptions against external sources It is possible that some value actually pertains to the domain of some attribute, but there is no database entry containing it. In this case, a contrast against the external sources is performed by searching the most related attribute to the value. This strategy is inspired by the Point wise Mutual Information (PMI-IR) algorithm [Tur02b] from Information Retrieval. It consists of counting the number of hits produced by an Internet search engine, involving two terms (the value and the attribute label) with the NEAR operator[28]. This operator finds documents where the query terms are next to each other. This algorithm ranks conjectures according to the PMI score computed to each pair of terms.

Since some attribute labels are not representative enough (e.g. REQ_DEGREE in Jobdata domain), the search is also performed considering every linguistic reference to the attribute contained in its corresponding lexicon entry. Table V.11 shows some examples of the hits found for terms *C++*, *Microsoft*, and *x86* in Jobdata domain (some attributes were omitted for simplicity).

Term	Platform	Application	Area	Recruiter	Company	Salary	Language
C++	1914	101862	349	87	1858	128	**270241**
Microsoft	168462	1146284	2329	1155	**3446883**	20185	7659
x86	**66599**	6030	71	0	972	98	449

Table V.11: Example of hits produced

Hypotheses are presented to the expert as a ranked list sorted by PMI score. The expert selects through a menu-driven interface the correct hypothesis —if any— to be accepted. If one hypothesis is accepted, this process is repeated in cascade in order to determine whether the new linguistic reference is a new value, or a synonymous of an already existing value. Let us consider the following table and production to illustrate this case:

...	**University**	...
...	University of Granada	...
...	University of Berkeley	...

Production

NUMSTUDENTS → How many students are [there] in the UNIVERSITY
UNIVERSITY → university of Granada | university of Berkeley | ...

Assume the system proposes the hypothesis UGR $\in^?$ UNIVERSITY at some time. Since *UGR* is an acronym of *University of Granada*, it is actually a synonym, not a new value. Thus, if UGR $\in^?$ UNIVERSITY is accepted by the teacher, the system computes the PMI between *UGR* and each value in UNIVERSITY and engages the teacher with a new validation dialogue. This heuristic present three main advantages: (i) since hypotheses are ranked, it is likely that the expert waste less time to validate them, (ii) the contrast of conjectures exploits information beyond the boundaries of the given domain, in contrast to other methods such as the *string similarity* used in FREyA [DAC12], and (iii) this method allows the learner to identify not only new linguistic data, but also synonymous.

[28] We used the exalead (`http://www.exalead.com/search/`) search engine to implement this method. In contrast to Turney's notation (NEAR), this operator is called NEXT in *exalead*. For example: `http://www.exalead.com/search/web/results/?q=Microsoft+NEXT+Company`

3.5 Performance Evaluation

Simplifying the customization is arguably one of the most valuable features one could expect in a NLI. For this reason, maintaining a good balance between performance and customization effort was established among our prior goals in this research. Since the performance of the system was already validated in the previous chapter, it is the objective of this section to evaluate the system in terms of customization effort and portability, and the suitability of hypotheses and deductions in this regard.

3.5.1 Customization Test

To put the customization method to test, we have designed the following experiment. For each dataset we took the same mutually exclusive base of questions that defined the Training and Testing sets in our previous evaluation (section 2.4). First, the expert was requested to manually initialize the lexicon for each domain. WordNet was used as the only tool at his disposal to complete this task. After that, the Training Bases of Questions were used to customize the system following the interaction protocol defined in section 3.3.3. A parallel thread was continuously trying to interpret each testing question in a background process. For each customization experiment, we trace the following measurements in function of the Interactions (represented in the abscissa): (i) percentage of remaining training questions, (ii) percentage of tractable testing questions, (iii) the number of Query Models, (iv) the number of explanations provided by the teacher, (v) the number of hypotheses and (vi) deductions accepted, and (vii) the total time spent so far. Figure v.8 represents how each parameter evolves as the teacher-learner interactions increased for the Geobase domain.

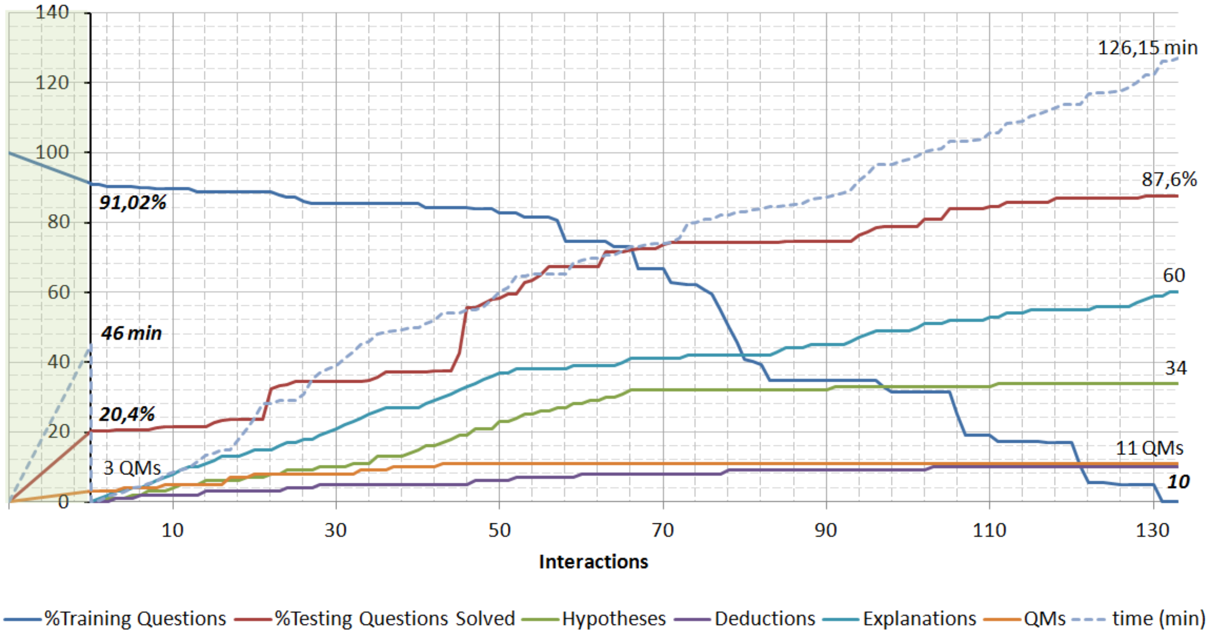


Figure v.8: Customization trace for Geobase domain

The shadowed region in the left area represents the initialization stage of the lexicon. Also in this case, three Query Models were defined —AofV, ErelatedToV, and FoperationE[29].

[29] AoperationE resolves queries requesting for the elements of an entity that have some values satisfying certain

Each pattern was initialized applying the join algorithm to three representative queries posed by the expert. Once the initialization process ended —after 46 minutes— the system was able to interpret 51 testing queries (20.4%), and 79 training questions (8,98%). It took 133 human-computer interactions, including explanations, and validation of hypothesis and deductions, to customize the system. Recall that an explanation could derive in the refinement of an already existing QM (as defined in section 3.4.1), or in the definition of a new one (see section 3.4.2). Note also that most of the conjectures took place after early explanations. Once the customization ended, the system counted with eleven Query Models. After approximately 3 hours (46min + 127min), the system was able to interpret 87.6% of the testing questions. We have omitted here the necessary time for the computer to interpret and retrieve the results. Thus, it should be pointed out that actually the total customization time was approximately 5 hours. The reason why we discarded these times in the calculation is because we think these times are more affected by our particular implementation and the specifications of the computer being used[30], rather than by the customization methodology under consideration. Some similar estimations reported for related methods in the literature are listed below: 6 hours in LTAG-grammars [UHC10], from "a few hours" to 6 hours depending on the domain for ORAKEL [CHH07], or from minutes to "a few hours" in TEAM [GAMP87].

3.5.2 Portability validation

This previous analysis helps us to appreciate how the customization evolves after each interaction, so as to empirically observe how effective is the interaction protocol. A second, possibly more interesting objective in this analysis, is to analyse the effectiveness of our policy in terms of portability. To this aim, we repeated the same process in the Restbase domain (Figure v.9 left), taking as starting point the QMs defined for Geobase. After that, we reuse the resulting QMs to customize the Jobdata domain (Figure v.9 right).

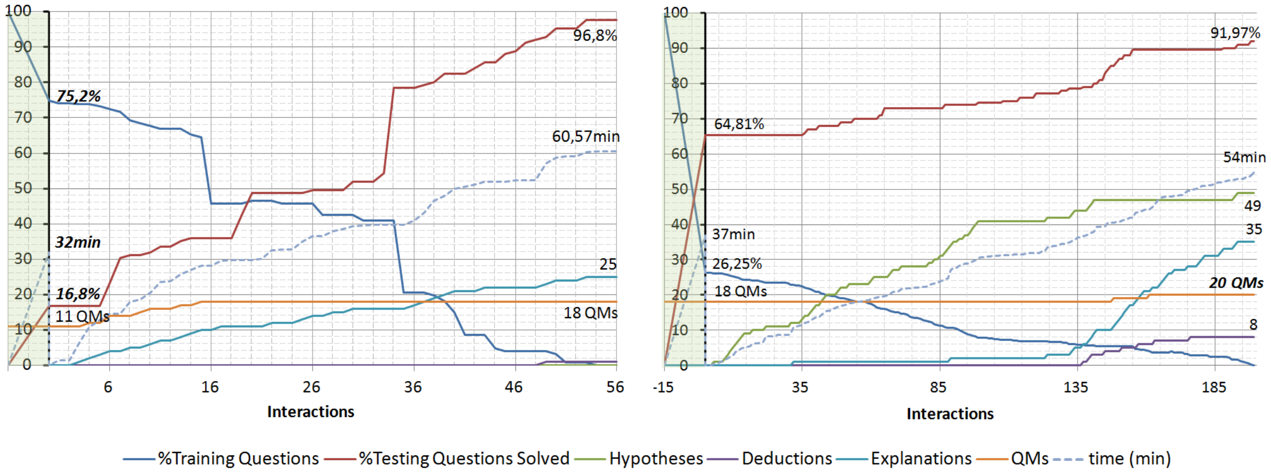


Figure v.9: Customization trace for Restbase (left) and Jobdata (right)

As could be seen, the customization time was significantly reduced. However, because each dataset presents different characteristics, claiming this reduction was due to portability would not be fair. Indeed, just few QMs from Geobase were also useful in Restbase. Notwithstanding, the

operation. For example "what is the longest river", where operation Greatest is applied to the numerical attribute Length in entity River

[30] An Intel(R) Core(TM)2 Quad Q8200 2.33GHz with 6GBytes RAM was used to carry out the tests.

collection of QMs obtained after customizing both Geobase and Restbase was determinant to the customization time of Jobdata —a large number of training and testing questions could be interpreted in advance. Concretely, it took approximately one hour and a half (32min+61min, and 37min+54min respectively) to customize both domains —while one may have expected more time for Jobdata than for Restbase. It was needed to add seven new Query Models —in the firsts interactions— to better cover the questions' variety for Restbase. Further explanations aimed to increase the lexicon variety or to improve the QMs representation. In the case of Jobdata, only two new QMs —dealing with domain-dependant operations involving the entity SALARY—, where needed, and almost all the customization effort was devoted to attend hypotheses (first interactions), and deductions (last interactions). It is fair to mention that the time to perform the grammar parsings and retrieve solutions was also removed from the calculations in these experiments. If considering it, the total time to customize each system was approximately three hours and half for Restbase, and four hours for Jobdata domain —where the backtracking algorithm along with the large number of registers in the database took their tool.

Since some Query Models were reused from one domain to another, the customization effort was alleviated. To support this claim, we have analysed the traces of execution in our experiments, and we have calculated the frequency each Query Model was involved in a parsing. According to our expectations, we found that just few Query Models were involved in most of the interpretations. Query Models such as ERELATEDTOV and AOFV contributed to 66% of all parsings (Figure v.10).

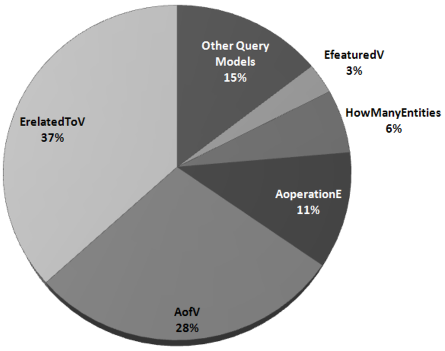


Figure v.10: Frequency of Query Models involved in all parsings of these experiments.

3.5.3 Hypotheses and Deductions validation

Jobdata is the only domain containing queries referring to data not stored in the database. In order to validate the contrast of hypotheses method, we have performed a test against all 53 terms from Jobdata not appearing in the database. Those 53 terms appeared in 34.11% of the base of questions. As we did in chapter IV section 2.4, the classification performance, of hypotheses in this case, will be evaluated by means of Accuracy and Mean Reciprocal Rank (MRR) measures. Recall that accuracy measures the proportion of correctly classified hypothesis among all classified cases (see Equation V.6). MRR is a statistic measure to evaluate any process that produces a ranked list of possible classifications to a case. The Reciprocal Rank of a hypothesis classification is the multiplicative inverse of the rank of the first correct case. The MRR is the average of the Reciprocal Ranks resulting for a sample of cases (see Equation V.7).

$$Accuracy = \frac{|\{h \in H | rank(h) = 1\}|}{|H|} \tag{V.6}$$

$$MRR = \frac{1}{|H|} \cdot \sum_{h \in H} \frac{1}{rank(h)} \tag{V.7}$$

We have achieved Accuracy = 49.05% and MRR = 0.721. Indeed, in most cases (79.16% in our experiments), the correct classification appeared in the first or second position. This is an indication that our method (section 3.4.3) simplifies the expert validation of hypotheses in favour of alleviating the customization effort. Although rank suggestions were also evaluated in terms of MRR for the FREyA system, a direct comparison is not feasible. The main difference is that, in FREyA, clarification dialogues are meant to solve ambiguities referring to already existing concepts in the KB, while in our case, the system could also attempt to solve references to non-existing data.

3.5.4 Final discussions on Customization

Time was usually reported in the literature as the most influential parameter affecting the customization effort. In this regard, we believe there are some other aspects that effectively take part while customizing. Besides the fact that times here reported are comparable to state-of-the-art ones, the customization methodology proposed relies on simple human-computer interactions, which are mainly directed by the learner. As a result, it is easier for the teacher to accomplish with this task. The most complex interaction is that of defining new QMs. However, since already existing QMs could be reused across domains, the necessity of defining new QMs decreases in the long term (as reflected in our portability study). In turn, beyond the difficulty of defining new QMs, we believe this mechanism increases the potential applicability of the methodology —in contrast to other methods such as ORAKEL or PANTO, the types of queries our system can deal with are not limited in advance.

The main drawback one may argue against this method, lies on the necessity of a suitable Base of Questions for the purpose. Indeed, there is not always the case an expert count in advance with such a list of representative questions. However, collecting query logs is an easy task that could be carried out in the early steps of the system. In turn, it could be regarded as a solid mechanism to continuously debug the system [CHH07].

3.6 Conclusions and Future Work

In this section, we have investigated how human-computer interactions could serve to alleviate the work an expert should undertake to customize a system. To this aim, we have designed an inductive grammar learning algorithm that continuously improves the language coverage of a NLI. The system proposed in the previous section was enhanced with language acquisition techniques based on conjectures about the inferred grammar. Empirical results indicate our method is comparable to state-of-the-art ones in terms of performance and customization time. The main contribution of this work consists of the ability of the system to reason about its previous knowledge and meta-knowledge to deal with unseen sentences and to propose conjectures.

HITS relies on the teacher-learner model inspired by the Angluin's paradigm. Usually, time was considered the most determining factor regarding the customization effort in the literature. The proposed learning protocol allows the customization process to be mainly directed by the system (as TEAM does), that engages the expert with dialogues (like FREyA). The main difference however is that our system generates dialogue based on conjectures (hypotheses and deductions) on the language. Beyond the time required to adapt the system, we believe the effective effort is substantially alleviated with our methodology.

Also, the collection of QMs could be iteratively refined to better fit the questions on a certain domain, that is, to gain in *habitability.* As a counter part, a suitable Base of Questions is required to this purpose. However, collecting query-logs is an easy task that could rather be regarded as a solid process to continuously debug the system. Arguably, the most costly task in the present methodology concerns with the definition of new QMs. To partially override this drawback, a method to assist the expert was here proposed. Also, it should be pointed out that, because QMs are portable, the necessity of adding new QMs decreases in the long term. Finally, as new QMs could be defined at any time, the system is not limited in advance to a certain sort of questions.

This research was also motivated by the study of grammatical inference from a cognitive point of view. Besides defining a new grammar learning algorithm, our goal was to bring the system the ability to reason by analogy, based on its previous knowledge. In this line, hypotheses and deductions have played a key role to simplify the adaptation of the system. We believe these techniques could be useful from a more general point of view regarding the SK/PM class of problems. Even if the mechanism that generates the grammar is independent of the particular DataBase under consideration, the grammar is aligned to elements that are inherent to the ontology conceptualization (such as entities, relations, or values). We are thus interested in investigating the potential of the hypotheses and deductions in KBs where the structure is not known in advance, that is, in the UK/PM class. In contrast to automatic techniques such as ABL [Zaa01], we plan to combine the conjectures with expert explanations and validations in order to improve the semantic representation of the grammar. We believe the hypotheses and deductions schema could become a direct contribution in the field of FAQ-retrieval, where the concept of Base of Questions is naturally represented in the FAQ list.

Chapter VI

Conclusions and Future Work

This PhD dissertation was devoted to offer an study on closed-domain NL approaches in basis of the domain knowledge. This final chapter is to offer our main conclusions and to outline the avenues of our future research.

