MIGUEL LÓPEZ CAMPOS

GENERACIÓN AUTOMÁTICA DE SÍNTESIS DE OPINIONES.

MINERÍA DE DATOS DESCRIPTIVA APLICADA AL PROCESAMIENTO DEL LENGUAJE NATURAL.

TESIS DOCTORAL
PROGRAMA DE DOCTORADO EN TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN

DIRECTORES:
MARÍA VICTORIA LUZÓN GARCÍA
EUGENIO MARTÍNEZ CÁMARA

UNIVERSIDAD DE GRANADA

UNIVERSIDAD DE GRANADA

Escuela Internacional de Posgrado

GENERACIÓN AUTOMÁTICA DE SÍNTESIS DE OPINIONES. MINERÍA DE DATOS DESCRIPTIVA APLICADA AL PROCESAMIENTO DEL LENGUAJE NATURAL

MIGUEL LÓPEZ CAMPOS

Tesis doctoral

Programa de Doctorado en Tecnologías de la Información y Comunicación

Directores
María Victoria Luzón García
Eugenio Martínez Cámara

ESCUELA INTERNACIONAL DE POSGRADO
E.T.S. INGENIERÍAS INFORMÁTICA Y DE TELECOMUNICACIÓN
Universidad de Granada

Editor: Universidad de Granada. Tesis Doctorales
Autor: Miguel López Campos
ISBN: 978-84-1117-428-2
URI: http://hdl.handle.net/10481/75956

ÍNDICE GENERAL

GLOSARIO DE ACRÓNIMOS

- **ABSA.** Análisis de opiniones a nivel de aspecto (*Aspect-Based Sentiment Analysis*)
- **AE.** Algoritmo Evolutivo.
- **AG.** Algoritmo Genético.
- **AM.** Algoritmo Memético
- **AO.** Análisis de Opiniones.
- **DA.** Detección de Aspectos.
- **ED.** Evolución Diferencial.
- **MCO.** Modelo para la Clasificación de la Opinión.
- **MPDS.** Minería de Patrones Descriptiva Supervisada.
- **OS.** Orientación Semántica.
- **PLN.** Procesamiento del Lenguaje Natural.
- **SD.** Descubrimiento de subgrupos (*Subgroup Discovery*).
- **SO.** Síntesis de Opiniones.

1

INTRODUCCIÓN

La Síntesis de Opiniones tiene como objetivo sintetizar la subjetividad expresada por distintos sujetos con respecto a una entidad. Como campo perteneciente a la Inteligencia Artificial, la Síntesis de Opiniones también se abre a los desafíos y necesidades de la Inteligencia Artificial, como son la democratización y transparencia de modelos y recursos. En base a estas mencionadas necesidades, en este capítulo de introducción primero motivaremos esta tesis y las distintas contribuciones realizadas a lo largo de la misma. Finalmente, se explicará cómo se estructura esta tesis.

1.1 MOTIVACIÓN

El desarrollo de la Inteligencia Artificial (IA) y el gran impacto que esta tiene en la sociedad ha derivado en la aparición de nuevos desafíos, campos de estudio y paradigmas que den solución a las necesidades que se presentan. Por un lado, nace la necesidad de la democratización de la IA, es decir, que esta sea más accesible, suponiendo esto el desarrollo de modelos cada vez más generales, abiertos y adaptables a distintos dominios [13]. Por otro lado, la IA es cada vez más influyente a la hora de la toma de decisiones en todos los ámbitos. Este hecho trae en consecuencia que exista una creciente necesidad de desarrollar modelos explicables y fácilmente interpretables por parte del ser humano de manera que este conozca en qué aspectos se basa un determinado modelo a la hora de dar una determinada respuesta [6].

La obtención de conocimiento a partir de opiniones es un claro ejemplo de aplicación de IA que también requiere de las necesidades ya descritas. En este ámbito, tanto fabricante o vendedor como potenciales clientes, requieren de modelos lo suficientemente explicables y accesibles para su toma de decisiones. El Análisis de Opiniones (AO), presentado en [81] como *Sentiment Analysis* en inglés, es un campo de estudio dentro del campo del Procesamiento del Lenguaje Natural (PLN) cuyo fin es el tratamiento y estudio computacional de la subjetividad, el sentimiento y la opinión expresadas en el texto [106]. El AO ha sufrido un incremento del interés debido al desarrollo de nuevos paradigmas en la Web 2.0 como son, entre otros, el comercio electrónico o las redes sociales, dos fuentes de opiniones que pueden aportar información valiosa.

El AO abarca tareas que pueden realizarse a diferentes niveles de granularidad. Por ejemplo, dada una opinión, por medio de técnicas de AO podemos determinar la polaridad del sentimiento global de la opinión (nivel de documento), obtener la polaridad de un segmento de texto u oración (nivel de oración) o determinar la polaridad del sentimiento expresado con respecto a distintos aspectos o características de la entidad sobre la que se opina (nivel de aspecto). Sin embargo, estas tareas o estrategias obtienen información puntual y dispersa sobre distintos aspectos y sobre sujetos de opinión (los autores de las opiniones) de forma individualizada, lo que hace que en un conjunto de opiniones con respecto a una entidad estos enfoques por sí solos sean poco informativos. De esta manera, surge la necesidad de sintetizar esta información dispersa, obteniendo así un conocimiento mucho más general y más legible de cara al usuario.

La Síntesis de Opiniones (SO) [58] tiene como objetivo sintetizar o agregar la opinión reflejada por varios sujetos opinadores con respecto a los distintos aspectos de una entidad en un conjunto de opiniones. De esta manera, la SO es capaz de ofrecer un conocimiento general al usuario, donde se ve sintetizada la

subjetividad expresada en múltiples opiniones. La SO puede realizarse mediante estrategias extractivas [4, 5], donde se realiza un resumen extrayendo las oraciones más relevantes, o mediante estrategias abstractivas [40, 17], donde las propias estrategias generan mediante texto los resúmenes. La SO está enfocada en dar información sobre qué se opina con respecto a distintos aspectos. Debido a esto, la SO comúnmente puede desgranarse en diversas subtareas como pueden ser: (1) detección de los aspectos mencionados en las opiniones, (2) obtención de la polaridad del sentimiento al nivel de granularidad deseado, y (3) empleo de técnicas para sintetizar la información.

La amplia complejidad del lenguaje puede provocar que una misma expresión suponga una polaridad distinta dependiendo del dominio en el que aparece. Por ejemplo, decir que una película es "muy rápida" puede suponer una polaridad negativa, mientras que usar esta misma expresión opinando sobre el tiempo de carga de un ordenador supondría una polaridad positiva. Esta complejidad del lenguaje, sumada a la limitación de los modelos de aprendizaje automático, se ve reflejada cuando un modelo es evaluado en un dominio o género[1] de texto distintos a los que el modelo está especializado, mostrando un claro deterioro en los resultados. De esta manera, surge el problema de adaptación al dominio en AO [81].

El problema de la adaptación al dominio es recurrente en el campo del PLN y, después de varios años, sigue siendo objetivo de estudio en gran número de trabajos de investigación [9, 14, 28, 36]. En el recorrido de esta tesis, aportamos propuestas que dan solución a este problema, siendo de esta manera el problema de adaptación al dominio un elemento que motiva y une gran parte de esta tesis.

La mayoría de aproximaciones de SO son propuestas que realizan SO mediante la generación o extracción de texto. Nosotros, sin embargo, planteamos la hipótesis de que es posible proporcionar resúmenes de opiniones estructurados de manera que estos sean independientes del dominio, interpretables y explicables. De esta forma, un modelo o metodología que cumpliera con estos requisitos sería por un lado más accesible debido a su independencia del dominio y, por otro, más interpretable para el ser humano y con una respuesta más explicable y justificada.

1.2 OBJETIVOS

El objetivo principal de esta tesis es contribuir a la generación de resúmenes de opiniones. Sin embargo, atendiendo a los mencionados desafíos emergentes de la IA relacionados con la democratización y transparencia de la IA, enfocamos

1 Llamamos género de texto a la naturaleza del texto en cuestión: opiniones, *micro-blogging*, literario, etc.

nuestras contribuciones de manera que los resúmenes generados sean explicables y fácilmente interpretables por parte del ser humano, además de aportar una solución fácilmente adaptable a distintos dominios. Para lograr este objetivo, nos planteamos los siguientes subobjetivos:

1. Aplicar técnicas de combinación de clasificadores para ofrecer una solución al reto de la adaptación al dominio para la SO y el AO.
2. Poner a disposición de la comunidad científica recursos lingüísticos de libre acceso para la evaluación de sistemas y metodologías de SO. Estos recursos deben tener en cuenta las distintas tareas que pueden conformar un sistema de SO.
3. Realizar una metodología que sea capaz de generar síntesis de opiniones estructuradas, interpretables y explicables gracias al empleo de técnicas de IA transparentes. Además, esta metodología debe ser flexible en la adaptación a distintos dominios.

1.3 ESTRUCTURA DE LA TESIS

En esta tesis los capítulos se asocian a los distintos objetivos expuestos y a los conocimientos preliminares necesarios para el desarrollo de los mismos. De esta manera, la tesis se estructura como:

CAPÍTULO 2.
: Las propuestas realizadas en esta tesis tienen en común, aparte de dar una solución al problema de adaptación al dominio, el empleo de técnicas poco comunes en el campo del PLN. Por ello, en el capítulo 2 detallamos las distintas técnicas de IA empleadas para las aproximaciones propuestas en esta tesis.

CAPÍTULO 3.
: En el capítulo 3 hacemos un repaso del campo del AO, detallando desde cómo representar texto para que sea computacionalmente tratable hasta los distintos niveles de complejidad del AO y los distintos aspectos característicos de esta tarea.

CAPÍTULO 4.
: Basándonos en los objetivos previamente expuestos, en el capítulo 4 presentamos la primera contribución realizada en esta tesis. Se trata una metodología cuyo fin es la optimización de la clasificación de la polaridad del sentimiento mediante la combinación de Modelos de Clasificación de la Opinión (MCO). Estos métodos son herramientas o recursos preentrenadas en distintos dominios que clasifican la polaridad de los textos dados como entrada. Nuestra metodología realiza un agregado donde cada MCO tie-

ne una contribución o ponderación optimizada por medio de un algoritmo evolutivo. De esta manera, se realiza una adaptación al dominio mediante la combinación ponderada de distintos MCO, donde un MCO obtendrá una mayor ponderación en dominios similares a los que este ha sido entrenado.

CAPÍTULO 5.
En la literatura existe una escasez de conjuntos de opiniones válidos y de calidad con los que se puedan obtener resúmenes realmente valiosos. Esto se debe a que para poder evaluar un flujo de una metodología de SO en su totalidad, es necesario que el conjunto de opiniones esté agrupado por entidad para que así los resúmenes tengan valor en sus conclusiones. En el capítulo 5 presentamos un conjunto de opiniones sobre una única entidad en el dominio de restaurantes, enfrentándonos de esta manera al segundo subobjetivo de los tres marcados en esta tesis. Este conjunto de opiniones tiene como objetivo ser un recurso válido para la evaluación de metodologías de SO y sus distintas posibles subtareas, presentando una anotación a nivel de categorías de aspecto y de polaridad por oración.