1 Conclusions

Following the methodology proposed in chapter I section 4, the closed-domain NL approaches could be categorized attending to the knowledge structure and the presence or absence of meta-knowledge.

Because we have approached each category from a representative problem member, we will first present our main conclusions on that problem and then some more general conclusions extensible to the entire category. After that and for the sake of consistency, the global conclusions of this Thesis will be presented.

1.1 Unstructured Knowledge, Absence of Meta-Knowledge: Text Classification

The main goal of this preliminary study concerned with the identification of main characteristics of the most relevant terms in a domain. This problem is of great importance insofar the term-relevancy criterion is present in most of the NL technologies. To our aim, we took the Feature Selection for Text Classification problem as a starting point.

The Feature Selection problem consists of reducing the initial feature space to generate a reduced set of features before training the classifier. To this end, filtering methods evaluate the informativeness of each feature separately. In our study, we investigated the influence of different filtering functions, selection policies, learner devices, evaluation metrics, and datasets, taken among the most popular in the field. Results reinforced the robustness of SVMs classifiers and the Round Robin policy. Surprisingly, we found that the filtering function GSS outperformed other filtering functions such as Information Gain or Chi-Square when combined with the Round Robin policy in our experiments.

From a more general point of view regarding the UK/AM approaches, following conclusions could be drawn (i) while searching for the most relevant terms in a domain, all categories should be equally considered, (ii) the classification performance could be improved from an aggressive reduction in the initial feature space, and (iii) the positive correlation seems to be the best criterion to represent the informativeness of a term to a domain.

1.2 Unstructured Knowledge, Presence of Meta-Knowledge: Sentiment Analysis

We devoted our second case study to address the problem of context delimitation before analysing the text documents. To this purpose, we took the Sentiment Analysis on News items as a starting point. This problem is a branch of Sentiment Analysis, that received a great deal of attention given the large number of potential applications for marketing strategies, political campaigns, or decision making for costumers.

Loosely speaking, the problem consists of identifying the main focuses on which opinions are expressed and to assign a central value summarizing the polarity and strength to each one. In pursuing this goal, we defined a hierarchical meta-knowledge model —a domain-lexicon— based on the notion of two concepts: abstract entities and specific or domain-dependant entities. Since the lexicon was modularised, the expert in charge of customising the system could tune the model before performing the analysis. Apart from this fact, hierarchical structure of the lexicon provides implicit information that could be effectively exploited in order to automatically identify main opinion focuses in the text. In our experiments, we obtained an improvement of 5% ($F-measure$) in the analysis of sentiments by delimiting the context in advance.

From a more general perspective regarding the UK/PM approaches, following conclusions could be drawn. Even if there is not any clue on the structure of the documents, the presence of a suited meta-knowledge model allows a potential application to previously analyse the context in order to identify and delimit the main entities involved. Semantic information in the model could, in addition, be exploited to resolve certain NL-related problems such as the anaphora, ellipsis, or ambiguity ones. Moreover, since a domain lexicon facilitates the compilation of colloquial expressions, technical terms, and domain-dependant terms, a more reliable analysis of related texts could be achieved.

1.3 Structured Knowledge, Absence of Meta-Knowledge: FAQ Retrieval

Our third case study concerned with collaborative NL systems. We based our study on automatic FAQ retrieval, a clear representative problem where knowledge could be naturally enhanced just by adding new question/answer pairs.

More specifically, we focused on (i) designing a new question/answer retrieval algorithm, (ii) investigating how the quality of a FAQ could be monitored, and (iii) bringing interpretability to templates to allow an expert revise, and possibly refine, the system performance. First, we formally proved that obtaining a suited set of templates satisfying certain desirable properties could be effectively computable. Then, we proposed an specific algorithm to obtain such templates based on the notion of differentiability and minimality criteria, and paying special attention to the scalability and efficiency of the method. Concretely, our quadratic algorithm shown an improvement between 2% and 7% in Accuracy and about 6% in MRR while compared against other state-of-the-art FAQ retrieval algorithms in our experiments. We had the opportunity to put the system to test in a real scenario: *FAQtory*. This framework implemented usage mining techniques to discover knowledge gaps and weaknesses in order to provide the expert in charge with the necessary information so as to improve the FAQ content. As a result, the knowledge in the system could evolve according to real users' information needs, instead of just by expert intuition. Finally, we proposed a method for bringing interpretability to linguistic templates. To this aim we framed the problem as an optimisation problem in basis of the interpretability, correctness, and generality degrees of each regular expression. This formulation was, in addition, tuned according to the specific judgements of the expert in charge. We estimated the time to create a template-based system could be reduced from 40% to 50% with our method.

The SK/AM configuration represents an appropriate scenario for developing collaborative systems. Since there is no meta-knowledge modelling, there is no technical barrier for users, who are not expected to be experts in computational linguistics or Knowledge Engineering either. Users are allowed to collaborate in a shared environment just by adding Information Units following the knowledge structure. In light of the results it seems that scalable techniques based on the notion of differentiability could be defined for the purpose. In this regard, results seemed to indicate that isolated words do not provide enough information to classify. Instead, is the concept of "relations among words" which might better contribute to the classification task. In addition, collaborative systems could incorporate usage mining techniques to discover information weaknesses and users tendencies. Thus, the system provides the expert with the necessary information to improve the knowledge content, if needed. We devoted the final part of this study to critical systems requiring additional expert monitoring. In this regard, we investigated how the retrieval mechanisms could become interpretable according to experts' criteria. By appropriately tuning a fitness function reflecting some important parameters, we believe our method could be extensible to reduce the costs associated to the creation of critical SK/AM systems, while the performance could still be monitored by experts.

1.4 Structured Knowledge, Presence of Meta-Knowledge: Natural Language Interfaces

The last part of this Thesis concerned with systems presenting the highest knowledge levels. We focused in this case on Natural Language Interfaces as a representative problem. The interest of this problem is justified by the fact that much of the digital information is currently stored in Databases and regular users do not know any formal query language. Even if good solutions were reached in the past, the truth is that alleviating the costs associated to customise or adapt the system do still represents an open-problem with enough room for improvement.

NLIs are meant to overcome technical barriers allowing non-experienced users to access information stored in knowledge bases in a natural manner. In this study, we presented a complete framework based on the notion of Query Model, an abstract mechanism to represent query resolution procedures. Since these models are domain-independent, they could be reused across domains in favour of portability. We defined a customisation protocol relying on human-computer interactions to reduce the effort an expert should spend while configuring the system. To this aim, we designed an inductive algorithm for learning context-free grammars. This algorithm exploits the sustitutability criterion in language to allow reasoning by analogy and hypotheses inference. Since Query Models could be added at any time, the answering capabilities of the system are not delimited in advance. Furthermore, the system was able to learn from users queries, incurring in a continuous improvement of its habitability. Experimental results seemed to indicate our system is comparable to state-of-the-art methods both in performance ($F - measure$) and in customization effort (time), while obtaining high-precision scores —close to 100% in all cases. Arguably, the most representative feature in our system consists of the language hypotheses, a mechanism to attempt interpreting unseen sentences by analogy.

The SK/PM configuration represents the highest knowledge level achievable: knowledge is structured and there is a suited meta-knowledge model available representing main entities and relations in the domain. This scenario facilitates automatic reasoning and knowledge inference to reach high-level interpretations of NL. One of the main problems to be fixed in this scenario concerns with how to successfully map domain concepts with users expectations on the domain. In this regard, we have proposed abstract and portable methods to face the so-called "bridge the gap" problem that could be extensible to different structured knowledge bases. This work was mainly motivated by the study of grammatical inference from a cognitive and computational point of view. Our proposal was thus based on language learning through an interactive process that continuously refine the knowledge. The sustitutability criteria along with the intermediate semantic representations led us incorporate reasoning by analogy (i) to infer structural analogies with past cases (*deductions*), and (ii) to propose conjectures about unseen expressions (*hypotheses*), two mechanisms that could be useful in different SK/PM problems.

1.5 General Conclusions

The study of the NL phenomenon from a computational point of view is motivated by the large number of potential applications that could be created in the field of information technologies. In such context, our hypothesis is that both the system performance and the reduction in the associated costs could be benefited from specific techniques being aware of these knowledge-levels. Bearing this principle in mind we have here addressed some open-problems of social interest to which there is still room for improvement. Our main goal was to offer better solutions to these problems that are extensible to a broader set of problems under similar conditions. To that end, we have first proposed a categorization of the different NL technologies in basis of the knowledge levels available.

More specifically, the following issues were investigated in this dissertation (i) what makes a term become relevant to a given domain, (ii) how to delimit the discourse context to improve the analysis of NL, (iii) knowledge modelling and refinement methods, and (iv) the reduction in customisation times while adapting the system to new domains. Our main conclusions in this regard could be summarized as follows:

- Differentiability and Minimality criteria, based on positive correlation and aggressive features reduction, seem to represent an effective mechanism to classify linguistic expressions in closed-domains. Relations among words contribute better to the classification task than isolated terms.

- The analysis of text documents could be improved by delimiting the discourse context in advance. To this purpose frequency-based techniques and semantic inference seem to be effective.

- Information generated by users' reactions while interacting with the system could be mined in order to discover certain knowledge gaps and users tendencies. Results seemed to indicate that these usage mining techniques could be exploited in favour of collaborative systems allowing the knowledge evolve according to real users' information needs.

- Meta-knowledge models could be useful to implement specific techniques dealing with the semantic level of language. Building a suited meta-knowledge model entails however a sizeable amount of effort. According to our study, it seems that this effort could be alleviated through reasoning by analogy. Identifying structural analogies may allow to discover new concepts to improve the current knowledge representation and to deal with unseen expressions by means of hypotheses.

This Thesis results from an engineering process of scientific nature. From the point of view of the engineering process, this work has resulted in various computer applications of commercial interest. This is the case of *Opinium*, a framework for Sentiment Analysis on News items, and *FAQtory*, a framework for automatic FAQ retrieval and FAQs management. Furthermore, some techniques here proposed have finally been integrated into commercial applications, or are currently being developed as part of different working projects.

From the scientific point of view, this work has resulted in various publications in international journals as well as in scientific conferences (chapter VII).

2 Future Work

We have already outlined our future work and research interests throughout each chapter. We thus will here highlight only the most imminent and interesting research to our eyes.

First, we will focus our efforts on the joint research we started during the period of stay. Our aim is to explore new filtering methods in Feature Selection for Text Classification. We are motivated by the conclusions drawn in chapter II regarding the effectiveness of positive correlation, and the work [TA03, XLL$^+$08, Joa05] revealing the influence of evaluation metrics in the classification task. We believe there is still much work ahead investigating new feature selection strategies aware of the evaluation metrics.

We are currently involved in the development of two prototypes of *Opinium*, one devoted to the analysis of political comments in social networks, and other devoted to the automatic summarization of user satisfaction surveys. In a similar way, we intend to combine our FAQ retrieval algorithm to the collaborative learning system proposed in chapter IV section 5.2, as well as integrating the visualization functionality of *FAQ clouds* [RMC13a] into *FAQtory*.

Finally, we are interested in investigating the duality of the conclusions stated in chapter V. Results seemed to indicate that starting from a structured knowledge model it could be possible to exploit analogy in language to improve the knowledge representation. We are thus aimed for investigating to what extent the inner structure of an initially unstructured knowledge resource could be inferred by analogy. It could be the case that this study —inherently related to automatic ontology learning and grammar inference— could benefit from publicly available resources, such as Wikipedia, to automatise certain aspects involved in the customization stage. In this regard, we believe methods relying on human-computer interaction, along the lines of methods here discussed, could serve to the purpose by properly generalizing the concept of *explanation*.

Chapter VI

Conclusiones y Trabajos Futuros

Este trabajo de Tesis ha servido para ofrecer un estudio sobre los sistemas basados en LN de dominio cerrado en base a los distintos niveles de conocimiento disponibles. Dedicaremos esta última sección a exponer y comentar las principales conclusiones que se derivan de este proyecto y a esbozar lo que representarán nuestros futuros objetivos.

1 Conclusiones

De acuerdo a nuestra metodología (capítulo I, sección 4) se propone la clasificación de los sistemas de dominio cerrado en función de la estructura interna del conocimiento (estructurado o no estructurado) y la presencia o ausencia de metaconocimiento.

Puesto que hemos propuesto una aproximación desde el estudio de problemas particulares hasta su generalización a grupos de problemas semejantes, expondremos en primer lugar las conclusiones principales que hemos extraído del estudio pormenorizado del problema seleccionado y luego aquellas de carácter más general. Por coherencia, expondremos en último lugar las conclusiones generales que se derivan de esta Tesis.

1.1 Conocimiento No Estructurado y Ausencia de Metaconocimiento: el problema de la Clasificación de Textos

En este estudio preliminar, nos marcamos como objetivo la identificación de las principales propiedades que confieren relevancia a un término en un dominio concreto. Recordemos que este problema es importante por representar un criterio común a la mayoría de tecnologías de LN en dominio cerrado. Para ello, tomamos como referencia la Selección de Características como parte fundamental del problema de la Clasificación de Textos.

El problema seleccionado consiste en reducir el conjunto de términos inicial (o *características*, según la nomenclatura del problema) antes de entrenar un clasificador de documentos. Para ello, los llamados métodos de filtrado pretenden evaluar la capacidad informativa de cada término con respecto al dominio, de forma aislada. En nuestro estudio preliminar, consideramos algunas de las funciones de filtrado y políticas de selección más relevantes, algunos de los clasificadores más destacados, y las métricas y bancos de datos más extendidos en la literatura especializada. Nuestros resultados reforzaron la valía tanto de las Máquinas de Vectores Soporte como de la política *Round*

Robin en la tarea de clasificación. Sin embargo, y en contra de lo que cabría esperar de acuerdo a las tendencias actuales, observamos en nuestros resultados que la función de filtrado *GSS* junto con la política *Round Robin* obtuvo mejores resultados que las clásicas *Information Gain* o *Chi-Square* para la selección de características.

Desde un punto de vista más general, y con respecto a los sistemas con conocimiento no estructurado y sin metaconocimiento disponible (CN/AM), podríamos concluir que (i) al buscar los términos más relevantes del dominio, todas las categorías deberían ser tenidas en igual consideración, (ii) la eficacia del clasificador puede mantenerse, e incluso mejorarse, tras una reducción drástica del conjunto de términos inicial, y (iii) el criterio de correlación exclusivamente positiva parece ser un mejor representante de la capacidad informativa de una palabra en un texto.

1.2 Conocimiento No Estructurado y Presencia de Metaconocimiento: el problema del Análisis Automático de Opiniones

En el segundo estudio de caso abordamos el problema de la delimitación del contexto como paso previo al desempeño del análisis de documentos escritos. A este fin, tomamos como referencia el problema del Análisis Automático de Opiniones en noticias. Este es a su vez un subproblema del Análisis de Opiniones, de actual interés para estrategias de marketing, campañas políticas y toma de decisiones de potenciales consumidores.

En términos generales, el problema consiste en identificar los principales focos de opinión y decidir, para cada uno, la polaridad e intensividad de las opiniones mediante un valor centralizado que representa el resumen de opinión. Para abordar el problema, definimos un modelo de metaconocimiento estructurado basado en dos tipos de conceptos principales: entidades abstractas, y entidades específicas o del dominio. La modularización de este recurso nos permitió adaptar el conocimiento en un paso previo al análisis. Por otro lado, la estructura jerárquica de estos lexicones encierra un conocimiento implícito que podría explotarse para mejorar la detección de los principales focos de opinión en un texto. Gracias a la delimitación previa del contexto obtuvimos una mejora en torno al 5% ($F - measure$) en nuestros experimentos de análisis de opinión en noticias.

Con respecto a los sistemas de conocimiento no estructurado y con metaconocimiento disponible (CN/PM) en general, podemos esbozar las siguientes conclusiones. A pesar de que no haya un conocimiento previo de la organización interna de los documentos, el metaconocimiento podría permitir a una potencial aplicación realizar un análisis previo del contexto destinado a delimitar las principales entidades involucradas. La información semántica recogida en el metaconocimiento puede ser de utilidad para abordar algunas complejidades presentes en el LN, como son la anáfora, la elipsis, o la ambigüedad. Además, la presencia de un lexicón de dominio facilita la recopilación de expresiones coloquiales, de tecnicismos y de terminología propia de un dominio que permitiría a una aplicación de LN realizar un análisis más fiable del texto.

1.3 Conocimiento Estructurado y Ausencia de Metaconocimiento: el problema de la Recuperación de FAQs

Nuestro tercer estudio de caso se dedicó a los sistemas colaborativos basados en LN. Como ejemplo, tomamos el problema de la recuperación automática de FAQs, un representante claro de sistemas incrementales que se sustenta de la adición de preguntas y respuestas para mejorar el conocimiento.

En concreto, nos centramos en diseñar un algoritmo para la recuperación de las parejas de pregunta/respuesta más relevantes con respecto a una consulta del usuario; cómo mejorar la calidad de un sistema de recuperación de FAQs; y cómo diseñar plantillas interpretables por un humano para facilitar la monitorización y el refinado experto. En primer lugar, demostramos formalmente que es posible obtener un conjunto de plantillas satisfaciendo determinadas propiedades deseables. Presentamos luego un algoritmo para obtener un conjunto de plantillas basado en los criterios de diferenciabilidad y minimalidad, haciendo especial hincapié en la escalabilidad y eficiencia del método. En concreto, nuestro algoritmo cuadrático obtuvo mejoras entre un 2% y un 7% en *Accuracy*, y aproximadamente un 6% en MRR con respecto a otros algoritmos de recuperación de FAQs en nuestros experimentos. Hemos tenido la oportunidad de poner a prueba nuestro algoritmo en un sistema real, que a su vez explotaba información de uso para descubrir debilidades y carencias en sus FAQs. Esto nos ha permitido diseñar un sistema de FAQ colaborativo que permite en cierto grado evolucionar a partir de las necesidades de sus usuarios, más que por la intuición del experto al cargo. Por último, abordamos el problema de generación de plantillas interpretables, de interés en sistemas críticos en los que una compañía u organización decide invertir ciertos costes adicionales en favor de monitorizar su funcionamiento. Para ello, replanteamos el problema como un método de optimización. En este caso, diseñamos una formulación reflejando criterios de interpretabilidad, correctitud y generalidad, que eran ponderados según las preferencias del experto. Según nuestras observaciones experimentales, el tiempo de creación manual de plantillas podría reducirse en torno a un 40% y un 50% con este método.

La configuración CE/AM representa un marco de trabajo muy apropiado para el desempeño de sistemas colaborativos, dado que la ausencia de un nivel de metainformación elimina la barrera tecnológica que se impone a sus usuarios: no se espera de ellos que sean expertos en materias como lingüística computacional o ingeniería del conocimiento. Únicamente respetando la estructura de información, distintos usuarios podrían colaborar activamente en sistemas incrementales a partir de la adición de Unidades de Información (UIs). En este estudio hemos propuesto técnicas de recuperación escalables basadas en la diferenciación de las UIs. Nuestros experimentos parecen indicar que las relaciones entre palabras aportan más información al proceso de clasificación que los grupos de palabras aisladas. Generalizando nuestros resultados, pensamos que sería posible explotar la información de uso para detectar algunas carencias de información en la base de conocimiento, e incluso desvelar ciertas tendencias de interés de los usuarios. Finalmente, hemos estudiado aquellos casos en los que el funcionamiento del sistema es crítico, y requiere de un control y monitorización experta. Hemos investigado métodos para conferir interpretabilidad a los mecanismos de recuperación de las UIs adaptándolos a las preferencias de los expertos. Aplicando técnicas similares a las aquí propuestas pensamos que sería posible reducir el coste de este tipo de sistemas críticos al tiempo que su funcionamiento podría seguir siendo monitorizado por expertos.

1.4 Conocimiento Estructurado y Metaconocimiento Disponible: el problema de las Interfaces en Lenguaje Natural

El último estudio se centró en los sistemas con mayor nivel de conocimiento disponible. Tomamos en este caso el problema de las Interfaces en Lenguaje Natural (NLI) como un ejemplo representativo. Puesto que gran parte de la información digital se halla almacenada de forma estructurada y la mayoría de usuarios desconoce el uso de lenguajes de consulta formales, este tipo de sistemas es de gran interés. Aunque se han alcanzado buenas soluciones para dominios específicos en el pasado, lo cierto es que el coste derivado del proceso de configuración y adaptación entre dominios sigue representando un problema abierto donde aún hay mucho por hacer.

Las NLI se proponen como un medio para superar las barreras tecnológicas que imponen los complejos lenguajes de consulta formales para el usuario medio. En este trabajo, hemos presentado un sistema completo basado en la noción de *modelo de pregunta*, un mecanismo de representación abstracto de tipos de consultas. Ya que estos mecanismos se formulan de forma independiente de los datos, pueden considerarse portables entre dominios y podrían, por tanto, reutilizarse para disminuir el coste de configuración. Hemos definido un protocolo de adaptación del conocimiento basado en la interacción hombre-máquina que permite al sistema dirigir parte del proceso de configuración, facilitando así la tarea del experto al cargo. Para ello, hemos diseñado un algoritmo inductivo de aprendizaje de gramáticas libres de contexto. Este algoritmo explota los criterios de sustitución en el lenguaje para simular razonamiento por analogía y propuesta de hipótesis. Puesto que los modelos de preguntas son aditivos, la capacidad del sistema no está limitada a un cierto tipo de preguntas de antemano. De esta forma, el sistema aprende con el uso cómo interpretar las preguntas que los usuarios tienden a realizar. Nuestros resultados parecen indicar que el sistema es comparable con otros modelos del estado del arte en (i) rendimiento ($F-measure$) manteniendo además una índice de precisión muy cercano al 100% en todos los casos, y (ii) en tiempo de configuración. Además, nuestro modelo incorpora el concepto de hipótesis en el lenguaje, un proceso basado en intentar deducir interpretaciones de expresiones nunca vistas por medio de analogía con casos conocidos.

La configuración CE/PM representa el mayor nivel de conocimiento disponible: el conocimiento está estructurado y existe un nivel de metaconocimiento explicando las entidades y relaciones del mismo. Por medio de estos niveles de conocimiento podrían alcanzarse interpretaciones avanzadas del LN. Uno de los principales problemas en este contexto consiste en establecer una relación entre la base de conocimiento y las expectativas de los usuarios. En este trabajo hemos propuesto una metodología para afrontar este problema conocido como “rellenar el hueco”. Esta metodología ha estado motivada principalmente por el estudio de la inferencia gramatical desde un punto de vista cognitivo y computacional. Para abordarlo, nos hemos basado en el criterios de *sustitución* y hemos intentado explotar la representación semántica para simular razonamiento. Este proceso de razonamiento, destinado a deducir congruencias entre expresiones previas (*deducciones*) y expresiones nuevas (*hipótesis*), podría ser extensible a otros muchos problemas en CE/PM. De esta forma, hemos pretendido abordar el problema del aprendizaje del lenguaje como un proceso interactivo de razonamiento y validación constante.

1.5 Conclusiones Generales

El estudio del LN como un sistema de comunicación desde un punto de vista computacional encierra un enorme potencial de cara a las tecnologías de la información. En este contexto, nuestra hipótesis de partida es que las tecnologías de LN pueden beneficiarse de la premisa de dominio cerrado explotando los diferentes niveles de conocimiento para ofrecer mejores soluciones a problemas de gran demanda social y que actualmente no podemos considerar resueltos. Nuestro objetivo principal consistió precisamente en proponer mejoras puntuales para estos problemas de interés y que fueran, en la medida de lo posible, extensibles a una clase de problemas más general. Para ello, hemos planteado primero una clasificación de los problemas y tecnologías de LN en función de las características del conocimiento que gestionan.

Entre los principales técnicas propuestas para explotar o mejorar los diferentes niveles de conocimiento podemos resaltar el estudio de la importancia relativa de las palabras en un dominio, la delimitación del contexto como paso previo al análisis, el modelado y refinado de los recursos de conocimiento, y la disminución de los costes de configuración de un sistema de LN. Podemos extraer las siguientes conclusiones generales de este trabajo:

- Los criterios de diferenciabilidad y minimalidad, basados a su vez en el principio de correlación positiva y reducción agresiva de características, parecen representar un mecanismo eficaz para la clasificación de expresiones lingüísticas en dominios cerrados. Nuestros resultados indican que son las relaciones entre palabras, más que conjuntos de palabras aisladas, las que mejor información aportan a la clasificación de expresiones lingüísticas.

- Las técnicas basadas en frecuencia y deducciones semánticas pueden ser de ayuda para mejorar el análisis automático de documentos escritos por medio de un pre-análisis del contexto.
- La información generada por las interacciones de uso puede minarse para intentar descubrir carencias de información en la base de conocimiento, o incluso tendencias de interés en los usuarios. Esta información podría ser utilizada en beneficio de los sistemas colaborativos para mejorar sus propios recursos con respecto a las necesidades de información de los usuarios.
- Construir los niveles de metainformación necesarios para implementar técnicas de razonamiento semántico conlleva un esfuerzo considerable. Nuestros resultados parecen indicar que este esfuerzo podría reducirse aplicando técnicas de aprendizaje basadas en razonamiento por analogía del lenguaje y deducción de hipótesis para afrontar expresiones nunca vistas.

Con respecto a la vertiente de ingeniería, este proyecto de Tesis ha dado como fruto varias aplicaciones informáticas de interés comercial. Entre ellas, podríamos destacar *Opinium*, un framework para el Análisis de Opiniones, y *FAQtory*, un framework para la recuperación y gestión de FAQs. Además, varias de las técnicas aquí desarrolladas han sido integradas en aplicaciones comerciales reales o se encuentran en fase de integración en futuras aplicaciones.

Desde el punto de vista de la producción científica, esta Tesis ha dado como resultado la publicación de varios artículos en diversas revistas internacionales y comunicación en congresos de índole científica (capítulo VII).

2 Trabajos Futuros

A lo largo de cada capítulo, hemos ido remarcando los que supondrán nuestros trabajos futuros y líneas de interés a abordar en un futuro. Nos centraremos aquí en remarcar las más inmediatas y de mayor interés desde nuestro punto de vista.

En primer lugar, centraremos nuestra atención en el proyecto que comenzamos durante el periodo de estancia, a fin de explorar nuevas funciones de filtrado para el problema de la Selección de Características en Clasificación de Textos. Nuestra motivación principal viene dada por las conclusiones extraídas en el capítulo II acerca de la correlación positiva y los estudios [TA03, XLL^{+}08, Joa05] en los que se pone de manifiesto la influencia de las métricas de evaluación en el proceso de clasificación. Pensamos que hay bastante camino por recorrer investigando estrategias más sofisticadas en función de las métricas de evaluación.

Actualmente estamos trabajando en un prototipo de *Opinium* para el análisis de comentarios de ámbito político en las redes sociales y otro para el resumen de opinión en encuestas de satisfacción de usuarios que de seguro marcará las líneas de nuestra investigación más cercana. De igual forma, estamos trabajando en incorporar el motor de recuperación de FAQs en el sistema colaborativo comentado en el capítulo IV sección 5.2 e incorporar la funcionalidad de *FAQ clouds* [RMC13a] a *FAQtory*.

En el capítulo V propusimos un método que partiendo de niveles de conocimiento elevados era capaz de mejorar su conocimiento. Nos preguntamos finalmente hasta qué punto esta metodología es dual, es decir, si sería posible utilizar el mismo criterio de razonamiento por analogía para inducir una estructura interna y oculta en recursos de conocimiento inicialmente desestructurados. Esta propuesta, que está relacionada con la generación automática de ontologías y la inferencia gramatical, podría tal vez sacar partido de algunos recursos públicos, como la Wikipedia, para automatizar eficientemente algunas tareas de configuración. Pensamos que en este ámbito podrían ser de utilidad la aplicación de mecanismos de interacción como los propuestos en este trabajo, generalizando convenientemente el concepto de *explicación*.

Chapter VII

List of Publications: Submitted, Published, and Accepted Articles

- A. Moreo, J.L. Castro, V. López, J.M. Zurita, EMD: una metodología de recuperación para sistemas de lenguaje natural basados en casos, XV Congreso Español sobre Tecnologías y Lógica Fuzzy, 2010, February, Pages 217-222, Puntaumbría, Huelva
  - Status: **Published**.
  - Type: National Conference paper
  - Tipo: Artículo de Conferencia, Congreso Nacional
  - Correspondence with: chapter IV, section 2

Abstract: Muchas de las aplicaciones que trabajan con el Lenguaje Natural, como los Sistemas de FAQ retrieval, los Sistemas de Diálogo, los Asistentes Virtuales, etc. pueden ser modelados mediante un enfoque de Razonamiento Basado en Casos (CBR), transformando el problema del Question Answering al de buscar el caso de la base de casos más parecido a la pregunta introducida por el usuario. Gran parte de los modelos propuestos se valen de Palabras Clave para tal propósito. Sin embargo, la elección de este conjunto de palabras no es una tarea trivial, puesto que depende del conocimiento experto y posiblemente, las palabras que lo compongan no sean suficientes de cara a la diferenciación de casos.

- A. Moreo, M. Navarro, J.L. Castro, J.M. Zurita, A high-performance FAQ retrieval method using minimal differentiator expressions, Knowledge-Based Systems, Volume 36, December 2012, Pages 9-20, ISSN 0950-7051, 10.1016/j.knosys.2012.05.015.
    - Status: **Published**.
    - Type: Journal Article.
    - Impact Factor (JCR 2012): 4.104.
    - Subject Category: Computer Science, Artificial Intelligence. Ranking 6 / 114 (Q1).
    - Correspondence with: chapter IV, section 2

Abstract: Case-Based Reasoning (CBR) has proven to be a very useful technique to solve problems in Closed- Domains Question Answering such as FAQ retrieval. Instead of trying to uderstand the question this method consists of retrieving the most similar case (Question/Answer pairs) among all cases by analogy. Keyword comparison criterion or statistical approaches are often used to implement similarity measure. However, those methods present the following disadvantages. On the one side, choosing keywords is an expert-knowledge domain-dependant task that is often performed manually. Furthermore, keyword comparison criterion does not guarantee the total differentiation among cases. On the other side, statistical approaches do not perform with enough information in sentence-level problems and are not interpretable. In order to alleviate these deficiencies we present a new method called the Minimal Differentiator Expressions (MDE) algorithm. This algorithm automatically obtains a set of linguistic patterns (expressions) used to retrieve the most relevant case to the user question. Those patterns present the following advantages: they are composed by the simplest sets of words which permit differentiation among cases and they are easily interpretable.

- A. Moreo, M. Romero, J.L. Castro, J.M. Zurita, FAQtory: A framework to provide high-quality FAQ retrieval systems, Expert Systems with Applications, Volume 39, Issue 14, 15 October 2012, Pages 11525-11534, ISSN 0957-4174, 10.1016/j.eswa.2012.02.130.
  - Status: **Published**.
  - Type: Journal Article.
  - Impact Factor (JCR 2012): 1.854.
  - Subject Category: Computer Science, Artificial Intelligence. Ranking 31 / 114 (Q2).
  - Subject Category: Engineering, Electrical & Electronic. Ranking 56 / 242 (Q1).
  - Subject Category: Operations Research & Management Science. Ranking 13 / 78 (Q1).
  - Correspondence with: chapter IV, section 3

Abstract: To facilitate access to information, companies usually try to anticipate and answer most typical customer's questions by creating Frequently Asked Questions (FAQs) lists. In this scenario, FAQ retrieval is the area of study concerned with recovering the most relevant Question/Answer pairs contained in FAQ compilations. Despite the amount of effort that has been devoted to investigate FAQ retrieval methods, how to create an maintain high quality FAQs has received less attention. In this article, we propose an entire framework to use, create and maintain intelligent FAQs. Usage mining techniques have been developed to take advantage of usage information in order to provide FAQ managers with meaningful information to improve their FAQs. Usage mining techniques include weaknesses detection and knowledge gaps discovery. In this way, the management of the FAQ is no longer directed only by expert knowledge but also by users requirements.

- M. Romero, A. Moreo, J.L. Castro, J.M. Zurita, Using Wikipedia concepts and frequency in language to extract key terms from support documents, Expert Systems with Applications, Volume 39, Issue 18, 15 December 2012, Pages 13480-13491, ISSN 0957-4174, 10.1016/j.eswa.2012.07.011.
  - Status: **Published**.
  - Type: Journal Article.
  - Impact Factor (JCR 2012): 1.854.
  - Subject Category: Computer Science, Artificial Intelligence. Ranking 31 / 114 (Q2).
  - Subject Category: Engineering, Electrical & Electronic. Ranking 56 / 242 (Q1).
  - Subject Category: Operations Research & Management Science. Ranking 13 / 78 (Q1).
  - Correspondence with: none

Abstract: In this paper, we present a new key term extraction system able to handle with the particularities of "support documents". Our system takes advantages of frequency-based and thesaurus-based approaches to recognize two different classes of key terms. On the one hand, it identifies multi-domain key terms of the collection using Wikipedia as knowledge resource. On the other hand, the system extracts specific key terms highly related with the context of a support document. We use the frequency in language as a criterion to detect and rank such terms. To prove the validity of our system we have designed a set of experiment using a Frequently Asked Questions (FAQ) collection of documents. Since our approach is generic, minor modifications should be undertaken to adapt the system to other kind of support documents. The empirical results evidence the validity of our approach.

- M. Romero, A. Moreo, J.L. Castro, A Cloud of FAQ: A Highly-Precise FAQ Retrieval System for the Web 2.0, Knowledge-Based Systems
  - Status: **Accepted**.
  - Type: Journal Article.
  - Impact Factor (JCR 2012): 4.104.
  - Subject Category: Computer Science, Artificial Intelligence. Ranking 6 / 114 (Q1).
  - Reflected in: chapter IV, section 5.2

Abstract: FAQ (Frequency Asked Questions) lists have recently attracted increasing attention for companies and organizations as a way to other a trusted and well-organized source of knowledge. There is thus a need for high-precise and fast methods able to manage large FAQ collections. In this context, we present a new FAQ retrieval system as part of a FAQ exploiting project. Following the growing trends towards Web 2.0, our goal is to provide users with mechanisms to navigate through the domain of knowledge and to facilitate both learning and searching, beyond classic FAQ retrieval algorithms. To this purpose, our system involves two different modules: an efficient and precise FAQ retrieval module, and a tag cloud generation module designed to help users to complete the comprehension of the retrieved information. Empirical results evidence the validity of our approach with respect to a number of state-of-the-art algorithms in terms of the most popular metrics in the field.

- A. Moreo, E.M. Eisman, J.L. Castro, J.M. Zurita, Learning Regular Expressions to Template-based FAQ Retrieval Systems, Knowledge-Based Systems
  - Status: **Submitted** (under second revision).
  - Type: Journal Article.
  - Impact Factor (JCR 2012): 4.104.
  - Subject Category: Computer Science, Artificial Intelligence. Ranking 6 / 114 (Q1).
  - Correspondence with: chapter IV, section 4.