CAPÍTULO 6.
Asociado a nuestro tercer objetivo y al objetivo principal de la tesis, en el capítulo 6 presentamos nuestra última contribución. Se trata de una metodología cuyo fin es generar síntesis de opiniones mediante la hibridación de técnicas de *Deep Learning* para la detección y agrupamiento de aspectos y técnicas de SD para la generación de resúmenes estructurados, interpretables y explicables. Además, proponemos una metodología que sigue un enfoque débilmente supervisado. De esta manera, nuestra contribución puede considerarse una propuesta metodológica fácilmente adaptable al dominio debido a que no requiere conjuntos de opiniones etiquetados para entrenar en distintos dominios.

CAPÍTULO 7.
Finalmente, en el capítulo 7 mostramos las conclusiones extraídas a partir del desarrollo de esta tesis, así como de distintas vías abiertas para el desarrollo de posibles trabajos futuros.

2

CONOCIMIENTOS PREVIOS: COMBINACIÓN, OPTIMIZACIÓN Y MINERÍA DESCRIPTIVA

En esta tesis nos planteamos determinados subobjetivos dentro de la Síntesis de Opiniones y realizamos propuestas que mejoren la adaptación al dominio y la interpretabilidad y explicabilidad de propuestas actuales. Las propuestas que realizamos tienen en común el empleo de técnicas de la inteligencia artificial poco frecuentes en el campo del Procesamiento del Lenguaje Natural, como son el empleo de métodos de combinación de modelos de aprendizaje, algoritmos evolutivos y técnicas de descubrimiento de subgrupos. En este capítulo, presentamos conceptos generales de estas técnicas de Inteligencia Artificial para introducir al lector a conceptos que serán recurrentemente utilizados a lo largo de la tesis.

2.1 INTRODUCCIÓN

La IA abarca un gran conjunto de tareas que, a su vez, dan lugar a técnicas y estrategias de distinta naturaleza para dar soluciones a estas tareas. Sin embargo, el empleo de estas técnicas no son excluyentes entre sí, sino que el empleo de técnicas poco frecuentes en determinadas tareas o ámbitos pueden dar lugar a una visión *out of the box*. Esto nos lleva a la posibilidad de aportar soluciones originales a problemas gracias al empleo de técnicas contrastadas en otros ámbitos de la IA.

En este capítulo introducimos algunos campos de la IA ajenos o no tan asociados al PLN debido al empleo de técnicas de estos ámbitos para realizar las contribuciones presentadas en esta tesis. En la sección 2.2 presentamos qué es un método de combinación de modelos predictivos y mostramos distintos enfoques existentes en la literatura. En la sección 2.3 hacemos una descripción de qué es un problema de optimización y cómo los Algoritmos Evolutivos (AEs) son una solución a este tipo de problemas. Estos dos campos de la IA están relacionados directamente con una de nuestras contribuciones presentada en el capítulo 4. Finalmente, en la sección 2.4 realizamos una descripción de qué es la Minería de Patrones Descriptiva Supervisada (MPDS) y el Descubrimiento de Subgrupos (SD por sus iniciales en inglés de *Subgroup Discovery*), técnicas empleadas en esta tesis con el fin de generar síntesis de opiniones estructuradas, interpretables y explicables.

2.2 MÉTODOS DE COMBINACIÓN DE MODELOS DE APRENDIZAJE

El empleo de modelos de clasificación y regresión de forma individual puede llevar a sesgos propios de estos modelos y reducir la generalización de las respuestas de los mismos [119]. De forma similar a la que un paciente prefiere tener varias opiniones médicas con respecto a un cuadro sintomático para así tener una respuesta más robusta, una respuesta a un problema de predicción puede ser más robusta al combinar respuestas de distintos modelos. Esto hace que nazca la idea de combinar distintas respuestas de distintos métodos con el fin de producir una respuesta más robusta y con mayor capacidad de generalización.

En diversos estudios se demostró cómo la combinación de distintos modelos predictivos incrementan la capacidad de generalización [10, 105] y mejora los resultados de modelos de forma individual. Esto hace que se investiguen distintas formas de combinación de estos modelos. En la sección 2.2.1 haremos un repaso de los distintos esquemas de combinación de modelos predictivos de la literatura. Posteriormente, en la sección 2.2.2 veremos aplicaciones de estas estrategias en AO.

2.2.1 *Esquemas para la combinación de modelos predictivos*

El estudio de la combinación de modelos predictivos ha llevado a distintos esquemas para la combinación de los mismos. Basándonos en [144], nosotros clasificamos estos esquemas o estrategias en tres grupos:

- **Agregación de modelos de distinta naturaleza.** Se trata del entrenamiento de distintos modelos predictivos de distinta naturaleza y realizar una agregación de las predicciones de estos. En clasificación esta agregación se puede realizar mediante voto mayoritario y en regresión mediante mecanismos de agregación como la media, mediana, etc. También, existen otras maneras más complejas de agregar las respuestas, como es mediante un sistema de Stacking [141], donde las respuestas de los modelos son agregadas por otro meta-clasificador que usa las respuestas de los clasificadores como características de entrada.
- **Bagging [19].** Esta estrategia consiste en el entrenamiento de clasificadores sobre distintos muestreos de datos obtenidos de forma aleatoria con reemplazamiento. En esta estrategia de combinación, la diversidad no viene dada por los algoritmos, el cual siempre es el mismo, sino que viene dada por los distintos conjuntos de entrenamiento empleados en cada uno de los clasificadores gracias al muestreo aleatorio. Las estrategias de Bagging suelen decidir la respuesta final mediante voto mayoritario o agregaciones de variables continuas para problemas de regresión. Un ejemplo clásico de algoritmo basado en Bagging es Random Forest [20], un algoritmo predictivo consistente en el entrenamiento de un número configurable de árboles de decisión sobre muestreos con reemplazamiento del conjunto de entrenamiento.
- **Boosting [133, 46].** La estrategia de Boosting se basa en el entrenamiento de varios modelos, donde el modelo m_i depende del modelo previamente entrenado m_{i-1} y, a diferencia de Bagging donde el muestreo de datos se realiza de forma aleatoria, en Boosting se realiza un muestreo mediante la ponderación de las muestras más complicadas según los modelos previamente entrenados. Además, la respuesta final se trata de una agregación ponderada donde cada modelo recibe un peso asociado a su error individual.

2.2.2 *Combinación de modelos en Análisis de Opiniones*

La combinación de modelos es un esquema muy empleado en muchos dominios distintos. En AO existen un diverso número de propuestas en las que se emplean estos métodos. Para la clasificación de la opinión, encontramos trabajos como el presentado en [71], donde los autores proponen la combinación de un método para la clasificación basado en la aparición de términos positivos o negativos y un SVM, demostrando que la combinación de ambos ofrece mejo-

res resultados que los modelos de forma individualizada. En [93] los autores proponen un sistema basado en voto mayoritario para la clasificación de tweets en español. En [95] realizan un *stacking* de métodos para la clasificación no supervisados para la clasificación de la opinión en textos en español. En [94] los autores realizan una combinación de métodos supervisados y no supervisados mediante voto mayoritario para la clasificación de la opinión en textos en español. Más recientemente, [70] explora distintas estrategias para la combinación de clasificadores heterogéneos para la clasificación de la opinión del sentimiento en diversos dominios.

Por otro lado, otro paso clave en el AO es la detección de aspectos. En este dominio, también se han realizado diversas propuestas empleando métodos de combinación de modelos. En [115], los autores combinan un modelo de Deep Learning basado en capas convolucionales con una serie de reglas lingüísticas para la extracción de aspectos a nivel de término. También en [72] los autores proponen la combinación de un modelo basado en Deep Learning y un SVM para la tarea de análisis de opiniones a nivel de aspecto de Semeval2016 [142], tanto para clasificación de la opinión como para la detección de categorías de aspectos.

2.3 OPTIMIZACIÓN Y ALGORITMOS EVOLUTIVOS

Los problemas de optimización son problemas comunes en el campo de la computación y la IA en general y forman parte del complejo proceso de toma de decisiones en diversos dominios. Un problema de optimización es definido por [126] como un par (S,f), donde S representa el conjunto de soluciones posibles del problema y $f : S \rightarrow \mathbb{R}$ es la función objetivo a optimizar, asignando a cada $s \in S$ un valor numérico real (*fitness*) asociado a la calidad de s para resolver el problema. El principal objetivo de un problema de optimización es la búsqueda de una solución s^* que sea un óptimo global.

Un ejemplo de problema de optimización es el conocido problema del Viajante de Comercio (*Travelling Salesman Problem* o *TSP*). En este problema se trata de buscar un recorrido óptimo para un hipotético viajante de comercio a lo largo de n ciudades teniendo en cuenta un valor de coste de viaje (como las distancias) entre cada una de las ciudades. En ese caso, una solución s sería una lista de las n ciudades ordenadas según el orden de visita de estas y la función f sería la sumatoria de los costes de viaje. Este problema es un problema de optimización combinatoria [15], suponiendo esto que cuando n es un número grande, los tiempos de cómputo se disparan.

Ante la existencia de problemas cuyo espacio de búsqueda es muy grande o problemas de optimización continua donde el espacio de búsqueda es infinito,

es necesario el planteamiento de estrategias para afrontarlos, de manera que estas estrategias proporcionen una solución cercana al óptimo global. Entre las estrategias para afrontar este tipo de problemas nos encontramos con las metaheurísticas, las cuales se describen en la sección 2.3.1. Entre estas metaheurísticas aparecen estrategias bioinspiradas y algoritmos basados en la evolución natural. En las secciones 2.3.2 y 2.3.3, describimos dos tipos de algoritmos evolutivos empleados en esta tesis: los algoritmos genéticos (AGs) y los algoritmos de Evolución Diferencial (ED), respectivamente. Posteriormente, en la sección 2.3.4 describiremos dos algoritmos de ED empleados en una de las contribuciones de esta tesis. Finalmente, en la sección 2.3.5 hacemos un repaso de otras propuestas en PLN donde se han empleado algoritmos evolutivos, técnicas raramente usadas en este campo de la IA.

2.3.1 *Metaheurísticas: Estrategias para optimización*

La existencia de problemas de optimización cuya complejidad supondría tiempos de cómputo muy grandes o, incluso, problemas cuyo espacio de búsqueda es infinito, supone la necesidad de estrategias que aporten soluciones de calidad a estos problemas en unos tiempos asumibles computacionalmente. De esta necesidad, surgen las metaheurísticas.