Abstract: Template-based approaches have proven to be one of the most efficient and robustest ways of addressing Question Answering problems. Templates embody the expert's knowledge on the domain and his/her ability to understand and answer questions, but designing these templates may become a complex task since it is usually carried out manually. Although these methods are not automatic, companies may prefer to undertake this solution in order to offer a better service. In this article, we propose a semiautomatic method to reduce the problem of creating templates to that of validate, and possibly modify, a list of proposed templates. In this way, a better trade-off between reliability —the system is still monitored by an expert— and cost is achieved. In addition, updating templates after domain changes becomes easier, human mistakes are reduced, and portability is increased. Our proposal is based on inferring regular expressions that induce the language conveyed by a set of previously collected query reformulations. The main contribution of this work consists of the definition of a suitable optimisation measure that effectively reflects some important aspects of the problem and the theoretical soundness that supports it.

- A. Moreo, J.L. Castro, J.M. Zurita, Towards Portable Natural Language Interfaces based on Human-Computer Interaction, International Journal of Human-Computer Studies
    - Status: **Submitted** (under second revision).
    - Type: Journal Article.
    - Impact Factor (JCR 2012): 1.415.
    - Subject Category: Computer Science, Cybernetics. Ranking 7 / 21 (Q2).
    - Correspondence with: chapter V.

Abstract: Natural Language Interfaces allow non-technical people to access information stored in Knowledge Bases keeping them unaware of the particular structure of the model or the underlying formal query language. Although earlier research in the field was devoted to improve the performance for specific Knowledge Bases, adapting the system to new domains usually entailed much effort. Thus, how to bring Portability to NLI received renewed interest. In this article, we investigate how human-computer interactions could serve to assist the expert in porting the system so as to improve its retrieval performance. Our method HITS is based on a novel grammar learning algorithm combined with language acquisition techniques that exploits structural analogies. The learner (system) is able to engage the teacher (expert) with clarification dialogues to validate conjectures (hypotheses and deductions) about the language. Our method presents the following advantages: (i) because the interactions are mainly directed by the system, the customization effort is substantially alleviated, (ii) the teacher-learner model brings the expert the opportunity to continuously improve the *habitability* of the system —the types of questions the system can deal with are not delimited in advance—, and (iii) the system 'reasons' about its previous knowledge to deal with unseen questions.

- A. Moreo, J.L. Castro, J.M. Zurita, Handling Context in Lexicon-Based Sentiment Analysis, Advances in Computational Intelligence, Communications in Computer and Information Science, Springer Berlin Heidelberg, Computer Science, 2012, vol 298, Pages 245-254
  - Status: **Published**.
  - Type: International Conference paper.
  - Conference: IPMU 2012, Catania, Italia.
  - Correspondence with: chapter III, section 2.5.

Abstract: Internet has evolved to the Web 2.0 allowing people all around the world to interact with each other and to speak freely about any relevant topic. This kind of user-generated content represents an unstructured knowledge source of undeniable interest in decision-making for both common people and organizations. However, given the high volume of data stored in the Web, performing a manual analysis of this information becomes (practically) impossible. In such a context, Sentiment Analysis aims to automatically summarize opinions expressed in texts providing understandable sentiment reports. However, the computational analysis of opinions is inevitably affected by inherent difficulties presented in natural language. Ambiguity, anaphora, and ellipsis, are examples of context-dependant problems attached to natural language. In this paper, we present a lexicon-based algorithm dealing with sentiment analysis that takes advantage of context analysis to provide sentiment summarization reports.

- A. Moreo, M. Romero, J.L. Castro, J.M. Zurita, Lexicon-based Comments-oriented News Sentiment Analyzer system, Expert Systems with Applications, Volume 39, Issue 10, August 2012, Pages 9166-9180, ISSN 0957-4174, 10.1016/j.eswa.2012.02.057.
    - Status: **Published**.
    - Type: Journal Article.
    - Impact Factor (JCR 2012): 1.854.
    - Subject Category: Computer Science, Artificial Intelligence. Ranking 31 / 114 (Q2).
    - Subject Category: Engineering, Electrical & Electronic. Ranking 56 / 242 (Q1).
    - Subject Category: Operations Research & Management Science. Ranking 13 / 78 (Q1).
    - Correspondence with: chapter III, section 2.

Abstract: Thanks to the technological revolution that has accompanied the Web 2.0, users are able to interact intensively on the Internet, as reflected in social networks, blogs, forums, etc. In these scenarios, users can speak freely on any relevant topic. However, the high volume of user-generated content makes a manual analysis of this discourse unviable. Consequently, automatic analysis techniques are needed to extract the opinions expressed in users' comments, given that these opinions are an implicit barometer of unquestionable interest for a wide variety of companies, agencies, and organisms. We thus propose a lexicon-based Comments-oriented News Sentiment Analyzer (LCN-SA), which is able to deal with the following: (a) the tendency of many users to express their views in non-standard language; (b) the detection of the target of users' opinions in a multi-domain scenario; (c) the design of a linguistic modularized knowledge model with low-cost adaptability. The system proposed consists of an automatic Focus Detection Module and a Sentiment Analysis Module capable of assessing user opinions of topics in news items. These modules use a taxonomy-lexicon specifically designed for news analysis. Experiments show that the results obtained thus far are extremely promising.

- M. Romero, A. Moreo, J.L. Castro, Collaborative System for Learning based on Questionnaires and Tasks, *to appear in* 4th International Conference on EUropean Transnational Education (ICEUTE'13), 2013, September, Salamanca, Spain
    - Status: **Accepted**.
    - Type: International Conference paper.
    - Conference: ICEUTE 2013, Salamanca, Spain.
    - Reflected in: chapter IV, section 5.3.

Abstract: Virtual Learning Environments allow to improve the learning interactivity in a collaborative scenario where the learning contents are proposed and accessed by both learners and teachers. In this work, we present CSLQT, a new Collaborative System for Learning based on Questionnaires and Tasks. This system is independent of any course structure or content, and provide users with functionalities to create, review, and evaluate new knowledge resources through questions and tasks. The benefits are two-fold: teachers are released from the tedious task of creating all resources, and students are encouraged to gain the necessary knowledge background before creating any new content. Additionally, a Fuzzy controller generates exams satisfying a customized set of objectives, that could be used for evaluation or auto-evaluation purposes. Our experiences with the system in real courses of the University of Granada indicate the tool is actually useful to improve the learning process.

Appendices

1 Minimal Differentiator Expressions: Example trace

Trace of the MDE algorithm for sentence *What type of company is this?*, considering α=2)

MDEs	**OPENED**
Initialization	
[]	**[{what} {type} {of} {company} {is} {this}]**
Checking {what} → *false* (explore combinations of "what")	
[]	[{type} {of} {company} {is} {this} **{what type} {what of} {what company} {what is} {what this}**]
Checking {type} → *true*	
[{type}]	[{of} {company} {is} {this} {what type} {what of} {what company} {what is} {what this}]
Checking {of} → *false*	
[{type}]	[{company} {is} {this} {what type} {what of} {what company} {what is} {what this} **{of company} {of is} {of this}**]
Checking {company} → *false*	
[{type}]	[{is} {this} {what type} {what of} {what company} {what is} {what this} {of company} {of is} {of this} **{company is} {company this}**]
Checking {is} → *false*	
[{type}]	[{this} {what type} {what of} {what company} {what is} {what this} {of company} {of is} {of this} {company is} {company this} **{is this}**]
Checking {this} → *false*	
[{type}]	[{what type} {what of} {what company} {what is} {what this} {of company} {of is} {of this} {company is} {company this} {is this}] ∪ **[ ]**
Checking {what type} → *false*, α-bound reached, no further exploration	
[{type}]	[{what of} {what company} {what is} {what this} {of company} {of is} {of this} {company is} {company this} {is this}]
Checking {what of} → *false*, α-bound reached, no further exploration	
After some steps...	
[{type}]	[{company is} {company this} {is this}]
Checking {company is} → *true*	
[{type}, {company is}]	[{company this} {is this}]
Checking {company this} → *true*	
[{type}, {company is}, {company this}]	[{is this}]
Checking {is this} → *false*, α-bound reached, no further exploration	
[{type}, {company is}, {company this}]	[]
End	

2 Calculating $|L_{\Sigma^n}(r)|$, an Example

The following example exposes the calculus of $|L_{\Sigma^n}(r)|$, being

$$r = *\,((telephone|fax)\ number|email) * (director|manager)\ *$$

First step consists of creating the equivalent DFA (see Figure .1, where Σ represents the finite set of symbols, and labels represent transitions).

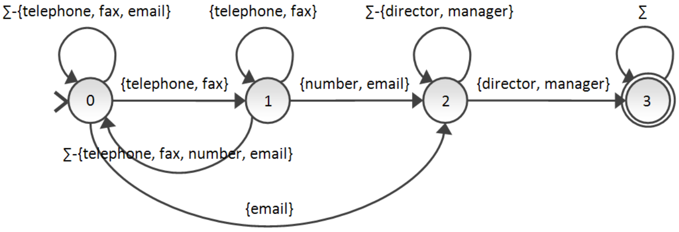


Figure .1: DFA equivalent to regular expression.

Second step consists of obtaining the equivalent directed graph. In this case, labels represent the number of transitions between states (see Figure .2).

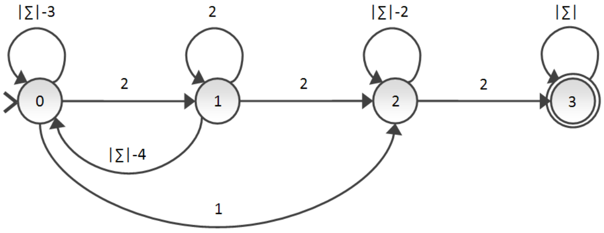


Figure .2: Directed Graph equivalent to DFA.

Once the directed graph is calculated, the following step is to calculate its adjacency matrix:

$$m = \begin{bmatrix} |\Sigma|-3 & 2 & 1 & 0 \\ |\Sigma|-4 & 2 & 2 & 0 \\ 0 & 0 & |\Sigma|-2 & 2 \\ 0 & 0 & 0 & |\Sigma| \end{bmatrix} \tag{1}$$

Finally, to illustrate the calculus, we are considering $|\Sigma| = 10$ and $n = 5$. Note that the initial state is $s = \{0\}$, and the final state is $f = \{3\}$. Since the size of $|\Sigma|^n = 111111$, the result is:

$$generalization(r, n) = \lg_2\left(\frac{2 + 58 + 1078 + 16466}{111111} + 1\right) = 0.2121 \tag{2}$$

3 Calculating $structuralsimplicity(r)$, an Example

This example illustrates the structural complexity calculus for the following regular expression

$$r = *(email|fax) * director*$$

It should be taken into account that wildcard '*' acts as an additional symbol in our procedure. The equivalent NFA obtained through the Thompson's algorithm is shown in Figure .3.

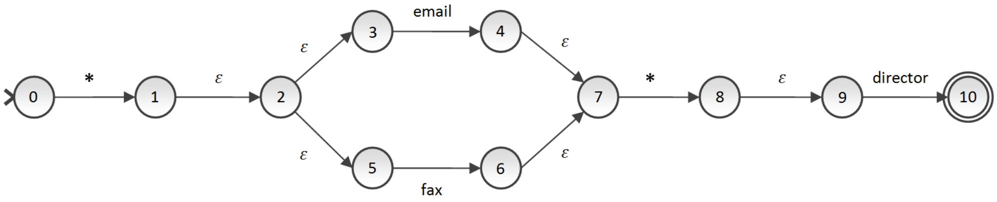


Figure .3: Equivalent NFA obtained using Thompson's algorithm

Then, the calculus of the structural complexity is:

$$structuralsimplicity(r) = \frac{1}{11 + 11} = 0.045 \qquad (3)$$

4 Entity Relationship Diagram of Dataset Domains

Figures .4, .5, and .6 show the Entity-Relationship diagram for Geobase, Jobdata, and Restbase domain, respectively.

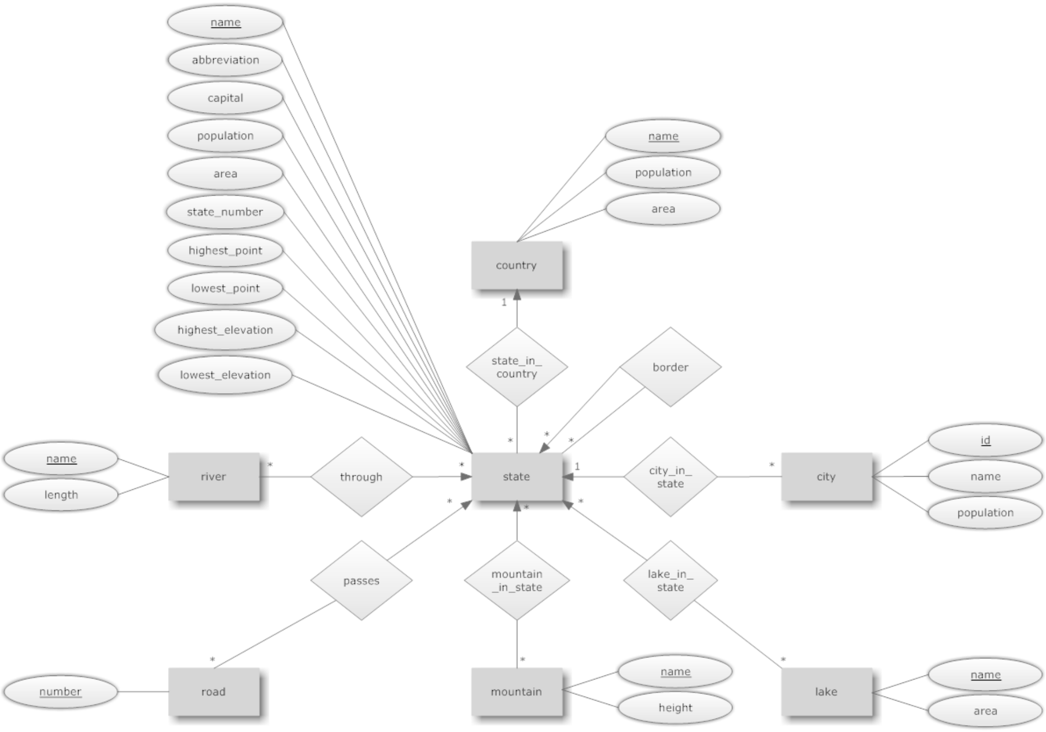


Figure .4: Entity Relationship Diagram of Geobase Domain.

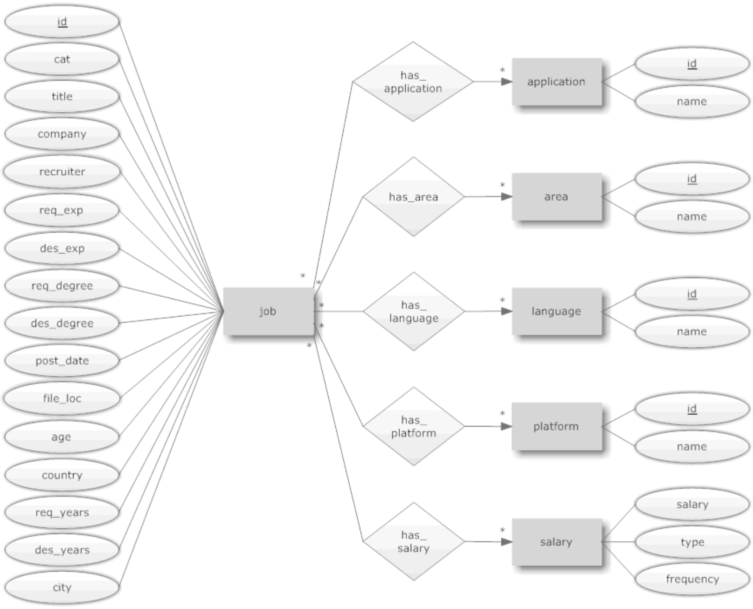


Figure .5: Entity Relationship Diagram of Jobdata Domain.

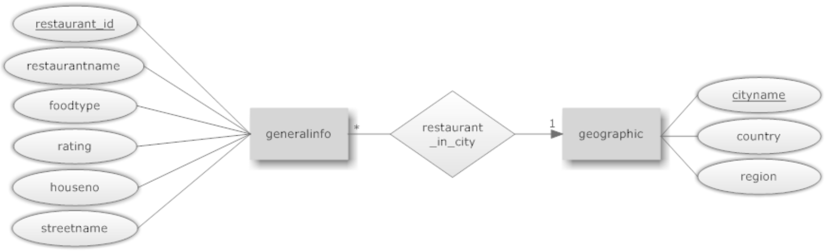


Figure .6: Entity Relationship Diagram of Restbase Domain.

5 From DB to Ontology. Technical details.

How to obtain a suited ontology describing the semantic level of a given database is not a trivial problem. Recent work in NLI usually assume the KB —not necessarily a database— was entirely mapped in the ontology in advance. Other approaches simply consider this ontology was previously set up. Even considering that it is not among the scope of this work to deal in deep with this issue, we present here some technical guidance that helped us to obtain automatically the ontologies from the metadata of the database. Readers interested in more sophisticated methods are referred to [GC07, CGY07].

This procedure was coded in Java JDK 1.6, using Jena[1] to create the ontology, JDBC[2] to stablish the connection with the database, and MySQL[3] as the DBMS.

First, all categories such as Entity, Relation, Attribute, and so on, are initialized using Jena. Taking Geobase as an example, all tables involved in the database schema could be retrieved using:

```
SELECT ‘TABLE_NAME‘ FROM ‘TABLES‘ WHERE ‘TABLE_SCHEMA‘="geobase"
```

All primary keys from a given table could be retrieved by inspecting 'KEY_COLUMN_USAGE' table. For example, to obtain the primary keys from geobase.border:

```
SELECT ‘COLUMN_NAME‘ FROM ‘KEY_COLUMN_USAGE‘
WHERE ‘CONSTRAINT_NAME‘="PRIMARY" and ‘CONSTRAINT_SCHEMA‘="geobase" and
      ‘TABLE_SCHEMA‘="geobase" and ‘TABLE_NAME‘="border"
```

The following SQL command obtains the foreign key of a given primary key (geobase.border.state1 in this example). Note that the only restriction to bear in mind is that tables should be created with the InnoDB engine.

```
SELECT ‘REFERENCED_TABLE_SCHEMA‘,‘REFERENCED_TABLE_NAME‘,‘REFERENCED_COLUMN_NAME‘
FROM ‘KEY_COLUMN_USAGE‘
WHERE ‘CONSTRAINT_NAME‘<>"PRIMARY" and ‘CONSTRAINT_SCHEMA‘="geobase" and
      ‘TABLE_SCHEMA‘="geobase" and ‘TABLE_NAME‘="border" and ‘COLUMN_NAME‘="state1"
```

All columns, including primary ones, could be obtained easily using this command:

```
SELECT * FROM ‘COLUMNS‘
WHERE ‘TABLE_SCHEMA‘="geobase" and ‘TABLE_NAME‘="state"
```

To discern only the primary keys, the following command is helpful:

```
SELECT * FROM ‘COLUMNS‘
WHERE ‘TABLE_SCHEMA‘="geobase" and ‘TABLE_NAME‘="state" and ‘COLUMN_KEY‘="PRI"
```

[1] http://jena.apache.org/documentation/ontology/index.html

[2] http://docs.oracle.com/javase/tutorial/jdbc/index.html

[3] http://www.mysql.com/

To differentiate Relational-tables from Entity-tables, one could simply inspect whether the table contains foreign keys or not.

Above mentioned snippets suffice to create a code to automatically initialize the ontologies we used in our experiments, including the necessary information for the *Inspector* methods.

Bibliography

[ACC$^+$92] Alshawi H., Carter D., Crouch R., Pulman S., Rayner M., y Smith A. (1992) Clare: A contextual reasoning and cooperative response framework for the core language engine.

[ACS08] Abbasi A., Chen H., y Salem A. (2008) Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. *ACM Transactions on Information Systems (TOIS)* 26(3): 12.

[AG05] Adamic L. A. y Glance N. (2005) The political blogosphere and the 2004 u.s. election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*, pp. 36–43. Chicago, Illinois.

[AGI07] Archak N., Ghose A., y Ipeirotis P. G. (August 2007) Show me the money!: deriving the pricing power of product features by mining consumer reviews. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining.* USA, San Jose, California.

[AHBLGS$^+$12] Alario-Hoyos C., Bote-Lorenzo M. L., Gómez-Sánchez E., Asensio-Pérez J. I., Vega-Gorgojo G., y Ruiz-Calleja A. (2012) Glue!: an architecture for the integration of external tools in virtual learning environments. *Computers & Education* .

[Aka97] Akama S. (1997) Logic, language and computation. *Kulwer Academic publishers* pp. 7–11.

[All95] Allen J. (1995) *Natural Language Understanding.* The Benjamin/Cummings Publishing Company, Inc, 2nd edition edition. ISBN 0-8053-0334-0.

[ALM$^+$03] Avancini H., Lavelli A., Magnini B., Sebastiani F., y Zanoli R. (2003) Expanding domain-specific lexicons by term categorization. In *Proceedings of the 2003 ACM symposium on Applied computing*, SAC '03, pp. 793–797. ACM, New York, NY, USA.

[ALSZ06] Avancini H., Lavelli A., Sebastiani F., y Zanoli R. (May 2006) Automatic expansion of domain-specific lexicons by term categorization. *ACM Trans. Speech Lang. Process.* 3(1): 1–30.

[ALZ04] Avancini H., Lavelli A., y Zanoli R. (2004) Automatic expansion of domainspecific lexicons by term categorization. *ACM Transactions on Speech and Language Processing* 3: 2006.

[Ang87] Angluin D. (November 1987) Learning regular sets from queries and counterexamples. *Inf. Comput.* 75(2): 87–106.

[ARSX03] Agrawal R., Rajagopalan S., Srikant R., y Xu Y. (May 2003) Mining newsgroups using networks arising from social behavior. In *Proceedings of the 12th international conference on World Wide Web.* Hungary, Budapest.

[ART93] Androutsopoulus I., Ritchie G., y Thanish P. (1993) Masque/sql, an efficient and portable natural language query interface for relational databases. In *Proc. 6th International Conference on Industrial & Engineering Applications of Artificial Intelligence and Expert Systems*, pp. 327–330. Edinburgh, UK.

[ART95] Androutsopoulos I., Ritchie G. D., y Thanish P. (1995) Natural language interfaces to databases - an introduction. *Natual Language Engineering* 1(1): 29–81.

[AS05] Andrenucci A. y Sneiders E. (2005) Automated question answering: Review of the main approaches. In *in the Third International Conference on Information Technology and Applications*, volumen I, pp. 514–519.

[AV02] Adriaans P. y Vervoort M. (2002) The emile 4.1 grammar induction toolbox. In *Grammatical Inference: Algorithms and Applications*, volumen 2484 of *Lecture Notes in Artificial Intelligence*, pp. 293–295.

[AZ12] Aggarwal C. C. y Zhai C. (2012) A survey of text clustering algorithms. In *Mining Text Data*, pp. 77–128. Springer.

[Bat89] Bates M. (1989) Rapid porting of the parlance natural language interface. In *Proc. Workshop on Speech and Natural Language*, pp. 83–88.

[Bat97] Bateman J. A. (1997) The theoretical status of ontologies in natural language processing. *CoRR* cmp-lg/9704010.

[BB75] Brown J. y Burton R. (1975) Multiple representations of knowledge for tutorial reasoning. *In Representation and Understanding, D.G. Bobiow and A. Collins, Eds., Academic Press, New York* pp. 311–349.

[BBMS96] Bagnasco C., Bresciani P., Magnini B., y Strapparava C. (1996) Natural language interpretation for public administration database querying in the tamic demonstrator. In *Proc. 2nd International Workshop on Applications of Natural Language to Information Systems.* Amsterdam, The Netherlands.

[BCC+00] Berger A., Caruana R., Cohn D., Freitag D., y Mittal V. (2000) Bridging the lexical chasm: Statistical approaches to answer-finding. *SIGIR Forum (ACM Special Interest Group on Information Retrieval)* pp. 192–199.

[BCM05] Buitelaar P., Cimiano P., y Magnini B. (2005) *Ontology learning from text: methods, evaluation and applications*, volumen 123. IOS press.

[BCM+11] Buitelaar P., Cimiano P., McCrae J., Montiel-Ponsoda E., y Declerck T. (11 2011) Ontology lexicalisation: The lemon perspective. In Slodzian M., Valette M., Aussenac-Gilles N., Condamines A., Hernandez N., y Rothenburger B. (Eds.) *Workshop Proceedings of the 9th International Conference on Terminology and Artificial Intelligence. International Conference on Terminology and Artificial Intelligence (TIA-11), 9th, November 8-10, Paris, France*, pp. 33–36. INALCO, INALCO, Paris.

[BCP+07] Benamara F., Cesarano C., Picariello A., Reforgiato D., y Subrahmanian V. (2007) Sentiment analysis: Adjectives and adverbs are better than adjectives alone. In OMNIPRESS (Ed.) *Proceedings of ICWSM 2007*. Boulder, Colorado.

[BDDL+12] Bartoli A., Davanzo G., De Lorenzo A., Mauri M., Medvet E., y Sorio E. (2012) Automatic generation of regular expressions from examples with genetic programming. In *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference companion*, GECCO Companion '12, pp. 1477–1478. ACM, New York, NY, USA.

[BDSV91] Binot J., Debille L., Sedlock D., y Vandecapelle B. (1991) Natural language interfaces: A new philosophy. *SunExpert, Magazine* .

[BES10] Baccianella S., Esuli A., y Sebastiani F. (2010) Feature selection for ordinal regression. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, SAC '10, pp. 1748–1754. ACM, New York, NY, USA.

[BGNV10] Bex G. J., Gelade W., Neven F., y Vansummeren S. (September 2010) Learning deterministic regular expressions for the inference of schemas from xml data. *ACM Trans. Web* 4(4): 14:1–14:32.

[BHK+97] Burke R., Hammond K., Kulyukin V., Lytinen S., Tomuro N., y Schoenberg S. (1997) Question answering from frequently asked question files: Experiences with the faq finder system. *AI Magazine* 18(2): 57–66.

[BKGK05] Bernstein A., Kaufmann E., Göhring A., y Kiefer C. (2005) Querying ontologies: A controlled english interface for end-users. In Gil Y., Motta E., Benjamins V., y Musen M. (Eds.) *The Semantic Web-ISWC 2005*, volumen 3729 of *Lecture Notes in Computer Science*, pp. 112–126. Springer Berlin Heidelberg.

[BKK05] Bernstein A., Kaufmann E., y Kaiser C. (2005) Querying the semantic web with ginseng: A guided input natural language search engine. In *In: 15th Workshop on Information Technologies and Systems, Las Vegas, NV*, pp. 112–126.

[BLB+01] Brill E., Lin J., Banko M., Dumais S., y Ng A. (2001) Data-intensive question answering. In *In Proceedings of the Tenth Text REtrieval Conference (TREC*, pp. 393–400.

[BM07] Bloehdorn S. y Moschitti A. (2007) Exploiting structure and semantics for expressive text kernels. In *In Proceeding of the Conference on Information Knowledge and Management*. Lisbon, Portugal.

[BNST06] Bex G. J., Neven F., Schwentick T., y Tuyls K. (2006) Inference of concise dtds from xml data. In *Proceedings of the 32nd international conference on Very large data bases*, VLDB '06, pp. 115–126. VLDB Endowment.

[BP11] Bhargava S. y Purohit G. N. (February 2011) Article: Construction of a minimal deterministic finite automaton from a regular expression. *International Journal of Computer Applications* 15(4): 16–27. Published by Foundation of Computer Science.

[BPM+11] Billiet C., Pons J. E., Matthé T., De Tré G., y Capote O. P. (2011) Bipolar fuzzy querying of temporal databases. In *Flexible Query Answering Systems*, pp. 60–71. Springer.

[BS01] Bickart B. y Schindler R. (2001) Internet forums as infuential sources of consumer information. *Journal of Interactive Marketing* 15(3): 31–40.

[BS03] Boldasov M. y Sokolova G. (2003) Qgen - generation module for the register restricted in-base system. In *Computational Linguistics and Intelligent Text Processing, 4th International Conference*, pp. 465–476.

[BSA00] Busemann S., Schmeier S., y Arens R. (2000) Message classification in the call center. In *Proceedings of the Sixth Conference on Applied Natural Language Processing, ACL (2000)*, pp. 158–165. Seattle, Washington.

[Bur76] Burton R. (1976) Semantic grammar: An engineering technique for constructing natural language understanding systems. *BBN Rep. 3453, Bolt, Beranek, and Newman, Boston, Mass.* .

[BV04] Baroni M. y Vegnaduzzo S. (2004) Identifying subjective adjectives through web-based mutual information. In *In Proceedings of the 7th German Conference on Natural Language Processing (KONVENS'04)*, pp. 613–619.

[BZHZ10] Bu F., Zhu X., Hao Y., y Zhu X. (2010) Function-based question classification for general qa. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pp. 1119–1128. Association for Computational Linguistics, Stroudsburg, PA, USA.

[CBC06] Choi Y., Breck E., y Cardie C. (2006) Joint extraction of entities and relations for opinion recognition. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, pp. 431–439. Sydney.

[CC07] Carmona P. y Castro J. (2007) An ant colony optimization plug-in to enhance the interpretability of fuzzy rule bases with exceptions. In *Analysis and Design of Intelligent Systems using Soft Computing Techniques*, pp. 436–444. Springer.

[CC08a] Carmona P. y Castro J. (2008) An improved aco based plug-in to enhance the interpretability of fuzzy rule bases with exceptions. In Dorigo M., Birattari M., Blum C., Clerc M., Stützle T., y Winfield A. (Eds.) *Ant Colony Optimization and Swarm Intelligence*, volumen 5217 of *Lecture Notes in Computer Science*, pp. 13–24. Springer Berlin Heidelberg.

[CC08b] Choi Y. y Cardie C. (2008) Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '08)*, pp. 793–801. Stroudsburg, PA, USA.

[CCSZ99] Castro J., Castro-Schez J., y Zurita J. (1999) Learning maximal structure rules in fuzzy logic for knowledge acquisition in expert systems. *Fuzzy Sets and Systems* 101(3): 331 – 342.

[CCSZ01] Castro J., Castro-Schez J., y Zurita J. (2001) Use of a fuzzy machine learning technique in the knowledge acquisition process. *Fuzzy Sets and Systems* 123(3): 307 – 320.

[CCV$^+$07] Casellas N., Casanovas P., Vallbé J.-J., Poblet M., Blázquez M., Contreras J., López-Cobo J.-M., y Benjamins V. R. (2007) Semantic enhancement for legal information retrieval: Iuriservice performance. In *Proceedings of the 11th international conference on Artificial intelligence and law*, ICAIL '07, pp. 49–57. ACM, New York, NY, USA.

[CCZ04a] Carmona P., Castro J., y Zurita J. (2004) Friwe: fuzzy rule identification with exceptions. *Fuzzy Systems, IEEE Transactions on* 12(1): 140–151.

[CCZ04b] Carmona P., Castro J., y Zurita J. (2004) Learning maximal structure fuzzy rules with exceptions. *Fuzzy Sets and Systems* 146(1): 63 – 77. Selected Papers from {EUSFLAT} 2001.

[Cet07] Cetinkaya A. (2007) Regular expression generation through grammatical evolution. In *Proceedings of the 2007 GECCO conference companion on Genetic and evolutionary computation*, GECCO '07, pp. 2643–2646. ACM, New York, NY, USA.

[ÇG08] Çıltık A. y Güngör T. (2008) Time-efficient spam e-mail filtering using n-gram models. *Pattern Recognition Letters* 29(1): 19–33.

[CGY07] Cullot N., Ghawi R., y Yétongnon K. (2007) DB2OWL : A Tool for Automatic Database-to-Ontology Mapping. In Ceci M., Malerba D., Tanca L., Ceci M., Malerba D., y Tanca L. (Eds.) *SEBD*, pp. 491–494.

[CHH07] Cimiano P., Haase P., y Heizmann J. (2007) Porting natural language interfaces between domains: an experimental user study with the orakel system. In *Proceedings of the 12th international conference on Intelligent user interfaces*, IUI '07, pp. 180–189. ACM, New York, NY, USA.

[CHWC05] C.-H. Wu J.-F. Y. y Chen M.-J. (2005) Domain-specific faq retrieval using independent aspects. *ACM Transactions on Asian Language Information Processing* 4(1): 1–17.

[CKC07] Cui H., Kan M.-Y., y Chua T.-S. (April 2007) Soft pattern matching models for definitional question answering. *ACM Transactions of Information Systems* 25(2).

[CM06] Chevalier J. A. y Mayzlin D. (2006) The effect of word of mouth on sales: Online book reviews. *Journal of Marketing Research* pp. 345–354.

[CMP$^+$93] Cercone N., Mcfetridge P., Popowish F., Fass D., Groeneboer C., y Hall G. (1993) The system x natural language interface. design, implementation and evaluation. Technical report, Centre for System Science, Simon Fraser University, British Columbia, Canada.

[CNZ05] Carenini G., Ng R., y Zwart E. (2005) Extracting knowledge from evaluative text. In *Proceedings of the International Conference on Knowledge Capture (K-CAP'05)*, pp. 11–18. Banff, Alberta, Canada.

[Cod74] Codd E. (1974) Seven steps to rendezvous with the casual user. *In Data Base Management, J.W. Klimbie and K.I. Koffeman, Eds.* pp. 179–200.

[CRM07] Camacho D. y R.-Moreno M. D. (March 2007) Dynjaq: An adaptive and flexible dynamic faq system: Research articles. *Int. J. Intell. Syst.* 22(3): 303–318.

[CV95] Cortes C. y Vapnik V. (1995) Support-vector networks. *Machine Learning* 20(3): 273–297.

[CYC$^+$96] Chu W., Yang H., Chiang K., Minock M., Chow G., y Larson C. (1996) Cobase: A scalable and extensible cooperative information system. *Journal of Intelligent Information Systems* 6: 223–259.

[CZ10] Chen H. y Zimbra D. (June 2010) Ai and opinion mining. *IEEE Intelligent Systems* 25(3): 74–80.

[DA07] Devitt A. y Ahmad K. (2007) Sentiment polarity identification in financial news: A cohesion-based approach. In Press A. (Ed.) *Proc. 45th Ann. Meeting Assoc.*, pp. 984–991. Computational Linguistics.

[DAC10] Damljanovic D., Agatonovic M., y Cunningham H. (2010) H.: Natural language interfaces to ontologies: Combining syntactic analysis and ontology-based lookup through the user interaction. In *In: Proceedings of the 7th Extended Semantic Web Conference (ESWC 2010). Lecture Notes in Computer Science.* SpringerVerlag.

[DAC12] Damljanovic D., Agatonovic M., y Cunningham H. (2012) Freya: an interactive way of querying linked data using natural language. In *Proceedings of the 8th international conference on The Semantic Web*, ESWC'11, pp. 125–138. Springer-Verlag, Berlin, Heidelberg.

[DB09] Damljanović D. y Bontcheva K. (2009) Towards enhanced usability of natural language interfaces to knowledge bases. In Devedžić V., Gašević D., Sharda R., y Voß S. (Eds.) *Web 2.0 and Semantic Web*, volumen 6 of *Annals of Information Systems*, pp. 105–133. Springer US.

[DCFFCGGPP06] Delgado Calvo-Flores M., Fajardo Contreras W., Gibaja Galindo E., y Perez-Perez R. (2006) Xkey: A tool for the generation of identification keys. *Expert Systems with Applications* 30(2): 337–351.