Para entender qué es una metaheurística, término empleado por primera vez en [52], es necesario analizar la propia morfología de la palabra. Esta deriva del verbo griego *heuriskein*, que significa "encontrar" y la palabra griega *meta* que se refiere a "desde un nivel superior". Es decir, las metaheurísticas son algoritmos estocásticos de búsqueda generales de soluciones, pudiendo adaptarse a cualquier problema de optimización. Formalmente, una metaheurística es definida por [15] como *estrategias para la exploración de espacios de búsqueda empleando distintos métodos* de manera que estas mantengan un equilibrio dinámico entre diversificación (explorar distintas regiones del espacio de búsqueda) e intensificación (explotar regiones prometedoras del espacio de búsqueda). De manera general y basándonos en la definición de las metaheurísticas, estas tienen como elementos básicos (1) una representación o codificación de una solución al problema, (2) una función objetivo que asociará cada solución a una calidad de la misma, (3) mecanismos para explorar el espacio de búsqueda y (4) mecanismos para explotar regiones prometedoras del espacio de búsqueda.

Existen distintas formas de clasificar una metaheurísca basándose en distintas características de las mismas. Esta clasificación puede realizarse atendiendo a si son inspiradas en la naturaleza o no, si son basadas en poblaciones o en búsqueda a partir de un único punto, si se basan en una o varias poblaciones, etc. Nosotros, como en [15], clasificamos las metaheurísticas en basadas en trayectorias o basadas en poblaciones:

- **Metaheurísticas basadas en trayectorias.** Son algoritmos que reciben este nombre por su proceso de búsqueda, basado en una trayectoria dentro del espacio de búsqueda. Estos algoritmos inician desde un estado inicial y van describiendo una trayectoria en el espacio de estados. Entre estos algoritmos encontramos algoritmos de búsqueda local, basados en ir encontrando mejoras de forma iterativa hasta llegar a un óptimo local. Con el objetivo de evitar estos óptimos locales, aparecen nuevas estrategias como el algoritmo de Enfriamiento Simulado [74], considerado una de las primeras metaheurísticas y basado en estadística mecánica. Este algoritmo permite salir de óptimos locales mediante el movimiento hacia soluciones peores para explorar otros estados dentro del espacio de estados. Otro algoritmo basado en trayectorias muy citado y empleado es la Búsqueda Tabú, presentado por primera vez en [52]. Este algoritmo se basa en un mecanismo de memoria por el que, en una lista llamada *lista tabú*, se almacenan los últimos movimientos realizados en la búsqueda con el fin de no entrar en óptimos locales y evitar ciclos dentro de la búsqueda. Para más información sobre metaheurísticas basadas en trayectorias, consultar [15] y [143].
- **Metaheurísticas basadas en poblaciones.** Estas metaheurísticas, a diferencia de las basadas en trayectorias, en cada iteración trabajan sobre un conjunto de soluciones (población) en lugar de una única solución, consistiendo cada iteración de estos algoritmos en la evolución o reemplazo de la población. En [143] destacan dos métodos de optimización dentro de las metaheurísticas basadas en poblaciones. Estos son los basadas en colonias de hormigas y los basados en computación evolutiva. Los algoritmos basados en colonias de hormigas [39] se basan en el empleo de feromonas en colonias de hormigas y son métodos especialmente enfocados en encontrar caminos mínimos. Parten de un conjunto de hormigas artificiales que tienen que generar un camino óptimo dentro de un grafo. Estas hormigas artificiales son mecanismos probabilísticos de construcción de soluciones del problema que mantienen memoria de los nodos del grafo visitados, toman decisiones sobre los siguientes nodos a visitar basadas en una regla probabilística de transición y "aportan feromona" en soluciones prometedoras. Existen muchas otras metaheurísticas basadas en poblaciones que se inspiran en los procesos naturales de animales[1].

  Por otro lado, están las metaheurísticas basadas en computación evolutiva. Estas metaheurísticas se basan en la evolución natural, partiendo de una población inicial de individuos (conjunto de soluciones). Esta población sufre a lo largo del proceso iterativo combinaciones entre ellas (simulando reproducción) y mutaciones, además de un proceso de selección donde "sobreviven" las soluciones más fuertes (simulando selección natural). Entre las estrategias evolutivas más empleadas están los AGs, estrategias introduci-

1 Para información más detallada, consultar [101].

das en la sección 2.3.2 y algoritmos de ED, los cuales se introducirán en la sección 2.3.3.

2.3.2 *Algoritmos Genéticos y algoritmos meméticos*

Los AGs son algoritmos de optimización estocásticos que son construidos basados en la evolución natural [53, 143]. Son metaheurísticas que parten de una población aleatoria inicial, ofreciendo de esta manera cierta diversidad a la búsqueda, que de forma iterativa sufre transformaciones y evoluciones de manera que el algoritmo tenderá a converger a regiones cercanas a un óptimo de calidad.

Los elementos más básicos de un algoritmo genético son los cromosomas, que a su vez están formados por genes, y una función objetivo que evalúa la bondad de estos cromosomas con respecto al entorno (problema de optimización). Un cromosoma es simplemente una codificación de una solución al problema y un gen es la unidad mínima de representación. Por ejemplo, en un problema de optimización de parámetros continuos, un cromosoma c podría ser un vector de números reales, donde cada número del vector es un gen. En la figura 1 mostramos un esquema de las etapas de un algoritmo genético, también descritas detalladamente a continuación en el mismo orden en el que se desarrollan [99]:

1. **Esquema de selección basado en torneo.** Este operador selecciona los cromosomas o soluciones que participarán en la siguiente generación. Esta nueva población será sobre la que se aplicará el resto de etapas. Un posible esquema de selección puede ser el torneo binario, por el que se comparan parejas aleatorias de individuos de los que solo sobrevivirá uno de los dos. Esta etapa simula el proceso de selección natural, donde los cromosomas de mayor calidad tendrán una mayor probabilidad de participar en la siguiente generación que los cromosomas de menos calidad.
2. **Operador de cruce.** Se trata del operador que modela la reproducción sexual o combinación de cromosomas. Este operador actúa con una probabilidad determinada con el fin de generar un máximo de n hijos (siendo n el tamaño de la población) que sustituirán a individuos de la nueva población. Un ejemplo de operador de cruce sería realizar la media de dos cromosomas, gen a gen, en el caso de un problema de optimización continua.
3. **Operador de mutación.** Este operador muta de forma aleatoria algunas componentes o genes de los individuos con el fin de explorar distintos subespacios del espacio de búsqueda, aportando así capacidad de exploración a la búsqueda. Estos operadores actúan bajo una probabilidad dada como parámetro configurable. Un posible operador de mutación podría ser agregar un valor aleatorio a un gen aleatorio.

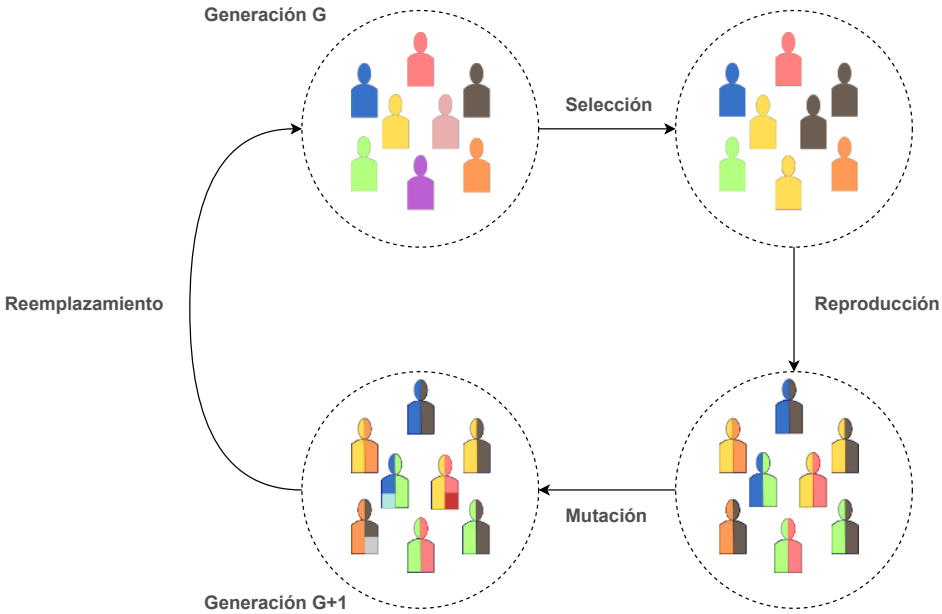

Figura 1: Esquema general de un algoritmo genético. Un genético parte de una generación G, a la cual se le realiza un proceso de selección donde los mejores individuos tendrán más probabilidad de permanecer en la población. Se realiza un cruce entre individuos (reproducción) y finalmente algunos individuos sufren mutaciones, dando lugar a la generación G+1.

4. **Esquema de reemplazamiento.** Define cómo una nueva población sustituirá la antigua población en la siguiente generación (iteración del algoritmo). Un posible esquema de reemplazamiento podría ser sustituir toda la población anterior por la nueva y mantener los m mejores cromosomas de la población anterior.

Los algoritmos genéticos son algoritmos que exploran muy bien el espacio de búsqueda. Sin embargo, su gran nivel de exploración contrasta con su pobre nivel de explotación de espacios prometedores, pudiendo esto llevar al algoritmo a que no converja de manera clara a un óptimo global. Por otro lado, existen metaheurísticas caracterizadas por un gran nivel de explotación. Por ejemplo, los algoritmos de búsqueda local son buenos métodos para la explotación local de regiones del espacio de búsqueda.

Debido a esta necesidad de mejorar la explotación por parte de los algoritmos genéticos, nacen los algoritmos meméticos (AM), métodos de optimización empleados en la experimentación de una de las contribuciones realizadas en esta

tesis descrita en el capítulo 4 y definidos en [102] como *una población de agentes que alternan periodos de auto-mejora (con búsqueda local) con periodos de cooperación (con recombinación) y competición (fase de selección)*. Por lo tanto, se puede definir un algoritmo memético como una hibridación de algoritmos evolutivos, generalmente algoritmos genéticos, con algoritmos de búsqueda local, de manera que así a la gran capacidad de exploración de un algoritmo genético se le sume la gran capacidad de explotación de un algoritmo de búsqueda local. Esta hibridación se suele llevar a cabo empleando una búsqueda local sobre algunos de los cromosomas de la población cada cierto número de generaciones o iteraciones del algoritmo genético.

2.3.3 *Evolución Diferencial*

La Evolución Diferencial (ED) [124] es un método de optimización evolutivo muy robusto para optimización de parámetros continuos. Estos métodos son modelos evolutivos que enfatizan en la mutación de los individuos, proporcionando una gran capacidad de exploración, aspecto clave para la optimización de parámetros continuos.