[DCFSSV00] Delgado Calvo-Flores M., Sánchez D., Serrano J. M., y Vila M. A. (2000) A survey of methods to evaluate quantified sentences. *Mathware & soft computing* 7(2): 149–158.

[Del06] Delort J. (2006) Identifying commented passages of documents using implicit hyperlinks. In *Proceedings of HYPERTEXT'06*, pp. 89–98. Odense, Denmark.

[Den08] Denecke K. (2008) Using sentiwordnet for multilingual sentiment analysis. In *Proceedings of the IEEE 24th International Conference on Data Engineering. Workshop (ICDEW 2008)*, pp. 507–512. IEEE Computer Society.

[Den09] Denecke K. (2009) Are sentiwordnet scores suited for multi-domain sentiment classification? In *International Conference on Digital Information Management.*

[DLY08] Ding X., Liu B., y Yu P. S. (2008) A holistic lexicon-based approach to opinion mining. In *Proceedings of the Conference on Web Search and Web Data Mining (WSDM'08)*, pp. 231–239.

[DM07] Das D. y Martins A. F. (2007) A survey on automatic text summarization. *Literature Survey for the Language and Statistics II course at CMU* 4: 192–195.

[DNML10] Danescu-Niculescu-Mizil C. y Lee L. (2010) Dont have a clue? unsupervised co-learning of downward-entailing operators. In *Proceedings of the ACL Short Papers*, pp. 247–252.

[DPHS98] Dumais S., Platt J., Heckerman D., y Sahami M. (1998) Inductive learning algorithms and representations for text categorization. In *Proceedings of the seventh international conference on Information and knowledge management*, pp. 148–155. ACM.

[DR97] Daniels J. J. y Rissland E. L. (1997) What you saw is what you want: Using cases to seed information retrieval. In *Proceedings of the Second International Conference on Case-Based Reasoning Research and Development*, ICCBR '97, pp. 325–336. Springer-Verlag, London, UK, UK.

[DSS$^+$02] Dillenbourg P., Schneider D., Synteta P., *et al.* (2002) Virtual learning environments. In *Proceedings of the 3rd Hellenic Conference'Information & Communication Technologies in Education'*, pp. 3–18.

[DSS11] Dalianis H., Sjöbergh J., y Sneiders E. (2011) Comparing manual text patterns and machine learning for classification of e-mails for automatic answering by a government agency. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6609 LNCS(PART 2): 234–243.

[DW07] Dalmas T. y Webber B. (2007) Answer comparison in automated question answering. *Journal of Applied Logic* 5(1): 104 – 120. Questions and Answers: Theoretical and Applied Perspectives.

[DYC08] D.-Y Chiu Y.-C Pan W.-C. C. (2008) Using rough set theory to construct e-learning faq retrieval infrastructure. In *Proceedings of the 1st IEEE International Conference on Ubi-Media Computing and Workshops*, U-Media2008, pp. 547–552.

[DZC10] Dang Y., Zhang Y., y Chen H. (2010) A lexicon-enhanced method for sentiment classification: An experiment on online product reviews. *IEEE Intelligent Systems* 25(4).

[EFS08] Esuli A., Fagni T., y Sebastiani F. (August 2008) Boosting multi-label hierarchical text categorization. *Inf. Retr.* 11(4): 287–313.

[ELC09] Eisman E. M., López V., y Castro J. L. (2009) Controlling the emotional state of an embodied conversationalagent with a dynamic probabilistic fuzzy rules based system. *Expert Systems with Applications* 36(6): 9698–9708.

[ELC12] Eisman E. M., López V., y Castro J. L. (February 2012) A framework for designing closed domain virtual assistants. *Expert Syst. Appl.* 39(3): 3135–3144.

[ES05] Esuli A. y Sebastiani F. (November 2005) Determining the semantic orientation of terms through gloss classification. In *Proceedings of the 14th ACM international conference on Information and knowledgemanagement*. Bremen, Germany.

[ES06a] Esuli A. y Sebastiani F. (2006) Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of the 5th Conference on Language Resources and Evaluation (LREC)*. Genova, Italy.

[ES06b] Esuli A. y Sebastiani F. (2006) Sentiwordnet: A publicly available lexical resource for opinion mining. In *In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC'06)*, pp. 417–422.

[Fer09] Fernau H. (April 2009) Algorithms for learning regular expressions from positive data. *Inf. Comput.* 207(4): 521–541.

[FKX+07] Frank A., Krieger H.-U., Xu F., Uszkoreit H., Crysmann B., Jörg B., y Schäfer U. (2007) Question answering from structured knowledge sources. *Journal of Applied Logic* 5(1): 20 – 48.

[FM07] Fraser A. y Marcu D. (September 2007) Measuring word alignment quality for statistical machine translation. *Comput. Linguist.* 33(3): 293–303.

[For03] Forman G. (March 2003) An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.* 3: 1289–1305.

[For04] Forman G. (2004) A pitfall and solution in multi-class feature selection for text classification. In *Proceedings of the twenty-first international conference on Machine learning*, ICML '04, pp. 38–. ACM, New York, NY, USA.

[FS95] Freund Y. y Schapire R. E. (1995) A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of the Second European Conference on Computational Learning Theory*, EuroCOLT '95, pp. 23–37. Springer-Verlag, London, UK, UK.

[FS96] Freund Y. y Schapire R. E. (1996) Experiments with a New Boosting Algorithm. In *International Conference on Machine Learning*, pp. 148–156.

[GAMP87] Grosz B. J., Appelt D. E., Martin P. A., y Pereira F. C. N. (May 1987) Team: an experiment in the design of transportable natural-language interfaces. *Artif. Intell.* 32(2): 173–243.

[GBMAHS02] Guerrero Bote V. P., Moya Anegón F. d., y Herrero Solana V. (2002) Document organization using kohonen's algorithm. *Information processing & management* 38(1): 79–89.

[GC07] Ghawi R. y Cullot N. (2007) Database-to-Ontology Mapping Generation for Semantic Interoperability. In *Third International Workshop on Database Interoperability (InterDB 2007)*.

[GGR+00] Garofalakis M., Gionis A., Rastogi R., Seshadri S., y Shim K. (May 2000) Xtract: a system for extracting document type descriptors from xml documents. *SIGMOD Rec.* 29(2): 165–176.

[GHW07] G. Hu D. Liu Q. L. y Wang R.-H. (2007) Supervised learning approach to optimize ranking function for chinese faq-finder. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 4426 LNAI: 531–538.

[GLP+10] Gunawardena T., Lokuhetti M., Pathirana N., Ragel R., y Deegalla S. (2010) An automatic answering system with template matching for natural language questions. In *Proceedings of the 2010 5th International Conference on Information and Automation for Sustainability, ICIAfS 2010*, pp. 353–358.

[GM06] Ge R. y Mooney R. J. (2006) Discriminative reranking for semantic parsing. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (COLING/ACL 2006)*, pp. 263–270.

[Gol67] Gold M. E. (1967) Language identification in the limit. *Information and Control* 10(5): 447–474.

[Gol78] Gold M. E. (1978) Complexity of automaton identification from given data. *Information and Control* 37(3): 302 – 320.

[Gru93] Gruber T. R. (June 1993) A translation approach to portable ontology specifications. *Knowl. Acquis.* 5(2): 199–220.

[GSS00] Galavotti L., Sebastiani F., y Simi M. (2000) Experiments on the use of feature selection and negative evidence in automated text categorization. In *Proceedings of the 4th European Conference on Research and Advanced Technology for Digital Libraries*, ECDL '00, pp. 59–68. Springer-Verlag, London, UK, UK.

[GSS07] Godbole N., Srinivasaiah M., y Skiena S. (2007) Large-scale sentiment analysis for news and blogs. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*.

[GZ09a] Guo Q.-l. y Zhang M. (August 2009) Semantic information integration and question answering based on pervasive agent ontology. *Expert Syst. Appl.* 36(6): 10068–10077.

[GZ09b] Guo Q. y Zhang M. (August 2009) Question answering based on pervasive agent ontology and semantic web. *Know.-Based Syst.* 22(6): 443–448.

[Hal06] Hallett C. (2006) Generic querying of relational databases using natural language generation techniques. In *Proceedings of the Fourth International Natural Language Generation Conference*, INLG '06, pp. 95–102. Association for Computational Linguistics, Stroudsburg, PA, USA.

[HANS90] Hayes P., Andersen P., Nirenburg I., y Schmandt L. (1990) Tcs: a shell for content-based text categorization. In *Artificial Intelligence Applications, 1990., Sixth Conference on*, pp. 320–326 vol.1.

[Har51] Harris Z. S. (1951) Structural linguistics. *University of Chicago Press, Chicago:IL, USA and London, UK, 7th (1966) edition.* .

[Har77] Harris L. (1977) Robot : A high performance natural language processor for data base query. *ACM SIGART Newsletter* 61: 39–40.

[HBLH94] Hersh W., Buckley C., Leone T. J., y Hickam D. (1994) Ohsumed: an interactive retrieval evaluation and new large test collection for research. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '94, pp. 192–201. Springer-Verlag New York, Inc., New York, NY, USA.

[HBML95] Hammond K., Burke R., Martin C., y Lytinen S. (feb 1995) Faq finder: a case-based approach to knowledge navigation. In *Artificial Intelligence for Applications, 1995. Proceedings., 11th Conference on*, pp. 80–86.

[Her01] Hermjakob U. (2001) Parsing and question classification for question answering. In *Proceedings of the workshop on Open-domain question answering - Volume 12*, ODQA '01, pp. 1–6. Association for Computational Linguistics, Stroudsburg, PA, USA.

[HL04a] Hu M. y Liu B. (August 2004) Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. USA, Seattle, WA.

[HL04b] Hu M. y Liu B. (2004) Mining opinion features in customer reviews. In *Proceedings of Nineteenth National Conference on Artificial Intellgience (AAAI'04)*.

[HLP97] Helander M. G., Landauer T. K., y Prabhu P. V. (Eds.) (1997) *Handbook of Human-Computer Interaction*, chapter Using Natural Language Interfaces. Elsevier Science Inc., New York, NY, USA, 2nd edition.

[HM97] Hatzivassiloglou V. y McKeown K. R. (1997) Predicting the semantic orientation of adjectives. In *Proceedings of 35th Meeting of the Association for Computational Linguistics*, pp. 174–181. Madrid, Spain.

[HMU00] Hopcroft J. E., Motwani R., y Ullman J. D. (November 2000) *Introduction to Automata Theory, Languages, and Computation*. Addison Wesley, 2nd edition.

[Hof08] Hoffman T. (2008) Online reputation management is hot - but is it ethical? *Computerworld* .

[HPS07] Hallett C., Power R., y Scott D. (2007) Composing questions through conceptual authoring. *Computational Linguistics* 33: 105–133.

[HSCD09] Hsu C.-H., Song G., Chen R., y Dai S. K. (2009) Using domain ontology to implement a frequently asked questions system. In *2009 WRI World Congress on Computer Science and Information Engineering, CSIE 2009*, volumen 4, pp. 714–718.

[HSL07] Hu M., Sun A., y Lim E. (November 2007) Comments-oriented blog summarization by sentence extraction. In *Proceedings of the sixteenth ACM Conference on information and knowledge management*. Portugal, Lisbon.

[HSSS78] Hendrix G., Sacerdoti E., Sagalowicz D., y Slocum J. (1978) Developing a natural language interface to complex data. *ACM Transactions on Database Systems* 3(2): 105–147.

[HW03] Hacioglu K. y Ward W. (2003) Question classification with support vector machines and error correcting codes. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: companion volume of the Proceedings of HLT-NAACL 2003–short papers - Volume 2*, NAACL-Short '03, pp. 28–30. Association for Computational Linguistics, Stroudsburg, PA, USA.

[HYJ10] Hu W. C., Yu D. F., y Jiau H. C. (2010) A faq finding process in open source project forums. In *Proceedings - 5th International Conference on Software Engineering Advances, ICSEA 2010*, pp. 259–264.

[ID10] Indurkhya N. y Damerau F. J. (2010) *Handbook of natural language processing*, volumen 2. Chapman and Hall/CRC.

[IV98] Ide N. y Véronis J. (1998) Introduction to the special issue on word sense disambiguation: the state of the art. *Computational linguistics* 24(1): 2–40.

[JCL05] Jeon J., Croft W. B., y Lee J. H. (2005) Finding similar questions in large question and answer archives. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, CIKM '05, pp. 84–90. ACM, New York, NY, USA.

[JL06] Jindal N. y Liu B. (2006) Mining comparative sentences and relations. In *AAAI'06.*

[JL08] Jindal N. y Liu B. (2008) Opinion spam and analysis. In *Proceedings of the Conference on Web Search and Web Data Mining (WSDM)*, pp. 219–230. Stanford, CA.

[Joa98] Joachims T. (1998) Text categorization with suport vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning*, ECML '98, pp. 137–142. Springer-Verlag, London, UK, UK.

[Joa02] Joachims T. (2002) *Learning to classify text using support vector machines: Methods, theory and algorithms.* Kluwer Academic Publishers.

[Joa05] Joachims T. (2005) A support vector method for multivariate performance measures. In *Proceedings of the 22nd international conference on Machine learning*, ICML '05, pp. 377–384. ACM, New York, NY, USA.

[Joa06] Joachims T. (2006) Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 217–226. ACM.

[JR05] Jijkoun V. y Rijke M. D. (2005) Retrieving answers from frequently asked questions pages on the web. In *International Conference on Information and Knowledge Management, Proceedings*, pp. 76–83.

[Jua10] Juan Z. M. (2010) An effective similarity measurement for faq question answering system. In *Proceedings of the 2010 International Conference on Electrical and Control Engineering*, ICECE '10, pp. 4638–4641. IEEE Computer Society, Washington, DC, USA.

[KAE11] K. Agbele B. Adetunmbi S. O. y Ekong D. (2011) Applying a novel query reformulation keywords algorithm in a mobile healthcare retrieval context. *Research Journal of Applied Sciences* 6(3): 184–193.

[Kap84] Kaplan S. (1984) Designing a portable natural language database query system. *ACM Transactions on Database Systems* 9: 1–19.

[Kat97] Katz B. (1997) Annotating the World Wide Web using natural language. In *Proceedings of the 5th Conference on computer assisted information searching on the internet (RIAO '97).*

[KB07] Kaufmann E. y Bernstein A. (2007) How useful are natural language interfaces to the semantic web for casual end-users? In *Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference*, ISWC'07/ASWC'07, pp. 281–294. Springer-Verlag, Berlin, Heidelberg.

[KBF07] Kaufmann E., Bernstein A., y Fischer L. (2007) NLP-Reduce: A "naive" but Domain-independent Natural Language Interface for Querying Ontologies.

[KBZ06] Kaufmann E., Bernstein A., y Zumstein R. (2006) Querix: A natural language interface to query ontologies based on clarification dialogs. In *In: 5th ISWC*, pp. 980–981. Springer.

[KCGS96] Karttunen L., Chanod J.-P., Grefenstette G., y Schille A. (December 1996) Regular expressions for language engineering. *Nat. Lang. Eng.* 2(4): 305–328.

[KEW01] Kwok C., Etzioni O., y Weld D. S. (July 2001) Scaling question answering to the web. *ACM Trans. Inf. Syst.* 19(3): 242–262.

[KFY$^+$02] Katz B., Felshin S., Yuret D., Ibrahim A., Lin J. J., Marton G., McFarland A. J., y Temelkuran B. (2002) Omnibase: Uniform access to heterogeneous data for question answering. In *Proceedings of the 6th International Conference on Applications of Natural Language to Information Systems-Revised Papers*, NLDB '02, pp. 230–234. Springer-Verlag, London, UK, UK.

[KGV83] Kirkpatrick S., Gelatt C. D., y Vecchi M. P. (1983) Optimization by simulated annealing. *Science* 220(4598): 671–680.

[KH04] Kim S.-M. y Hovy E. (2004) Determining the sentiment of opinions. In *Proceedings of the International Conference on Computational Linguistic (COLING)*. Geneva, Switzerland.

[KH06] Kim S.-M. y Hovy E. (2006) Automatic identification of pro and con reasons in online reviews. In *Proceedings of the COLING/ACL on Main conference poster sessions*, COLING-ACL '06, pp. 483–490. Association for Computational Linguistics, Stroudsburg, PA, USA.

[KIM07] Kobayashi N., Inui K., y Matsumoto Y. (2007) Extracting aspect-evaluation of aspect-of relations in opinion mining. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 1065–1074. Prague, Czech Republic.

[Kin08] Kinber E. (2008) On learning regular expressions and patterns via membership and correction queries. In *Proceedings of the 9th international colloquium on Grammatical Inference: Algorithms and Applications*, ICGI '08, pp. 125–138. Springer-Verlag, Berlin, Heidelberg.

[Kit82a] Kittredge R. (1982) Sublanguages. *American Journal of Computational Linguistics* 8(2): 79–84.

[Kit82b] Kittredge R. (1982) Variation and homogeneity of sublanguages. *R. Kittredge and J. Lehrberger, editors, Sublanguage: Studies of Language in Restricted Semantic Domains* pp. 107–137.

[KJ97] Kohavi R. y John G. H. (1997) Wrappers for feature subset selection. *Artificial intelligence* 97(1): 273–324.

[Kle56] Kleene S. (1956) *Representation of Events in Nerve Nets and Finite Automata*, pp. 3–42. Princeton University Press, Princeton, N.J., Editors Shannon, C. and Mccarthy, J.

[KM06] Kate R. J. y Mooney R. J. (2006) Using string-kernels for learning semantic parsers. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics*.

[KN06] Kanayama H. y Nasukawa T. (2006) Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pp. 355–363. Association for Computational Linguistics, Stroudsburg, PA, USA.

[Kor97] Korfhage R. R. (1997) *Information Storage and Retrieval*. Wiley, 1st edition.

[KPK01] Kolcz A., Prabakarmurthi V., y Kalita J. (2001) Summarization as feature selection for text categorization. In *Proceedings of the tenth international conference on Information and knowledge management*, CIKM '01, pp. 365–370. ACM, New York, NY, USA.

[KS06] Kim H. y Seo J. (2006) High-performance faq retrieval using an automatic clustering method of query logs. *Information Processing and Management* 42(3): 650–661.

[KS08a] Kim H. y Seo J. (2008) Cluster-based faq retrieval using latent term weights. *IEEE Intelligent Systems* 23(2): 58–65.

[KS08b] Kim H. y Seo J. (March 2008) Cluster-based faq retrieval using latent term weights. *IEEE Intelligent Systems* 23(2): 58–65.

[KWM05] Kate R. J., Wong Y. W., y Mooney R. J. (2005) Learning to transform natural to formal languages. In *Proceedings of the National Conference on Artificial Intelligence*.

[KZGS10] Kwiatkowski T., Zettlemoyer L., Goldwater S., y Steedman M. (2010) Inducing probabilistic ccg grammars from logical form with higher-order unification. In *In Empirical Methods in Natural Language Processing (EMNLP)*.

[LCDZ11] Lue Y., Castellanos M., Dayal U., y Zhai C. (2011) Automatic construction of a context-aware sentiment lexicon: An optimization approach. In *Proceedings of the 20th international conference on World Wide Web*, pp. 347–356. Hyderabad, India.

[LEC08] López V., Eisman E. M., y Castro J. L. (2008) A tool for training primary health care medical students: The virtual simulated patient. In *Tools with Artificial Intelligence, 2008. ICTAI '08. 20th IEEE International Conference on*, volumen 2, pp. 194–201.

[Leh12] Lehmann E. (2012) "student" and small-sample theory. In Rojo J. (Ed.) *Selected Works of E. L. Lehmann*, Selected Works in Probability and Statistics, pp. 1001–1008. Springer US.

[LFQL11] Liu L., Fan X. Z., Qi Q., y Liu X. M. (2011) Ontology-based question expansion for question similarity calculation. *Journal of Beijing Institute of Technology (English Edition)* 20(2): 244–248.

[LFW$^+$08] Li G., Feng J., Wang J., Yu B., y He Y. (2008) Race: Finding and ranking compact connected trees for keyword proximity search over xml documents. In *Proceedings of the 17th International World Wide Web Conference (WWW 2008)*, pp. 1045–1046. Beijing, China.

[LH08] Langdon W. B. y Harrison A. P. (2008) Evolving regular expressions for genechip probe performance prediction. In *Proceedings of the 10th international conference on Parallel Problem Solving from Nature: PPSN X*, pp. 1061–1070. Springer-Verlag, Berlin, Heidelberg.

[LHK98] Lenz M., Hübner A., y Kunze M. (1998) Question answering with textual cbr. In Andreasen T., Christiansen H., y Larsen H. (Eds.) *Flexible Query Answering Systems*, volumen 1495 of *Lecture Notes in Computer Science*, pp. 236–247. Springer Berlin / Heidelberg.

[LHM98] Liu B., Hsu W., y Ma Y. (1998) Integrating classification and association rule mining. In *KDD'98*.

[Li02] Li W. (2002) Question classification using language modeling. In *CIIR Technical Report*. University of Massachusetts, Amherst.

[Lie80] Lieber R. (1980) On the organization of the lexicon. Thesis, Massachusetts Institute of Technology. Dept. of Linguistics and Philosophy. Advisor: Morris Halle.

[Liu12] Liu B. (2012) Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies* 5(1): 1–167.

[LK03] Lapalme G. y Kosseim L. (July 2003) Mercure: Towards an Automatic E-Mail Follow-up System. *IEEE Computational Intelligence Bulletin* 2(1): 14–18.

[LKHC04] Lim S., Kim K.-M., Hong J.-H., y Cho S.-B. (6-7 December 2004) Interactive genetic programming for the sentence generation of dialog-based travel planning system. In Mckay R. I. y Cho S.-B. (Eds.) *Proceedings of The Second Asian-Pacific Workshop on Genetic Programming*. Cairns, Australia.

[LKR$^+$08] Li Y., Krishnamurthy R., Raghavan S., Vaithyanathan S., y Jagadish H. V. (2008) Regular expression learning for information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pp. 21–30. Association for Computational Linguistics, Stroudsburg, PA, USA.

[LLCM03] Liu T., Liu S., Chen Z., y Ma W. (2003) An evaluation on feature selection for text clustering. In *Machine Learning, International Workshop then Conference*, volumen 20 (2), page 488.

[LLL10] Liu H., Lin X., y Liu C. (2010) Research and implementation of ontological qa system based on faq. *Journal of Convergence Information Technology* 5(3): 79–85.

[LMS02] Lavelli A., Magnini B., y Sebastiani F. (2002) Building thematic lexical resources by bootstrapping and machine learning. In *Proc. of the Workshop "Linguistic Knowledge Acquisition and Rrepresentation: Bootstrapping Annotated Language Data", Workshop at LREC-2002.*

[LMS06] Lloyd L., Mehler A., y Skiena S. (2006) Identifying co-referential names across large corpora. *Lecture Notes on Combinatorial Pattern Matching* 4009: 12–23.

[LNLZ08] Lu W., Ng H. T., Lee W. S., y Zettlemoyer L. S. (2008) A generative model for parsing natural language to meaning representations. In *The Conference on Empirical Methods in Natural Language Processing.*

[LNS$^+$10] López V., Nikolov A., Sabou M., Uren V., Motta E., y D'Aquin M. (2010) Scaling up question-answering to linked data. In *Proceedings of the 17th international conference on Knowledge engineering and management by the masses*, EKAW'10, pp. 193–210. Springer-Verlag, Berlin, Heidelberg.

[LPM05] López V., Pasin M., y Motta E. (2005) Aqualog: An ontology-portable question answering system for the semantic web. In *In Proceedings of ESWC*, pp. 546–562.

[LR06] Li X. y Roth D. (September 2006) Learning question classifiers: the role of semantic information. *Nat. Lang. Eng.* 12(3): 229–249.

[LRM03] L. Razmerita A. A. y Maedche A. (2003) Ontology-based user modeling for knowledge management systems. In *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, volumen 2702, pp. 213–217.

[LSCP96] Lewis D. D., Schapire R. E., Callan J. P., y Papka R. (1996) Training algorithms for linear text classifiers. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '96, pp. 298–306. ACM, New York, NY, USA.

[LUSM11] López V., Uren V., Sabou M., y Motta E. (April 2011) Is question answering fit for the semantic web?: a survey. *Semant. web* 2(2): 125–155.

[MBF$^+$90] Miller G., Beckwith R., Fellbaum C., Gross D., y Miller K. (1990) Introduction to wordnet: An on-line lexical database. *International Journal of Lexicography (special issue)* 3(4): 235–312.

[MC04] Mullen T. y Collier N. (2004) Sentiment analysis using support vector machines with diverse information sources. In *Proceedings of EMNLP*, pp. 412–418.

[McC90] McCord M. C. (May 1990) Slot grammar: a system for simpler construction of practical natural language grammars. Technical report rc15582(d69261), IBM.

[MCLZ10] Moreo A., Castro J. L., López V., y Zurita J. M. (February 2010) Emd: una metodología de recuperación para sistemas de lenguaje natural basados en casos. In *XV Congreso Español sobre Tecnologías y Lógica Fuzzy*, pp. 217–222. Puntaumbría, Huelva.

[MCZ12] Moreo A., Castro J. L., y Zurita J. M. (2012) Handling context in lexicon-based sentiment analysis. In Greco S., Bouchon-Meunier B., Coletti G., Fedrizzi M., Matarazzo B., y Yager R. R. (Eds.) *Advances in Computational Intelligence*, volumen 298 of *Communications in Computer and Information Science*, pp. 245–254. Springer Berlin Heidelberg.

[MHG$^+$02] Moldovan D., Harabagiu S., Girju R., Morarescu P., Lacatusu F., Novischi A., Badulescu A., y Bolohan O. (2002) LCC Tools for Question Answering. *notebook of the Eleventh Text REtrieval Conference (TREC2002)* pp. 144–154.

[MHN$^+$07] McDonald R., Hannan K., Neylon T., Wells M., y Reynar J. (2007) Structured models for fine-to-coarse sentiment analysis. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 432–439. Prague, Czech Republic.

[Min04] Minock M. (2004) Natural language access to relational databases through step. Technical report, Dept. Computer Science, University of Umea, Umea, Sweden.

[Min07] Minock M. (2007) A step towards realizing codd's vision of rendezvous with the casual user. In *Proc. 33rd International Conference on Very Large Databases*, pp. 1358–1361. Vienna, Austria.

[Mit97] Mitchell T. (1997) *Machine Learning*. McGraw-Hill, New York.

[MLD08] Miao Q., Li Q., y Dai R. (October 2008) An integration strategy for mining product features and opinions. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pp. 1369–1370. Napa Valley, California.

[MM08] Malouf R. y Mullen T. (2008) Taking sides: User classification for informal online political discourse. *Internet Research* 18: 177–190.

[MN98] McCallum A. y Nigam K. (1998) A comparison of event models for naive bayes text classification. In *AAAI-98 Workshop on Learning For Text Categorization*, pp. 41–48. AAAI Press.

[MNCZ12] Moreo A., Navarro M., Castro J. L., y Zurita J. M. (2012) A high-performance faq retrieval method using minimal differentiator expressions. *Knowledge-Based Systems* 36(0): 9 – 20.

[Mø10] Møller A. (2010) dk.brics.automaton – finite-state automata and regular expressions for Java. `http://www.brics.dk/automaton/`.

[MOC$^+$08] Morrey I., Oram A., Cooper D., Rogers D., y Stephenson P. (2008) Grammatical inference techniques and their application in ground investigation. *Computer-Aided Civil and Infrastructure Engineering* 23(1): 17–30.

[MR01] Marín Ruiz N. (2001) *Estudio de la vaguedad en los sistemas de bases de datos orientados a objetos: tipos difusos y sus aplicaciones.* PhD thesis, Universidad de Granada.

[MRCZ12a] Moreo A., Romero M., Castro J. L., y Zurita J. M. (2012) Faqtory: A framework to provide high-quality faq retrieval systems. *Expert Systems with Applications* 39(14): 11525 – 11534.

[MRCZ12b] Moreo A., Romero M., Castro J. L., y Zurita J. M. (2012) Lexicon-based comments-oriented news sentiment analyzer system. *Expert Systems with Applications* 39(10): 9166 – 9180.

[MRPCVMCT96] Medina Rodríguez J. M., Pons Capote O., Vila Miranda M. A., y Cubero Talavera J. C. (1996) Client/server architecture for fuzzy relational databases. *Mathware & soft computing. 1996 Vol. 3 Núm. 3* .

[MRPVR07] Manjón B. F., Rodríguez J. B., Pulido J. A. G., y Vega-Rodríguez J. M. S. P. M. A. (2007) Computers and education: E-learning, from theory to practice.

[MS99] Manning C. y Schutze H. (1999) Foundations of statistical natural language processing. *In MIT Press, Cambridge, Massachusetts* .

[MS01] Maedche A. y Staab S. (2001) Ontology learning for the semantic web. *Intelligent Systems, IEEE* 16(2): 72–79.

[MSC11] McCrae J., Spohr D., y Cimiano P. (2011) Linking lexical resources and ontologies on the semantic web with lemon. In *Proceedings of the 8th extended semantic web conference on The semantic web: research and applications - Volume Part I*, ESWC'11, pp. 245–259. Springer-Verlag, Berlin, Heidelberg.

[MV07] Mollá D. y Vicedo J. L. (March 2007) Question answering in restricted domains: An overview. *Computational Linguistics* 33(1): 41–61.

[MWL$^+$07] Mukras R., Wiratunga N., Lothian R., Chakraborti S., y Harper D. (2007) Information gain feature selection for ordinal text classification using probability re-distribution. In *Proceedings of the Textlink workshop at IJCAI*, volumen 7.

[MZ09] Marom Y. y Zukerman I. (December 2009) An empirical study of corpus-based response automation methods for an e-mail-based help-desk domain. *Computational Linguistics.* 35(4): 597–635.

[MZT$^+$10] Mendel J., Zadeh L., Trillas E., Yager R., Lawry J., Hagras H., y Guadarrama S. (2010) What computing with words means to me [discussion forum]. *Computational Intelligence Magazine, IEEE* 5(1): 20–26.

[NG01] Noord G. v. y Gerdemann D. (2001) An extendible regular expression compiler for finite-state approaches in natural language processing. In *Revised Papers from the 4th International Workshop on Automata Implementation*, WIA '99, pp. 122–139. Springer-Verlag, London, UK, UK.

[NGL97] Ng H. T., Goh W. B., y Low K. L. (1997) Feature selection, perceptron learning, and a usability case study for text categorization. In *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '97, pp. 67–73. ACM, New York, NY, USA.

[NLWC98] Ngan P. S., Leung K. S., Wong M. L., y Cheng J. C. Y. (1998) Using grammar based genetic programming for data mining of medical knowledge. *Genetic Programming 1998: Proceedings of the Third Annual Conference* pp. 254–259.

[NM01] Nomoto T. y Matsumoto Y. (2001) A new approach to unsupervised text summarization. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 26–34. ACM.

[NPI11] Neviarouskaya A., Prendinger H., y Ishizuka M. (2011) Sentiful: A lexicon for sentiment analysis. *IEEE Transactions on Affective Computing* 2(1): 22–36.

[NY03] Nasukawa T. y Yi J. (October 2003) Sentiment analysis: capturing favorability using natural language processing. In *Proceedings of the 2nd international conference on Knowledge capture*. USA, Sanibel Island, FL.

[OB96] Ogden W. C. y Bernick P. (1996) Using natural language interfaces. In *Handbook Of Human-Ccomputer Interaction*. Elsevier Science Publishers B.V. (North-Holland).

[ÖG10] Özgür L. y Güngör T. (2010) Text classification with the support of pruned dependency patterns. *Pattern Recognition Letters* 31(12): 1598–1607.

[OMBC06] Ogden W., Mcdonald J., Bernick P., y Chadwick R. (2006) Habitability in question-answering systems. In Strzalkowski T. y Harabagiu S. (Eds.) *Advances in Open Domain Question Answering*, volumen 32 of *Text, Speech and Language Technology*, pp. 457–473. Springer Netherlands.

[OOMH08] Ou S., Orasan C., Mekhaldi D., y Hasler L. (2008) Automatic question pattern generation for ontology-based question answering. In *Proceedings of the 21th International Florida Artificial Intelligence Research Society Conference, FLAIRS-21*, pp. 183–188.

[OT09] Ohana B. y Tierney B. (2009) Sentiment classification of reviews using sentiwordnet. In *9th. IT & T Conference*.

[Ott92] Ott N. (1992) Aspects of the automatic generation of sql statements in a natural language query interface. *Information Systems* 17(2): 147–159.

[Owe00] Owei V. (2000) Natural language querying of databases: an information extraction approach in the conceptual query language. *International Journal of Human - Computer Studies* 53: 439–492.

[PC08] Pons Capote O. (2008) *Introducción a los sistemas de bases de datos*. Editorial Paraninfo.

[PE05] Popescu A. y Etzioni O. (2005) Extracting product features and opinions from reviews. In *Proceedings of EMNLP*, pp. 339–346.

[PEK03] Popescu A. M., Etzioni O., y Kautz H. (2003) Towards a theory of natural language interfaces to databases. In *8th Intl. Conf. on Intelligent User Interfaces*, pp. 149–157. Miami, FL.

[PL08] Pang B. y Lee L. (2008) Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval* 2(1): 1–135.

[PLV02] Pang B., Lee L., y Vaithyanathan S. (July 2002) Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 79–86. Association for Computational Linguistics, Philadelphia.

[Pow07] Powers D. M. W. (2007) Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation. Technical Report SIE-07-001, School of Informatics and Engineering, Flinders University, Adelaide, Australia.

[PPG+05] Pazos R. A., Pérez J., González J. J., Gelbukh A., Sidorov G., y Rodríguez M. J. (2005) A domain independent natural language interface to databases capable of processing complex queries. In *MICAI 2005*, pp. 833–842.

[PRGBAL+13] Pazos R. R. A., González B. J. J., Aguirre L. M., Martínez F. J. A., y Fraire H. H. J. (2013) Natural language interfaces to databases: An analysis of the state of the art. In Castillo O., Melin P., y Kacprzyk J. (Eds.) *Recent Advances on Hybrid Intelligent Systems*, volumen 451 of *Studies in Computational Intelligence*, pp. 463–480. Springer Berlin Heidelberg.

[Pus91] Pustejovsky J. (December 1991) The generative lexicon. *Comput. Linguist.* 17(4): 409–441.

[QLBC09] Qiu G., Liu B., Bu J., y Chen C. (2009) Expanding domain sentiment lexicon through double propagation. In *Proceedings of the 21st International Joint Conferences on Artificial Intelligence*, pp. 1199–1204. Pasadena, California.

[RF08] Rozenfeld B. y Feldman R. (October 2008) Self-supervised relation extraction from the web. *Knowl. Inf. Syst.* 17(1): 17–33.

[RGG11] Rafrafi A., Gigue V., y Gallinari P. (2011) Pénalisation des mots fréquents pour la classification de sentiments. *Les Cahiers du numérique* 7(2): 1622–1494.

[RH02] Ravichandran D. y Hovy E. (2002) Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pp. 41–47. Association for Computational Linguistics, Stroudsburg, PA, USA.

[RHM+02] Rinaldi F., Hess M., Mollá D., Schwitter R., Dowdall J., Schneider G., y Fournier R. (2002) Answer extraction in technical domains. In *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing '02, pp. 360–369. Springer-Verlag, London, UK, UK.