Las estrategias basadas en ED parten de una población de vectores de parámetros reales $x_i, i \in \{0, ..., NP-1\}$, de tamaño NP y dimensión D, que son inicializados de forma aleatoria y son parte de cada generación G. Un algoritmo de ED se compone de las siguientes etapas, descritas en el mismo orden en el que tienen lugar en el proceso de cada generación o iteración:

1. **Mutación.** El operador de mutación genera una variación del vector objetivo de solución x_i mediante una serie de operaciones entre varios vectores solución de la población. Estas operaciones vienen determinadas por el esquema empleado en el algoritmo, existiendo diversos esquemas en la literatura [30]. Por ejemplo, en el esquema general de mutación, también conocido como DE/Rand/1, uno de los primeros esquemas de ED, el vector mutado v_i viene determinado mediante la suma de un vector x_{r1} y la diferencia de dos vectores (x_{r2} y x_{r3}) ponderada por un parámetro configurable constante F (ver ecuación 1).

$$v_i = x_{r1} + F(x_{r2} - x_{r3}) \tag{1}$$

   donde los índices r1, r2 , r3 $\in \{0, ..., NP-1\}$ son elegidos de forma aleatoria.

2. **Reproducción.** Para cada $x_i \in G$, se genera un vector hijo u_i, participando en esta generación tanto el vector objetivo x_i como su vector mutado v_i. Esta etapa sigue el esquema mostrado en la ecuación 2.

$$u_{ji,G+1} = \begin{cases} v_{ji,G+1} \text{ if}(randb(j) \leq CR) \, or \, j = rnbr(i) \\ x_{ji,G} \quad \text{if } (randb(j) > CR) \, and \, j \neq rnbr(i) \end{cases} \tag{2}$$

   Donde $j = 0, 1, ..., D-1$, $CR \in [0,1]$ es un parámetro de control, $randb(j)$ es un número aleatorio que verifica $randb(j) \in [0,1]$, $rnrb(i)$ es un entero aleatorio que verifica $rnrb(i) \in \{0, ..., D-1\}$ y G+1 se refiere a la siguiente generación.
3. **Selección.** Selecciona el vector u_i que formará parte de la próxima generación $G+1$. Consiste en la comparación del valor en la función objetivo de u_i y su vector objetivo o padre x_i. Si el *fitness* es superado, entonces u_i sustituirá a x_i en $G+1$.

2.3.4 *Algoritmos de Evolución Diferencial*

En la contribución realizada en esta tesis asociada al subobjetivo de proponer una solución a la adaptación al dominio, y descrita en el capítulo 4, se emplean dos algoritmos de ED para la combinación de clasificadores del sentimiento. Estos dos algoritmos son L-Shade y jSO, algoritmos contrastados en competiciones CEC[2] y que se describen a continuación:

- **L-Shade [127].** Es un algoritmo de ED que destaca por las siguientes características:
  - Los parámetros CR y F (descritos en la sección 2.3.3) son aprendidos por el propio algoritmo. Para ello, se crean unas memorias de un tamaño determinado donde se almacenan distintos valores de estos parámetros e inicializados a 0.5. El algoritmo aplica una modificación a estos parámetros basada en el éxito de los mismos a la hora de generar nuevos individuos. De esta memoria, para cada individuo se elegirán de forma aleatoria los parámetros que participarán en su transformación.
  - Se realiza una reducción lineal del tamaño de la población, partiendo de un tamaño inicial de la misma configurable y un tamaño de población mínimo también configurable. Esta reducción lineal de la población se hará de manera que, a pesar de esta reducción del tamaño de la población, se mantengan siempre los mejores individuos.
  - Se emplea el esquema de mutación current-to-$p$best/1 [91]:

$$v_{i,G} = x_{i,G} + F_i(x_{pbest,G} - x_{i,G}) + F_i(x_{r1,G} - x_{r2,G}) \tag{3}$$

2 https://cec2021.mini.pw.edu.pl/

donde $x_{i,G}$ es el elemento actual al que se le calcula el vector de mutación, $r1$ y $r2$ son índices aleatorios de la población y $x_{pbest,G}$ es un vector solución escogido entre los NP$\times p$ mejores vectores ($p \in [0,1]$). F_i se trata del parámetro F aleatoriamente elegido de la memoria de parámetros de Fs para el vector $v_{i,G}$.
  - Se realiza una reproducción similar a la básica presentada en la sección 2.3.3, donde participa el parámetro CR_i, elegido aleatoriamente de la memoria de CRs.
- **jSO [21].** Es un algoritmo basado en L-Shade que incorpora las siguientes novedades:
  - Las memorias para los parámetros CR y F son inicializadas a un valor más alto (0.8) para favorecer la exploración en las primeras iteraciones.
  - Se fuerza a que ningún valor de CR y F sea bajo en etapas tempranas del evolutivo.
  - Ambas memorias guardan un valor constante de 0.9 para dar la posibilidad de saltar a nuevas regiones del espacio de búsqueda independientemente de la etapa del algoritmo.
  - El esquema de mutación es el current-to$p$best/1 (igual que en L-Shade) pero con el empleo de un parámetro F_w con el fin de orientar la búsqueda hacia los mejores individuos en etapas avanzadas del algoritmo tal y como se muestra en la ecuación 4:

$$v_{i,G} = x_{i,G} + F_w(x_{pbest,G} - x_{i,G}) + F_i(x_{r1,G} - x_{r2,G}) \tag{4}$$

    siendo F_w calculado según la ecuación 5 y donde max_nfes es el máximo de evaluaciones de la función objetivo y $nfes$ el número actual de evaluaciones de la función objetivo.

$$F_w = \begin{cases} 0{,}7 \cdot F & nfes < 0{,}2 \cdot max\_nfes \\ 0{,}8 \cdot F & nfes < 0{,}4 \cdot max\_nfes \\ 1{,}2 \cdot F & nfes \geq 0{,}4 \cdot max\_nfes \end{cases} \tag{5}$$

  - El parámetro p para obtener $x_{pbest,G}$ en el operador de mutación current-to$p$best/1 es actualizado teniendo como referencia unos valores umbrales máximo (p_{max}) y mínimo (p_{min}) configurables.

$$p = \frac{p_{max} - p_{min}}{max\_nfes} \cdot nfes + p_{min} \tag{6}$$

2.3.5 *Algoritmos Evolutivos en Análisis de Opiniones*

Los AEs son técnicas poco empleadas en la literatura para solucionar problemas de PLN y, más concretamente, de AO. Sin embargo, existen diversas propuestas

en las que emplean este tipo de técnicas, normalmente con el fin de selección de características de texto.

En [1] los autores emplean AGs para la selección de características para un problema de clasificación de la opinión a nivel de documento. En [104] los autores realizan una combinación de selectores de características empleando para ello un AG, mostrando cómo su propuesta de combinación ofrece mejores resultados que empleando los selectores de forma individual. En [29] también emplean AGs con el fin de seleccionar las mejores características para la clasificación de opiniones en inglés y bengalí. Por otro lado, en [100] los autores crean una combinación no lineal de clasificadores SVM mediante programación genética pero siendo entrenados estos clasificadores en el mismo dominio.

Más recientemente, en [118] los autores emplean AEs para la optimización de hiperparámetros de un modelo *Deep Learning* para AO en *Twitter*. En [103] los autores proponen un sistema de soporte de decisión basado en AO sobre opiniones con respecto a la pandemia COVID-19 empleando AEs para la combinación de varios modelos SVM. En [34] los autores emplean un AE multi-objetivo con el fin de realizar una selección de características óptima en distintos problemas de AO.

2.4 MINERÍA DE PATRONES DESCRIPTIVA SUPERVISADA

El concepto de *patrón* en datos se puede definir como un conjunto de elementos que están relacionados dentro de una base de datos [136]. El descubrir patrones en datos puede dar lugar a conclusiones valiosas y, por lo tanto, una gran ventaja a la hora de la toma de decisiones a partir de datos.

La minería de patrones es un campo dentro de la ciencia de datos cuyo objetivo es encontrar patrones que se cumplan en subconjuntos de individuos, transacciones o instancias de dentro de una base de datos. En definitiva, el objetivo de un algoritmo de minería de patrones es, en su esencia, crear conocimiento a partir de datos "desordenados", obteniendo información sobre determinados subconjuntos de individuos que se comportan de forma similar con respecto a ciertos atributos. Un ejemplo de uso de este tipo de técnicas es el presentado en [45] donde los autores se enfrentan a un problema de *marketing* en el que quieren asociar la compra de determinados productos con la compra de marcas concretas.

En esta sección presentaremos técnicas de minería de patrones empleadas en esta tesis. Concretamente, en la sección 2.4.1 describiremos qué es la Minería de patrones descriptiva supervisada (MPDS), un campo de la minería de patrones que hibrida elementos característicos de tareas predictivas con elementos de tareas descriptivas. En la sección 2.4.2 describiremos una técnica del campo de

MPDS. Esta técnica es conocida como Descubrimiento de subgrupos (SD por su nombre en inglés *Subgroup Discovery*), y su fin es la generación de reglas que relacionan ciertos atributos con una propiedad de interés y es empleada en esta tesis para la generación de resúmenes de opiniones. Debido a las particularidades del concepto de SD, la evaluación de las reglas obtenidas a partir de algoritmos de SD no es trivial. Por ello, en la sección 2.4.3 describimos algunas de las métricas más empleadas en SD y, particularmente, las empleadas en esta tesis.

2.4.1 *Técnicas en minería de patrones descriptiva supervisada*

El análisis de datos se ha clasificado típicamente en tareas predictivas, formadas por modelos que aprenden a predecir una propiedad de interés basándose en datos anotados, y en tareas descriptivas, más relacionadas con la obtención de patrones que expliquen comportamientos de interés en datos no anotados. Sin embargo, existen dominios que requieren el empleo de ambos tipos de tareas. Por ello nace el MPDS, un campo que hibrida ambos tipos de tareas, predictivas y descriptivas, con el fin de obtener patrones en los datos en relación a una propiedad de interés.

Dentro del MPDS encontramos tres tareas principales. Una de ellas es la minería de conjuntos de contraste. Un conjunto de contraste [11, 37] es un patrón o conjunto de atributos y valores de los mismos, que muestran diferencias significativas entre grupos, siendo estos grupos excluyentes entre sí según una propiedad de interés para el usuario, que supondría el elemento "supervisado" de este tipo de tareas. Un algoritmo especializado en encontrar conjuntos de contraste es STUCCO [11] (*Search and Testing for Understandable Consistent Contrasts*) que obtiene conjuntos de contraste junto a sus soportes (frecuencias) y emplea tests estadísticos para descartar conjuntos de contraste de poca sinificancia estadística.

La segunda tarea dentro del MPDS es la de descubrimiento de patrones emergentes [38]. Un patrón emergente es aquel que incrementa de forma significativa su soporte de un conjunto de datos a otro. La búsqueda de estos patrones emergentes se realiza teniendo en cuenta el ratio de los soportes. Un ejemplo de algoritmo especializado en detección de patrones emergentes es iEPMiner [42], que selecciona patrones emergentes definidos como aquellos que tienen un soporte mínimo, un ratio de crecimiento mínimo y que están muy correlacionados según métricas estadísticas comunes.