[RMC13a] Romero M., Moreo A., y Castro J. L. (2013) A cloud of faq: A highly-precise faq retrieval system for the web 2.0.

[RMC13b] Romero M., Moreo A., y Castro J. L. (September 2013) Collaborative system for learning based on questionnaires and tasks. In *(to appear) 4th International Conference on European Transnational Education, (ICEUTE'13)*, page xxx. Salamanca, Spain.

[RMCZ12] Romero M., Moreo A., Castro J. L., y Zurita J. M. (2012) Using wikipedia concepts and frequency in language to extract key terms from support documents. *Expert Systems with Applications* 39(18): 13480 – 13491.

[RMM97] Reis P., Matias J., y Mamede N. (1997) Edite - a natural language interface to databases: a new dimension for an approach. In *Proc. 4th International Conference on Information and Communication Technology in Tourism.* Edinburgh, Scottland.

[Ros00] Ross B. J. (November 2000) Probabilistic pattern matching and the evolution of stochastic regular expressions. *Applied Intelligence* 13(3): 285–300.

[RS99] Ruiz M. E. y Srinivasan P. (1999) Hierarchical neural networks for text categorization (poster abstract). In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pp. 281–282. ACM, New York, NY, USA.

[SA95] Shinohara T. y Arikawa S. (1995) Pattern inference. In *Algorithmic Learning for Knowledge-Based Systems, GOSLER Final Report*, pp. 259–291. Springer-Verlag, London, UK, UK.

[Sak88] Sakakibara Y. (1988) Learning context-free grammars from structural data in polynomial time. In *Proceedings of the first annual workshop on Computational learning theory*, COLT '88, pp. 330–344. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[Sak90] Sakakibara Y. (1990) Learning context-free grammars from structural data in polynomial-time. *Theoretical Computer Science* 76(2-3): 223–242.

[Sak05] Sakakibara Y. (2005) Grammatical inference in bioinformatics. *Ieee Transactions on Pattern Analysis and Machine Intelligence* 27(7): 1051–1062.

[Sat97] Satoshi S. (1997) A new direction for sublanguage nlp. In *New Methods in Language Processing*, pp. 165–177.

[SB88] Salton G. y Buckley C. (1988) Term-weighting approaches in automatic text retrieval. *Information processing & management* 24(5): 513–523.

[SB04] Shamsfard M. y Barforoush A. A. (2004) Learning ontologies from natural language texts. *International Journal of Human-Computer Studies* 60(1): 17–63.

[SB06] Soricut R. y Brill E. (March 2006) Automatic question answering using the web: Beyond the factoid. *Inf. Retr.* 9(2): 191–206.

[SC07] Sun M. y Chai J. Y. (August 2007) Discourse processing for context question answering based on linguistic knowledge. *Know.-Based Syst.* 20(6): 511–526.

[SDYH08] Sung C.-L., Day M.-Y., Yen H.-C., y Hsu W.-L. (2008) A template alignment algorithm for question classification. In *IEEE International Conference on Intelligence and Security Informatics, 2008, IEEE ISI 2008*, pp. 197–199.

[Seb02] Sebastiani F. (March 2002) Machine learning in automated text categorization. *ACM Comput. Surv.* 34(1): 1–47.

[Seb05] Sebastiani F. (2005) Text categorization. In *Text Mining and its Applications to Intelligence, CRM and Knowledge Management*, pp. 109–129. WIT Press.

[SFK00] Stamatatos E., Fakotakis N., y Kokkinakis G. (2000) Automatic text categorization in terms of genre and author. *Computational linguistics* 26(4): 471–495.

[SGS06] Schlaefer N., Gieselmann P., y Schaaf T. (2006) A pattern learning approach to question answering within the ephyra framework. In *Proceedings of the Ninth International Conference on TEXT, SPEECH and DIALOGUE.*

[SJ05] Spärck Jones K. (March 2005) Some points in a time. *Comput. Linguist.* 31(1): 1–14.

[SKA68] Stone P., Kirsch J., y Associates C. C. (1968) *The General Inquirer: A Computer Approach to Content Analysis.* Massachusetts Institute of technology.

[SM86] Salton G. y McGill M. J. (1986) *Introduction to Modern Information Retrieval.* McGraw-Hill, Inc., New York, NY, USA.

[Sne99] Sneiders E. (1999) Automated faq answering: Continued experience with shallow language understanding. In *Proceedings for the 1999 AAAI Fall Symposium on Question Answering Systems.*

[Sne02] Sneiders E. (2002) Automated question answering using question templates that cover the conceptual model of the database. In *Proceedings of the 6th International Conference on Applications of Natural Language to Information Systems-Revised Papers*, NLDB '02, pp. 235–239. Springer-Verlag, London, UK, UK.

[Sne09] Sneiders E. (2009) Automated faq answering with question-specific knowledge representation for web self-service. In *Proceedings of the 2nd conference on Human System Interactions*, HSI'09, pp. 295–302. IEEE Press, Piscataway, NJ, USA.

[Sne10] Sneiders E. (2010) Automated email answering by text pattern matching. In *Proceedings of the 7th international conference on Advances in natural language processing*, IceTAL'10, pp. 381–392. Springer-Verlag, Berlin, Heidelberg.

[SNSP01] Saiz-Noeda M., Suárez A., y Palomar M. (2001) Semantic pattern learning through maximum entropy-based wsd technique. In *Proceedings of the 2001 workshop on Computational Natural Language Learning - Volume 7*, ConLL '01. Association for Computational Linguistics, Stroudsburg, PA, USA.

[Sod99] Soderland S. (February 1999) Learning information extraction rules for semi-structured and free text. *Mach. Learn.* 34(1-3): 233–272.

[Sou01] Soubbotin M. M. (2001) Patterns of potential answer expressions as clues to the right answers. In *In Proceedings of the Tenth Text REtrieval Conference (TREC*, pp. 293–302.

[SPG09] Salazar V., Pena J., y Granado J. (2009) Reducing the effort in the creation of new patients using the virtual simulated patient framework. In *Fuzzy Systems, 2009. FUZZ-IEEE 2009. IEEE International Conference on*, pp. 764–769.

[Spo93] Spoerri A. (1993) Infocrystal: a visual tool for information retrieval & management. In *Proceedings of the second international conference on Information and knowledge management*, CIKM '93, pp. 11–20. ACM, New York, NY, USA.

[SS00] Schapire R. E. y Singer Y. (2000) Boostexter: A boosting-based system for text categorization. *Machine learning* 39(2-3): 135–168.

[SSS98] Schapire R. E., Singer Y., y Singhal A. (1998) Boosting and rocchio applied to text filtering. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 215–223. ACM.

[SVFBCN$^+$12] Sánchez-Vera M. D. M., Fernández-Breis J. T., Castellanos-Nieves D., Frutos-Morales F., y Prendes-Espinosa M. P. (September 2012) Semantic web technologies for generating feedback in online assessment environments. *Know.-Based Syst.* 33: 152–165.

[TA03] Tsamardinos I. y Aliferis C. F. (2003) Towards principled feature selection: Relevancy, filters and wrappers. In *Proceedings of the ninth international workshop on Artificial Intelligence and Statistics*. Morgan Kaufmann Publishers: Key West, FL, USA.

[TBT$^+$11] Taboada M., Brooke J., Tofiloski M., Voll K., y Stede M. (2011) Lexicon-based methods for sentiment analysis. *Computational Linguistics* 37(2): 267–307.

[TG08] Tasci S. y Güngör T. (2008) An evaluation of existing and new feature selection metrics in text categorization. In *Computer and Information Sciences, 2008. ISCIS'08. 23rd International Symposium on*, pp. 1–6. IEEE.

[TG12] Takçı H. y Güngör T. (2012) A high performance centroid-based classification approach for language identification. *Pattern Recognition Letters* .

[TLL98] T.K. Landauer P. F. y Laham D. (1998) An introduction to latent semantic analysis. *Discourse Processes* 25(2-3): 259–284.

[TM01] Tang L. y Mooney R. J. (2001) Using multiple clause constructors in inductive logic programming for semantic parsing. In *Proceedings of the 12yh European Conference on Machine Learning (ECML-2001)*, pp. 466–477. Freiburg, Germany.

[TM02] Thompson C. A. y Mooney R. J. (2002) Acquiring word-meaning mappings for natural language interfaces. *Journal of Artificial Intelligence Research* .

[TM08] Titov I. y McDonald R. (2008) A joint model of text and aspect ratings for sentiment summarization. In *The 46th Annual Meeting of the Association for Computational Linguistic*, pp. 308–316. Columbus, Ohio.

[TM11] Tckstrm O. y McDonald R. (2011) Semi-supervised latent variable models for sentence-level sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pp. 569–574. Portland, Oregon.

[TMG$^+$07] Trillas E., Moraga C., Guadarrama S., Cubillo S., y Castiñeira E. (2007) Computing with antonyms. *Forging New Frontiers: Fuzzy Pioneers I* pp. 133–153.

[TPT05] Thompson C., Pazandak P., y Tennant H. (2005) Talk to your semantic web. *IEEE Internet Computing* 9(6): 75–78. cited By (since 1996) 22.

[TR08] Tapeh A. G. y Rahgozar M. (December 2008) A knowledge-based question answering system for b2c ecommerce. *Know.-Based Syst.* 21(8): 946–950.

[TRG06] Trillas E., Renedo E., y Guadarrama S. (2006) Fuzzy sets vs language. *Computational Intelligence: Theory and Practice, ser. Studies in fuzziness and soft computing, B. Reusch, Ed. Springer* 164: 353–366.

[TT75] Thompson F. y Thompson B. (1975) Practical natural language processing: The rel system as prototype. *Advances in Computers* 13: 39–40.

[TT85] Thompson B. H. y Thompson F. B. (April 1985) Ask is transportable in half a dozen ways. *ACM Trans. Inf. Syst.* 3(2): 185–203.

[TT10] Thangamani M. y Thangaraj P. (2010) Integrated clustering and feature selection scheme for text documents. *Journal of Computer Science* 6(5): 536–541.

[Tur02a] Turney P. D. (July 2002) Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Philadelphia, Pennsylvania.

[Tur02b] Turney P. D. (July 2002) Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Philadelphia, Pennsylvania.

[TW11a] Tan S. y Wang Y. (2011) Weighted scl model for adaptation of sentiment classification. *Expert Systems with Applications* 38(8): 10524–10531.

[TW11b] Tan S. y Wu Q. (2011) A random walk algorithm for automatic construction of domain-oriented sentiment lexicon. *Expert Systems with Applications* 38(10): 12094–12100.

[TZ08] Tan S. y Zhang J. (2008) An empirical study of sentiment analysis for chinese documents. *Expert Systems with Applications* 34(4): 2622–2629.

[UGB05] U. Galassi A. Giordana L. S. y Botta M. (2005) Learning profiles based on hierarchical hidden markov model. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 3488 LNAI: 47–55.

[UHC10] Unger C., Hieber F., y Cimiano P. (06/2010 2010) Generating ltag grammars from a lexicon-ontology interface. In Bangalore S., Frank R., y Romero M. (Eds.) *Proceedings of the 10th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+10)*, pp. 61–68. Yale University.

[Val84] Valiant L. G. (November 1984) A theory of the learnable. *Commun. ACM* 27(11): 1134–1142.

[Vap95] Vapnik V. N. (1995) *The nature of statistical learning theory.* Springer-Verlag New York, Inc., New York, NY, USA.

[VBGHM10] Velikovich L., Blair-Goldensohn S., Hannan K., y McDonald R. (2010) The viability of web-derived polarity lexicons. In *The 2010 Annual Conference of the North American Chapter of the ACL*, pp. 777–785. Los Angeles, California.

[Č85] Černý V. (January 1985) Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications* 45(1): 41–51.

[Voo01] Voorhees E. M. (December 2001) The trec question answering track. *Nat. Lang. Eng.* 7(4): 361–378.

[VZPM05] Van Zaanen M., Pizzato L. A., y Mollá D. (2005) Question classification by structure induction. In *Proceedings of the 19th international joint conference on Artificial intelligence*, IJCAI'05, pp. 1638–1639. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[Wal75] Waltz D. (1975) Natural language access to a large data base: An engineering approach. In *4th Int. Joint Conf. on Artificial Intelligence*, pp. 868–872. Tbilisi, U.S.S.R.

[Wal78] Waltz D. (1978) An english language question answering system for a large relational database. *Communications of the ACM* 21(7): 526–539.

[Wat68] Watt W. C. (1968) Habitability. *American Documentation* 19(3): 338–351.

[Whi95] Whitehead S. D. (1995) Auto-faq: an experiment in cyberspace leveraging. *Computer Networks and ISDN Systems* 28(1-2): 137–146.

[Win00] Winiwarter W. (November 2000) Adaptive natural language interfaces to faq knowledge bases. *Data Knowl. Eng.* 35(2): 181–199.

[WKW72] Woods W., Kaplan R., y Webber B. (1972) The lunar sciences natural language information system: Final report. In *BBN Report 2378, Bolt Beranek and Newman Inc.* Cambridge, Massachusetts.

[WM06] Wong Y. W. y Mooney R. J. (2006) Learning for semantic parsing with statistical machine translation. In *Proceedings of Human Language Technology Conference / North American Chapter of the Association for Computational Linguistics Annual Meeting (HLT-NAACL-06)*, pp. 439–446. New York City, NY.

[WM07] Wong Y. W. y Mooney R. J. (2007) Learning synchronous grammars for semantic parsing with lambda calculus. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL 2007)*, pp. 960–967.

[WMC09] Wang K., Ming Z., y Chua T.-S. (2009) A syntactic tree matching approach to finding similar questions in community-based qa services. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, pp. 187–194. ACM, New York, NY, USA.

[WT11] Wu Q. y Tan S. (2011) A two-stage framework for cross-domain sentiment classification. *Expert Systems with Applications* 38(11): 14269–14275.

[WTRM08] Wang F., Teng G., Ren L., y Ma J. (2008) Research on mechanism of agricultural faq retrieval based on ontology. In *Proceedings of the 2008 Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, SNPD '08, pp. 955–958. IEEE Computer Society, Washington, DC, USA.

[WWB$^+$04] Wiebe J., Wilson T., Bruce R., Bell M., y Martin M. (2004) Learning subjective language. *Computational Linguistics* 30: 277–308.

[WWC05] Wiebe J., Wilson T., y Cardie C. (2005) Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation* 2(1): 165–210.

[WWH04] Wilson T., Wiebe J., y Hwa R. (2004) Just how mad are you? finding strong and wake opinion clauses. In *Proceedings of AAAI*, pp. 761–769. San Jose, CA.

[WWH05] Wilson T., Wiebe J., y Hoffmann P. (2005) Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pp. 347–354. Vancouver.

[WXZY07] Wang C., Xiong M., Zhou Q., y Yu Y. (2007) Panto – a portable natural language interface to ontologies. In *4th ESWC, Innsbruck*, pp. 473–487. Springer-Verlag.

[WZT$^+$11] Wu L., Zhou Y., Tan F., Yang F., y Li J. (2011) Generating syntactic tree templates for feature-based opinion mining. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 7121 LNAI(PART 2): 1–12.

[XJC08] Xue X., Jeon J., y Croft W. B. (2008) Retrieval models for question and answer archives. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pp. 475–482. ACM, New York, NY, USA.

[XLL$^+$08] Xu J., Liu T.-Y., Lu M., Li H., y Ma W.-Y. (2008) Directly optimizing evaluation measures in learning to rank. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 107–114. ACM.

[Yan09a] Yang C.-Y. (2009) A semantic faq system for online community learning. *Journal of Software* 4(2): 153–158.

[Yan09b] Yang S.-Y. (March 2009) Developing of an ontological interface agent with template-based linguistic processing technique for faq services. *Expert Syst. Appl.* 36(2): 4049–4060.

[YCH07] Yang S.-Y., Chuang F.-C., y Ho C.-S. (June 2007) Ontology-supported faq processing and ranking techniques. *J. Intell. Inf. Syst.* 28(3): 233–251.

[YL99] Yang Y. y Liu X. (1999) A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pp. 42–49. ACM, New York, NY, USA.

[YNBN03] Yi J., Nasukawa T., Bunescu R., y Niblack W. (November 2003) Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In *Proceedings of the Third IEEE International Conference on Data Mining.*

[Yok95] Yokomori T. (1995) Machine intelligence 13. In Furukawa K., Michie D., y Muggleton S. (Eds.) *Learning non-deterministic finite automata from queries and counterexamples*, pp. 169–189. Oxford University Press, Inc., New York, NY, USA.

[YP97] Yang Y. y Pedersen J. O. (1997) A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference* *on Machine Learning*, ICML '97, pp. 412–420. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[YYC10] Yessenalina A., Yue Y., y Cardie C. (2010) Multi-level structured models for document-level sentiment classification. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 1046–1056. Massachusetts, USA.

[Zaa01] Zaanen M. V. (2001) *Bootstrapping Structure into Language: Alignment-Based Learning*. PhD thesis, School of Computing, University of Leeds, U.K.

[ZC07] Zettlemoyer L. S. y Collins M. (2007) Online learning of relaxed ccg grammars for parsing to logical form. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLPCoNLL 2007)*, pp. 678–687.

[ZL02] Zhang D. y Lee W. S. (2002) Web based pattern mining and matching approach to question answering. In *TREC'02*, pp. 497–504.

[ZL03] Zhang D. y Lee W. S. (2003) Question classification using support vector machines. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, pp. 26–32. ACM, New York, NY, USA.

[ZV06] Zhang Z. y Varadarajan B. (November 2006) Utility scoring of product reviews. In *Proceedings of the 15th ACM international conference on Information and knowledge management*. USA, Arlington, Virginia.

[ZXKJ11] Zhai Z., Xu H., Kang B., y Jia P. (2011) Exploiting effective features for chinese sentiment classification. *Expert Systems with Applications* 38(8): 9139–9146.