Finalmente, la tercera tarea dentro del MPDS es el SD, que abarca un conjunto de técnicas para la detección de subgrupos de individuos que muestran cierto interés estadístico con respecto a una propiedad de interés. En la sección 2.4.2 damos más detalles sobre esta tarea.

2.4.2 *Descubrimiento de subgrupos*

El SD, presentado por primera vez en [75], es una tarea dentro de la MPDS cuyo fin es encontrar relaciones entre características que expliquen el comportamiento de subgrupos de individuos estadísticamente interesantes con respecto a una propiedad objetivo de interés [136]. Las técnicas de SD generan un conjunto de reglas y métricas que dan una descripción estructurada del dominio de estudio. Una regla de SD está formada por un antecedente o condición, que se trata de un conjunto de atributos y valores de los mismos, y un consecuente que será la propiedad de interés. Esta propiedad de interés sería la componente "predictiva" o "supervisada" de este tipo de técnicas.

Un método de SD enfoca su objetivo en dar una visión explicativa de cómo ciertos atributos o características afectan a una propiedad objetivo en subgrupos de individuos de una base de datos. Para ello, un método de SD ofrece como salida un conjunto de reglas, donde una regla puede expresarse como $R : Cond \implies Target_{value}$ siendo $Cond$ el antecedente de la regla y $Target_{value}$ el consecuente que hace referencia a la propiedad de interés y a un valor particular de la misma. Las reglas generadas por un método de SD vienen además acompañadas de métricas que dan información sobre cómo de interesantes son estas reglas a nivel estadístico. De esta manera, las técnicas de SD pueden considerarse técnicas algorítmicamente transparentes, debido a que sus reglas o patrones son explicadas mediante métricas fácilmente interpretables por humanos. Este hecho ha provocado que se hayan empleado técnicas de SD para el desarrollo de modelos explicables [85].

Según [136], existen tres tipos algoritmos especializados en SD: los basados en clasificadores, los basados en algoritmos de reglas de asociación y los basados en AEs. Nosotros, además, incluimos una categoría más de métodos basados en *clustering*. A continuación hacemos una descripción de estos cuatro tipos de algoritmos de SD:

- **Algoritmos basados en clasificadores.** Como ya se ha mencionado, el SD puede interpretarse como una hibridación de técnicas predictivas con técnicas descriptivas. Esto da lugar a que varios algoritmos de SD se basen en modelos predictivos clásicos, concretamente clasificadores. Por ejemplo, el algoritmo *Expert-guided SD* [47], que se trata de una variación del algoritmo *beam search*. Otro ejemplo es CN2-SD [80], basado en el algoritmo CN2, un algoritmo clásico de clasificación basado en reglas.
- **Algoritmos basados en algoritmos de reglas de asociación.** La minería de reglas de asociación es similar al SD en el sentido de que ambos se basan en la búsqueda de reglas que relacionan propiedades entre sí. Por ello, existen adaptaciones de algoritmos de minería de reglas de asociación, como

por ejemplo SD-Map [8], un algoritmo de búsqueda exhaustivo que emplea umbrales de soporte para reducir el espacio de búsqueda. Por otro lado, basado en el clásico algoritmo para reglas de asocicación Apriori [2], nace Apriori-SD [69], algoritmo empleado en la experimentación de una de las contribuciones de esta tesis presentada en el capítulo 6. Apriori-SD incorpora con respecto a Apriori un mecanismo de posprocesamiento que filtra las reglas cuyo consecuente no es la propiedad de interés. Además, aparte de tener en cuenta valores umbral de confianza y soporte, se emplea un esquema de ponderación y una modificación de la medida WRAcc[3] con el fin de obtener reglas de calidad.

- **Algoritmos evolutivos.** Se trata de enfoques evolutivos para la obtención de reglas donde la codificación de soluciones se adapta al SD. Estos algoritmos tienen como función objetivo la optimización de determinadas métricas clásicas de SD. MESDIF [33] y NMEEF-SD [27] son dos ejemplos de algoritmos evolutivos pra SD. Concretamente, NMEEF-SD (*Non-dominated Multiobjective Evolutionary Algorithm for Extracting Reglas in Subgroup Discovery*), al igual que Apiori-SD, es empleado en la experimentación asociada a nuestra contribución presentada en el capítulo 6. Se trata de un algoritmo evolutivo multi-objetivo para SD basado en sistemas difusos. Este algoritmo evolutivo adapta los elementos clásicos de los algoritmos evolutivos (mutación, cruce, selección, etc.) al problema de SD y emplea una optimización multi-objetivo para optimizar varias métricas de calidad de SD. Este método acepta características numéricas sin que estas deban ser discretizadas previamente, debido a que son procesadas con un sistema difuso.
- **Aproximaciones basadas en clustering.** Son adaptaciones de algoritmos de clustering para enfocarlos al SD. Por ejemplo, en [145] se propone un algoritmo que extrae subgrupos mediante el empleo de lo que ellos llaman *cluster-grouping*. En [130] se propone un método basado en el algoritmo clásico de clustering *k-medoids* para enfrentarse al problema de SD.

2.4.3 *Evaluación en Subgroup Discovery*

El SD tiene la particularidad de ser un campo que hibrida elementos característicos de técnicas predictivas con elementos característicos de técnicas descriptivas. Por ello, es necesario el empleo de métricas que aporten información de tanto la capacidad predictiva, aunque esta no sea prioritaria en este tipo de técnicas, como la capacidad descriptiva.

Dado el conjunto de todos los individuos de una base de datos I, el subconjunto de individuos I_{Cond} en los que se cumple el antecedente $Cond$ de una regla, el subconjunto de individuos $I_{Target=value}$ en los se cumple que la propiedad

3 La métrica WRAcc es definida en la sección 2.4.3

de interés *Target* toma el valor *value* y siendo $|X|$ la cardinalidad del conjunto X, definimos las siguientes métricas para la evaluación y cuantificación de una regla:

- **Soporte.** Mide la frecuencia de la coaparición de los antecedentes y los consecuentes. Visto desde el punto de vista predictivo, también se puede definir como la frecuencia de instancias correctamente clasificadas según la regla. El soporte se puede calcular como:

$$\textbf{Soporte}(\mathbf{R}) = \frac{|I_{Target=value} \cap I_{Cond}|}{|I|}$$

- **Confianza.** Se define como la probabilidad de obtener el consecuente si el antecedente es cierto:

$$\textbf{Soporte}(\mathbf{R}) = \frac{|I_{Target=value} \cap I_{Cond}|}{|I_{Cond}|}$$

- ***Normalised Weighted Relative Accuracy* (NWRAcc).** Es una modificación de la métrica *Weighted Relative Accuracy* (WRAcc) o *Unusualness* (inusualidad), que es definida como el equilibrio entre la cobertura de la regla:

$$Cob(R) = \frac{|I_{Cond}|}{|I|}$$

y su ganancia de *Accuracy*:

$$AccG(R) = Conf(R) - \frac{|I_{Target=value}|}{|I|}$$

$$WRAcc = Cob(R) \cdot AccG(R)$$

La versión normalizada viene definida por la siguiente expresión:

$$\textbf{NWRAcc} = \frac{WRAcc(R) - LB_{WRAcc}}{UB_{WRAcc} - LB_{WRAcc}}$$

donde

$$UB_{WRAcc} = \frac{|I_{Target=value}|}{|I|} \cdot \left(1 - \frac{|I_{Target=value}|}{|I|}\right)$$

y

$$LB_{WRAcc} = \left(1 - \frac{|I_{Target=value}|}{|I|}\right) \cdot \left(0 - \frac{|I_{Target=value}|}{|I|}\right)$$

Según [26], un valor de NWRAcc mayor de 0.5 puede ser considerado como un buen nivel de inusualidad de la regla.

- **Significancia.** Mide la significancia de una regla como el ratio de probabilidad de la regla, donde V es el conjunto de todos los posibles valores de *Target*:

$$\mathbf{Sig}(\mathbf{R}) = 2 \cdot \sum_{k \in V} |I_{Target=k} \cap I_{Cond}| \cdot \log \frac{|I_{Target=k} \cap I_{Cond}|}{I_{Target=k} \cdot \frac{|I_{Cond}|}{|I|}}$$

Existen una gran cantidad de métricas para la evaluación de reglas de SD. Para más información al respecto, consultar [56] y [136].

3

ANÁLISIS DE OPINIONES

La obtención de conocimiento a partir de opiniones puede favorecer a la toma de decisiones estratégicas por parte de distintos grupos o sectores de la sociedad. La cada vez más creciente cantidad de opiniones en la Web y la necesidad de procesarlas para obtener conocimiento a partir de ellas hace que nazca el campo del análisis de opiniones y, con él, distintas estrategias y tareas que permiten la obtención de conocimiento a partir de las mencionadas opiniones. En este capítulo de la tesis, viajaremos a lo largo de los distintos elementos que componen el campo del análisis de opiniones. Concretamente, primero haremos una descripción de cómo puede representarse el texto de una opinión de manera que esta sea interpretable computacionalmente. Posteriormente, hablaremos del análisis de opiniones a nivel de aspecto y sus distintas tareas. Finalmente, describiremos distintas propuestas del estado del arte para obtener resúmenes o síntesis a partir de conjuntos de opiniones.

3.1 INTRODUCCIÓN

Gracias a la aparición de la Web 2.0, surgen nuevos paradigmas de comunicación, interacción social, comercio, etc. que suponen, a su vez, grandes fuentes de información. Un evidente ejemplo de fuente de información es el conjunto de opiniones dadas por usuarios o clientes con respecto a un producto, objeto o cualquier otro tipo de entidad por medio de texto. Obtener conocimiento a partir de estas opiniones puede suponer una ventaja competitiva para fabricantes o vendedores así como una información valiosa y de utilidad para clientes. El AO es un campo del PLN en el que se desarrollan distintas técnicas y tareas para la obtención de conocimiento a partir de opiniones.

Una opinión puede definirse como la expresión o valoración subjetiva de un sujeto opinador con respecto a los distintos aspectos o características de una entidad, objeto o producto. En [81], Bing Liu define formalmente una opinión en el contexto computacional como una quintupla $(e_i, a_j, p_k, h_l, t_z)$, donde e_i es la entidad sobre la que se opina, a_j es el aspecto sobre el que se opina (distintas características del producto), p_k es la polaridad del sentimiento expresada en la opinión, h_l es el sujeto opinador (el usuario que realiza la opinión) y t_z es el momento temporal en el que se realiza la opinión. Esta definición formal computacionalmente de una opinión sirve como base para las distintas tareas relacionadas con el AO.

A la hora del tratamiento computacional de opiniones es necesario, como en otras tareas de PLN, realizar una representación del texto de manera que esta sea interpretable computacionalmente. En la sección 3.2 realizamos un repaso de la evolución de las distintas formas existentes de representar términos o palabras. El AO se puede realizar a distintos niveles de granularidad y complejidad, entre ellos a nivel de aspecto. En la sección 3.3 presentamos el análisis de opiniones a nivel de aspecto o ABSA por sus iniciales en inglés (*Aspect-Based Sentiment Analysis*) y describimos las distintas tareas que lo conforman. Finalmente, en la sección 3.4 definimos qué es la síntesis de opiniones y realizamos una revisión de propuestas para esta tarea siguiendo distintas aproximaciones.

3.2 REPRESENTACIÓN Y CARACTERIZACIÓN DE TEXTO

El lenguaje, tanto escrito como hablado, es posiblemente el elemento más característico del ser humano. Sin embargo, las propiedades intrínsecas del lenguaje hacen que su tratamiento computacional no sea trivial. Estas propiedades están relacionadas con la semántica de los términos según el contexto y el dominio, el empleo de emoticonos, de ironías y sarcasmos, figuras literarias, etc.

En esta sección hacemos una revisión de las distintas formas de representación de texto existentes de manera que pueda ser tratado computacionalmente manteniendo la semántica del mismo y la evolución de las mismas. Concretamente, en la sección 3.2.1 describimos las primeras aproximaciones para representar texto de manera que este encapsule ciertas propiedades del lenguaje. Sin embargo, estas representaciones son representaciones fijas de términos que no tienen en cuenta el contexto en el que aparecen. En la sección 3.2.2 describimos representaciones del lenguaje dinámicas basadas en el contexto en el que aparecen los términos.

3.2.1 *Representación vectorial estática de palabras*

El tratamiento computacional de texto requiere una representación del mismo de manera que haga posible su procesamiento. Sin embargo, la representación de texto no es trivial y debe tener en cuenta ciertas propiedades intrínsecas al lenguaje, como pueden ser la similitud entre términos de un mismo dominio. La hipótesis distributiva del lenguaje [66, 54, 44], hace referencia a que dos palabras serán más parecidas cuanto más parecido es el contexto en el que aparecen. A partir de esta hipótesis, surge la semántica vectorial y la representación vectorial de palabras.

Una de las primeras representaciones de texto y la más simple es la representación como una matriz de co-ocurrencia, una matriz de tamaño $|V| \times N$, donde V es el vocabulario y N el número de documentos, representando cada casilla de la matriz el número de veces que aparece el término de esa fila en el documento de esa columna. Esta representación daría lugar a vectores de palabras de tamaño N. Otra representación común es la matriz de co-ocurrencia de términos. Esta representación parte de una matriz de dimensión $|V| \times |V|$ donde se cuentan las veces que aparecen dos términos del vocabulario en un mismo documento, oración o cualquier otra unidad de texto determinada por el usuario. De esta manera, cada palabra estaría representada por un vector numérico de dimensión $|V|$. Estas representaciones, sin embargo, conlleva ciertos problemas como es la no discriminación de ciertas palabras similares entre sí, como *camión* y *motocicleta*, de otras palabras muy frecuentes en texto como pueden ser *la*, *un*, *el* o *y*, entre otras.

Dando solución a este problema, aparece en escena la representación basada en la métrica *tf-idf* [87, 65]. Esta métrica se basa en asignar un valor numérico a cada término de un texto proporcionando menos peso a términos que aparecen de forma muy común en el texto y dando mayor importancia a términos que aparecen en pocos documentos. El tf-idf de una palabra en un documento se calcula como el producto de dos términos $tf \cdot idf$, donde $tf = log_{10}(count(t, d) + 1)$, siendo t el término o palabra y d el documento. El segundo término del

producto viene dado por la expresión $idf = log_{10}(\frac{N}{df_t})$, donde N es el número de documentos y df_t el número de documentos en los que aparece el término o palabra t [67]. Esta métrica es calculada para cada palabra en cada documento, obteniendo así una representación vectorial para cada palabra formada por estas métricas de tamaño N.

Las representaciones vectoriales descritas generalmente dan lugar a representaciones dispersas, es decir, las representaciones de palabras son vectores muy grandes donde la mayoría de componentes del vector son 0. Como solución a este problema, se introducen los *embeddings* [111, 67]. Los *words embeddings* son representaciones de palabras en forma de vector. Sin embargo, estos son vectores de menor dimensión y con mayor densidad, cuyas componentes no son fácilmente interpretable y que cumplen ciertas propiedades geométricas que se ven reflejadas al operar con ellos. Como ejemplo, en la figura 2 vemos una de las propiedades de los *embeddings* al operar con ellos. En la figura se aprecia como sumar el vector representación de la palabra *España* y el de la palabra *Capital* daría resultado el vector representación (o uno muy cercano) a la palabra *Madrid*.

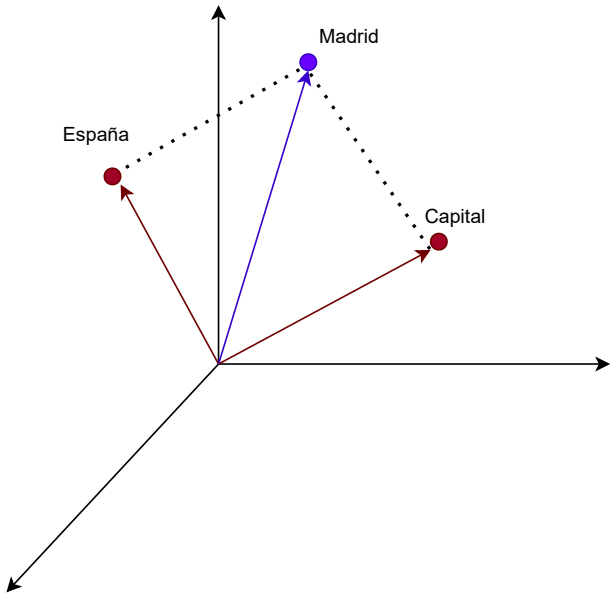

Figura 2: Ejemplo de propiedad de los *word embeddings*. Al realizar operaciones vectoriales sobre los vectores representación de palabras se dan resultados que muestran la semántica que estos vectores encapsulan. En este caso, sumar el vector *España* y el vector *Capital*, daría un resultado muy similar al vector *Madrid*.

El algoritmo Word2Vec [97, 98] se trata del primer algoritmo para la generación de *word embeddings* estáticos (un *embedding* fijo para cada palabra del vocabulario). Un modelo Word2Vec puede enfocarse mediante dos aproximaciones llamadas *skip-gram* y CBOW (*Continuous Bag of Words*). La aproximación *skip-gram* consiste en la construcción de un clasificador mediante una red neuronal con una capa densa oculta (de un tamaño dado como hiperparámetro) que responde a la pregunta "¿Cómo de probable es que la palabra w aparezca cerca de una palabra determinada?". Esta aproximación entrenaría la tarea de predecir el contexto dada una palabra de entrada. Por otro lado, la aproximación CBOW tiene una arquitectura similar pero difiere en la tarea, la cual consiste en predecir qué palabra aparecerá dado un contexto determinado. Una vez entrenada alguna de estas dos aproximaciones, los pesos de la capa densa oculta serían las representaciones vectoriales de las palabras.

A partir de word2vec nacen otras representaciones *word embeddings* como son Glove [109] o FastText [16], entre otras. Sin embargo, como ya se ha destacado, estos *word embeddings* tienen el problema de que tienen una representación fija para cada palabra, no teniendo en cuenta el contexto en el que aparecen estas [111]. Esta limitación abre la puerta a la investigación de nuevas alternativas más complejas que tengan en cuenta el contexto en el que aparece una palabra a la hora de la representación de la misma.

3.2.2 *Word Embeddings contextuales*

Una misma palabra puede tener varios significados (polisemia) o puede tener una interpretación semántica distinta dependiendo del contexto en el que se encuentre. Por ejemplo, la palabra *banco* de la oración *Me senté en un banco* tiene un significado totalmente distinto a la misma palabra en la oración *Fui al banco a realizar unas gestiones*. Las representaciones vectoriales estáticas, sin embargo, no tienen en cuenta esta particularidad muy común en el lenguaje. Debido a esta limitación de las representaciones vectoriales estáticas, aparecen aproximaciones que dan una solución y son capaces de representar un término en función del contexto en el que aparecen.

Uno de los pioneros en este tipo de aproximaciones es el modelo ELMo [110]. Se trata de un modelo del lenguaje neuronal basado en *Deep Learning* entrenado para la tarea de aprendizaje auto-supervisado de predecir la palabra t_k dado un contexto previo $(t_1, t_2, ..., t_{k-1})$. ELMo caracteriza un término según su contexto de forma bidireccional empleando como base de su arquitectura una red *Long-Short Term Memory* o LSTM [57] bidireccional. ELMo consiguió grandes resultados en varias tareas de PLN como son *Question Answering*, AO o implicación semántica, entre otras. Este modelo del lenguaje puede ser empleado como representación vectorial en cualquier tarea supervisada mediante la concatenación

del mismo con parámetros congelados con las capas necesarias para adaptarlo al problema supervisado de interés o mediante la aplicación de *fine-tuning* sobre el modelo.

Sin embargo, un modelo basado en LSTM puede presentar ciertas limitaciones como son por un lado los tiempos de cómputo, debido a la naturaleza secuencial de una LSTM y su difícil paralelización y, por otro lado, la dificultad para caracterizar dependencias entre palabras que están en posiciones alejadas entre sí en el segmento de texto. Dando solución a estas limitaciones, aparece en escena el *Transformer* [135], una de las contribuciones más revolucionarias e impactantes en el PLN. El *Transformer* es una arquitectura de *Deep Learning* cuya base es únicamente mecanismos de auto-atención, que son mecanismos basados en ponderar términos según su importancia semántica en el segmento de texto. La arquitectura *Transformer* es presentada mostrando resultados estado del arte en tareas como traducción automática y reduciendo de forma considerable los tiempos de cómputo de otras arquitecturas del estado del arte.

El impacto de los *Transformers* va más allá de los resultados expuestos en [135]. El *transformer* se convierte en una propuesta que sirve de base para los modelos del lenguaje más empleados y con mejores resultados en las distintas tareas de PLN. Uno de los modelos del lenguaje más populares basado en el *transformer* es BERT [35]. El entrenamiento de este modelo del lenguaje se realiza sobre un conjunto de gran tamaño de textos y es dirigido por dos tareas distintas: (1) un modelo del lenguaje enmascarado, basado en la predicción de palabras enmascaradas de forma aleatoria y (2) un problema de predicción de siguiente oración, donde el modelo debe responder si dos oraciones dadas como entrada son oraciones consecutivas en un documento. BERT es empleado en tareas supervisadas mediante la concatenación del modelo preentrenado con las capas necesarias para adaptarlo al problema deseado, realizando además un *fine-tuning* del modelo. Esta arquitectura supone resultados estado del arte en varias tareas del PLN como *Question Answering*. Además de BERT, existen muchos otros modelos del lenguaje, algunos como RoBERTa [84] basados en el propio BERT, u otros con un gran impacto debido a sus increíbles y realistas resultados como GPT-3 [22].

3.3 ANÁLISIS DE OPINIONES A NIVEL DE ASPECTO

El AO puede realizarse a distintos niveles de detalle y granularidad [81]: (1) a nivel de documento, asociando una polaridad del sentimiento a un documento completo, (2) a nivel de oración o segmento, asociando cada oración o segmento de texto a una polaridad y (3) a nivel de aspecto, asignando una polaridad a cada aspecto mencionado.

Entre los tres niveles de granularidad definidos, el análisis a nivel de aspecto (ABSA) es el de mayor detalle y mayor información, pero también el más complejo y desafiante. Por ejemplo, si una empresa fabricante de televisiones está interesada en conocer qué opinan sus clientes con respecto a un televisor concreto, es informativo para la empresa decir cuántas opiniones negativas y cuántas opiniones positivas posee este televisor. Sin embargo, decir qué opiniones se expresan con respecto a las distintas características o aspectos del televisor es mucho más informativo para el fabricante, pudiendo así este tomar decisiones basadas en conocimiento más concreto. La tarea de ABSA ha sido una tarea recurrente en el popular *workshop* de SemEval [113, 114, 142].

En esta sección, describiremos algunas propuestas para las dos tareas más esenciales en el ABSA. En la sección 3.3.1 enumeraremos propuestas para la detección de aspectos, es decir, para la detección de las características de una entidad mencionadas en el texto. Posteriormente, en la sección 3.3.2 se hablará de distintos enfoques de la literatura para la clasificación de polaridad del sentimiento expresado en texto.

3.3.1 *Detección de aspectos*

La detección de aspectos (DA) en opiniones es un paso clave en el análisis de opiniones a nivel de aspecto. Esta tarea consiste en identificar los aspectos o características sobre los que se opinan en una opinión. Esta tarea se puede afrontar a distintos niveles de granularidad, bien mediante técnicas para la obtención de aspectos a nivel de término, como las descritas en la sección 3.3.1.1 o bien mediante técnicas que se abstraen un poco más y detectan la mención de distintas categorías de aspectos como mencionamos en la sección 3.3.1.2. En la sección 3.3.1.3 explicamos *Attention-based Aspect Extraction*, modelo para la detección de aspectos de forma débilmente supervisada base para una de las contribuciones de esta tesis.

3.3.1.1 *Detección de aspectos a nivel de término*

Puede plantearse como el etiquetado de una secuencia de términos donde cada término obtendrá una etiqueta dependiendo de si forma parte o no de un aspecto. Existen diversos estudios en esta tarea del ABSA, la mayor parte de ellos con propuestas supervisadas. Por ejemplo, en [24] los autores proponen un modelo de Deep Learning basado en capas convolucionales para la extracción de aspectos a nivel de término. Más recientemente, en [146] los autores se enfrentan a la extracción de aspectos con un modelo Deep Learning basado en LSTMs bidireccionales y mecanismos de atención combinado con características POS (*Part of Speech Tagging*). También existen alternativas no supervisadas como [51], donde

los autores proponen un sistema de anotado automático para conjuntos de opiniones no anotados basado en reglas sintácticas y el empleo de estos conjuntos automáticamente anotados para entrenar un modelo Deep Learning de forma supervisada.

3.3.1.2 *Detección de categorías de aspecto*

Este enfoque para la DA detecta la categoría de aspectos a la que pertenecen los aspectos mencionados en una oración o segmento. Por ejemplo, en la oración *La pizza estaba buena pero el camarero fue antipático* un modelo de detección de categorías de aspectos podría detectar los aspectos *Comida* y *Servicio* u otras dependiendo del nivel de granularidad del análisis. Este enfoque también suele plantearse de forma supervisada, como por ejemplo en [72], donde los autores proponen un modelo Deep Learning basado en convoluciones para la clasificación de categorías de aspectos y de la polaridad del sentimiento de los mismos. De forma más reciente, en [23] se enfrentan al mismo problema usando una Hierarchical Graph Convolutional Network [73]. Sin embargo, también existen enfoques no supervisados, como en [55], descrito en detalle en la sección 3.3.1.3, donde se propone una arquitectura similar a la de un *autoencoder*, o enfoques semi-supervisados o débilmente supervisados como los propuestos en [4, 60, 78], propuestas en las que se parte de un conjunto de palabras semilla asociadas a cada categoría de aspecto que son utilizadas como "débil" supervisión en su proceso para la obtención de categorías de aspectos.

3.3.1.3 *Detección de aspectos de forma semi-supervisada: ABAE*

Attention-based Aspect Extraction (ABAE) se trata de un modelo neuronal no supervisado cuyo objetivo es encontrar los principales temas o aspectos mencionados en un conjunto de textos [55]. Debido a su naturaleza no supervisada, no requiere de un conjunto de datos previamente anotado, sino que solo requiere un conjunto de textos lo suficientemente grande y que pertenezcan al dominio sobre el que queremos trabajar. Además, este modelo puede ser guiado fácilmente partiendo de un conjunto de palabras semilla como detallaremos posteriormente.

Este modelo funciona como un autoencoder debido a que basa su funcionamiento en reducir la dimensión de la representación de una frase y volver a reconstruirla. A continuación, describimos cada una de las etapas del modelo neuronal también representadas en la figura 3:

1. Se obtienen las representaciones vectoriales o *word embeddings* e_i de tamaño D de cada palabra i de las n palabras de la oración de entrada. Para ello se emplea un diccionaro de *embeddings*.

2. ABAE obtiene una representación ponderada de la frase $z_s = \sum_{i=1}^{n} a_i e_i$, donde a_i es un peso de atención que se obtiene según las expresiones mostradas en la ecuación 7, donde M es una matriz de pesos aprendida por la red:

$$\begin{aligned} a_i &= \frac{exp(d_i)}{\sum_{j=1}^{n} exp(d_j)} \\ d_i &= e_i^{\intercal} \cdot M \cdot y_s \\ y_s &= \frac{1}{n} \sum_{i=1}^{n} e_i \end{aligned} \tag{7}$$

3. La representación z_s de dimensión $D \times 1$ es multiplicada por una matriz de pesos W de dimensión $K \times D$, donde K es un parámetro configurable que hace referencia al número de aspectos de interés. A este producto de matrices sumado al término independiente o *bias* b, se le aplica una función *softmax*, obteniendo de esta manera un vector p_t de dimensión $K \times 1$ que representa una distribución de probabilidad de pertenencia de la oración de entrada a cada uno de los K aspectos. De esta manera, hemos reducido la representación de la frase de tamaño D a tamaño K.
4. La representación reducida p_t es reconstruida mediante el producto matricial de p_t y la denominada matriz de aspectos T de dimensión $D \times K$. Esta matriz de aspectos es una matriz de pesos de la arquitectura de ABAE (son pesos entrenados por la red) y encapsula las representaciones vectoriales de los distintos aspectos de interés en nuestro dominio de estudio. Además, puede ser fácilmente inicializada con las representaciones vectoriales deseadas con el fin de guiar el entrenamiento del modelo.
5. Finalmente, obtenemos la reconstrucción r^s, la cual debe parecerse lo máximo posible a la representación z_s y lo menos posible a las y_s de otras oraciones aleatorias del conjunto de documentos. Para más información sobre la función de pérdida, consultar [55].

ABAE ofrece dos salidas de interés: (1) un conjunto de K *word embeddings* (matriz de aspectos) que representan las categorías de aspectos encontradas por el método y (2) una distribución de probabilidad p_t sobre las K categorías de aspectos, que clasifica una oración como una de las categorías de aspectos. En la propuesta original los autores realizan de forma preliminar un *clustering* con K-Means e inicializan la matriz de aspectos de ABAE con los centroides obtenidos.

Debido a la naturaleza no supervisada de ABAE, es frecuente que las categorías de aspectos encontradas no sean lo suficientemente precisas, proporcionando algunas con un nivel de granularidad muy bajo y otras que carecen de un significado concreto. Debido a esto, aparecen adaptaciones de ABAE como la propuesta en [4] y empleada en esta tesis en una de nuestras contribuciones, descrita en el capítulo 6, que consiste en la inicialización de la matriz de aspectos emplean-

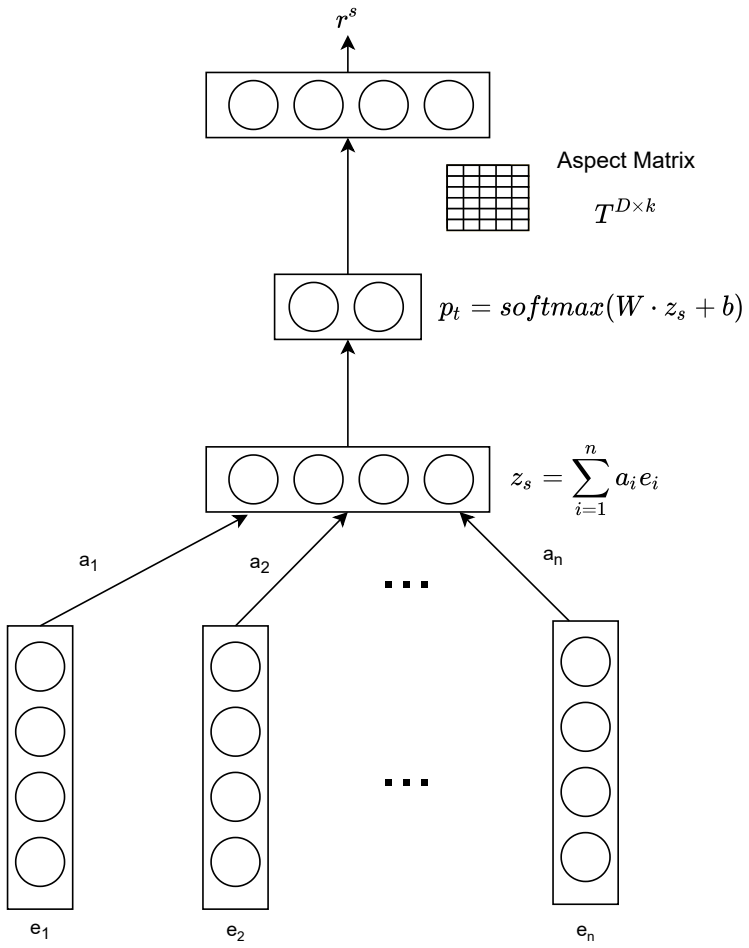

Figura 3: Esquema del modelo neuronal ABAE. Parte de la representación ponderada z_s de la oración, la cual sufre una reducción de dimensionalidad a una distribución de probabilidad p_t sobre los K aspectos. Finalmente, p_t es multiplicado por la matriz de aspectos T, dando a lugar a la reconstrucción r^s.

do un conjunto de palabras semilla por aspecto. Esta inicialización hace que el modelo pase a ser semi-supervisado, con una pequeña supervisión que guía al modelo para así converger a categorías de aspectos más significativas y homogéneas. Por ejemplo, un posible conjunto de palabras semilla para la categoría *Comida* podría ser {*Comida, cocinado, menú, delicioso, pizza*}.

3.3.2 *Clasificación de la opinión*

Conocer la polaridad expresada en una opinión es probablemente el elemento más importante a la hora de extraer conclusiones de la misma. Sin embargo, como ya hemos destacado, existen distintos niveles de análisis, los cuales están asociados también a un nivel de detalle y valor. En esta sección mostramos distintas propuestas para la clasificación de la polaridad a distintos niveles de granularidad. En la sección 3.3.2.1 mostramos algunas propuestas de la literatura para la clasificación de la opinión a nivel de oración y documento. En la sección 3.3.2.2 hablamos de algunas propuestas para la clasificación de la opinión a nivel de aspecto. Finalmente, en la sección 3.3.2.3 describimos algunos modelos de clasificación de la opinión pre-entrenados existentes e importantes para el desarrollo de una contribución de esta tesis descrita en el capítulo 4.

3.3.2.1 *Clasificación de la polaridad a nivel de oración y documento*

Estos niveles de granularidad son los de menor detalle y los que menos conclusiones invita a extraer. Por un lado, la clasificación de polaridad a nivel de oración consiste en dar respuesta a cuál es el sentimiento expresado en una oración o segmento de texto. Por otro lado, la clasificación de polaridad a nivel de documento puede interpretarse como la agregación de las distintas polaridades encontradas en el documento. Existen distintas propuestas para la clasificación de la polaridad a estos niveles de granularidad. Por ejemplo, en [83] los autores realizan una clasificación de la polaridad a nivel de documento en el dominio de correos electrónicos mediante el previo análisis de las polaridades expresadas con respecto a distintos temas o *topics*, proponiendo un *framekwork* que emplea LDA para la obtención de *topics*, embeddings de *topics* y una bi-LSTM para extraer características secuenciales en el texto. El AO a estos niveles de granularidad también se emplea en estudios basados en AO en *tweets*, como en [134] donde los autores investigan cómo afectan las opiniones financieras en Twitter al propio mercado financiero.

3.3.2.2 *Clasificación de la polaridad a nivel de aspecto*

Estas estrategias responden a la pregunta *¿Cuál es la polaridad de la opinión expresada con respecto a un determinado aspecto?*. Es decir, este nivel de granularidad no solo proporciona información sobre el sentimiento, sino que además nos proporciona información relacionada con sobre qué aspectos concretos se opina. Existen diversas propuestas de clasificación de la polaridad a nivel de aspecto. Por ejemplo, en [137] los autores proponen una aproximación en la que obtienen aspectos a nivel de término, categorías de aspectos y la polaridad asociada, descomponiendo el problema en varios subproblemas, donde al modelo (basado en BERT) se le introducen preguntas con el esquema *¿Se opina en la oración de forma*

positiva/negativa con respecto al aspecto x?, dando una respuesta binaria y además proporcionando un etiquetado de la secuencia para la DA a nivel de término. Otra aproximación original es la propuesta en [125] donde se plantean las tareas de detección de categorías de aspectos y de polaridad siguiendo un enfoque de tanto *Question Answering* como de inferencia del lenguaje empleando como base un modelo BERT.

3.3.2.3 *Modelos preentrenados para la clasificación de la polaridad*

Ante el creciente volumen de información subjetiva de la Web 2.0 y a la necesidad de la democratización de datos, técnicas de IA, etc., aparecen los Modelos para la Clasificación de la Opinión (MCO a partir de ahora), los cuales son métodos entrenados y especializados sobre determinados dominios, bien con fines comerciales o con fines educativos, para la clasificación de la polaridad en segmentos de texto o en documentos y ya listos para ser empleados por cualquier usuario. Existen MCO basados en técnicas de orientación semántica [129] o basadas en aprendizaje automático [107]:

- **Técnicas de orientación semántica (OS)**. Estos MCO realizan la clasificación de la polaridad mediante el empleo de reglas basadas en recursos lingüísticos como pueden ser listas de palabras, *lexicons*, reglas gramaticales, etc. Estas técnicas tienen como punto a favor una mayor facilidad para la interpretación. Sin embargo, son técnicas muy limitadas a la cobertura de los recursos lingüísticos y reglas empleados.
- **Técnicas de Aprendizaje Automático.** Emplean métodos de aprendizaje automático. Para el empleo de estos métodos, los textos son representados mediante un conjunto de características, como pueden ser características asociadas a la frecuencia de las palabras, características basadas en el número de palabras positivas o negativas o características más complejas como pueden ser los *Word Embeddings*.

En esta tesis, se realiza una propuesta para la combinación de varios MCO (ver capítulo 4) realizando un ponderado de la contribución de los mismos basados en la bondad de cada MCO en un dominio determinado. A continuación mostramos y describimos brevemente algunos MCO existentes empleados en esta tesis. Estos MCO se han escogido con el fin de tener heterogeneidad a la hora de realizar nuestra combinación de clasificadores. De esta manera, en nuestra propuesta participarán MCO de distinta naturaleza, que emplean distintas características de texto, y que están especializados en distintos dominios y géneros.

- **Azure.**[1] Modelo entrenado de la API de text analytics de la plataforma de Microsoft Azure. Se trata de un método de aprendizaje automático supervi-

1 https://docs.microsoft.com/en-us/legal/cognitive-services/language-service/transparency-note-sentiment-analysis

sado que funciona a nivel de documento. Su entrenamiento se realizó sobre un conjunto grande de opiniones sobre productos y servicios de microsoft. El género de los textos de entrenamiento son opiniones, el dominio productos y servicios y ofrece una salida $[0, 1] \in \mathbb{R}$.

- **Bing [59].** Es un método basado en OS que ofrece polaridad a nivel de documento. Para clasificar la opinión se basan primero en la construcción de listas de adjetivos positivos y negativos. Para ello parten de una serie de adjetivos semilla con orientación del sentimiento anotada y emplean el diccionario de sinónimos de WordNet [43] para buscar sinónimos o antónimos de estos adjetivos semilla y así incrementar la listas de adjetivos positivos y negativos. Para asignar una orientación del sentimiento al documento, se basan en la frecuencia de adjetivos positivos y negativos y tienen en cuenta la existencia de negaciones. Al ser un método basado en OS, no está especializado sobre ningún género concreto y el dominio podría definirse como general. La salida es un valor del conjunto {-1, 0, 1}.
- **CoreNLP [92, 123].** Se trata de un método de aprendizaje automático supervisado entrenado sobre un conjunto de opiniones en el dominio de *Cine*. El entrenamiento se realiza sobre un conjunto de oraciones representadas como estructura de árbol sintáctico y emplean una *Recursive Neural Tensor Network*, propuesta por los mismos autores en [123]. La salida es discreta en el conjunto de valores {0,1,2,3,4}, indicando 0 la polaridad más negativa y 4 la polaridad más positiva. Su nivel de granularidad en la clasificación es a nivel de oración y de documento.
- **MeaningCloud.**[2] Método basado en OS que trabaja a nivel de documento. Tiene en cuenta la categoría gramatical de las palabras y emplea un diccionario de opinión. Clasifica la intensidad de la polaridad en 5 niveles {0,1,2,3,4}. No se ha encontrado información relativa al dominio o género sobre el que está especializado ni grandes detalles de su implementación.
- **SentiStrength [128].** Método de aprendizaje automático supervisado basado en características construidas a partir de diccionarios de opinión y trabaja tanto a nivel de documento como a nivel de frase. Este método devuelve 5 niveles de intensidad de polaridad positiva (Pos) y 5 niveles de intensidad de polaridad negativa (Neg). Por lo tanto, la salida viene dada por las expresiones (Neg: $[-5, -1] \in \mathbb{Z}$; Pos: $[1, 5] \in \mathbb{Z}$). SentiStrength está especializado en el género de comentarios en redes sociales.
- **Syuzhet.**[3] Método basado en OS que funciona nivel de documento y de oración. Emplea un diccionario de opinión y la polaridad del documento viene determinada por la media de las polaridades de las oraciones presentes. Se trata de un método especializado en el género de texto de novelas y ofrece una salida discreta $\in$ {-1,0,1}.

2 https://learn.meaningcloud.com/developer/sentiment-analysis/2.1/doc
3 https://github.com/mjockers/syuzhet

- **Vader [61].** Se trata de un método de OS basado en el empleo de características léxicas y de reglas generales que tienen en cuenta aspectos gramaticales y sintácticos comunes a la hora de expresar sentimiento en texto. Vader está especializado en el género de *micro-blogging* y su salida es un número real en el intervalo [-1, 1], siendo -1 la polaridad más negativa y 1 la más positiva.

3.4 SÍNTESIS DE OPINIONES

Las técnicas de ABSA, como ya destacamos en la introducción de esta tesis en el capítulo 1, tienen como característica principal que aportan información puntual y dispersa sobre la mención de aspectos y su polaridad. Es decir, con estas técnicas podemos responder a las preguntas: *¿Qué aspectos son mencionados en esta oración?* o *¿cuál es la polaridad con respecto a este aspecto en este segmento de texto?*. Sin embargo, para que un fabricante o un cliente puedan obtener conclusiones sobre qué se opina sobre un determinado producto, es necesario contestar a otra pregunta: *¿Qué opina la gente de forma general sobre este producto?*.

La SO tiene como principal objetivo el empleo de técnicas para sintetizar o agregar las opiniones dadas por varios sujetos opinadores con respecto a los distintos aspectos o características de una entidad. De esta manera la SO, ya sea mediante técnicas extractivas, descritas en la sección 3.4.1, o técnicas abstractivas, como las mostradas en la sección 3.4.2, ofrece un conocimiento sintetizado y general sobre las opiniones con respecto a una entidad. Sin embargo, según la definición de SO dada por Bing Liu en [81], un resumen de opiniones es un tipo de resumen multi-documento (ampliamente investigado en el campo del PLN) pero que difiere en que estos resúmenes deben centrarse en entidades y aspectos concretos y además deben estar cuantificados. Esta definición casa con propuestas que van a un nivel más de abstracción con respecto a las técnicas extractivas o abstractivas, que son las técnicas para la síntesis de opiniones esquemáticas como las analizadas en la sección 3.4.3.

3.4.1 *Técnicas extractivas para la síntesis de opiniones*

Las técnicas extractivas para la síntesis de opiniones son aquellas que realizan la síntesis mediante la extracción de texto de las propias opiniones. Los segmentos de texto extraídos deben ser lo suficientemente relevantes como para que estos abarquen conocimiento general sobre las opiniones.

Existen diversas propuestas extractivas para la SO. Una de las primeras aproximaciones es la presentada en [12], donde los autores crean un clasificador de la opinión caracterizando las oraciones con características simples y seleccionan como oraciones para la síntesis aquellas con mayor probabilidad según el clasi-

ficador. En [77] los autores proponen un algoritmo para la síntesis de opiniones basado en la búsqueda de las oraciones más relevantes para cada aspecto o tema y que expresan opinión basándose principalmente en la frecuencia de términos. En [86] los autores realizan una serie de pasos para la obtención de resúmenes de opiniones relacionados con la DA y la clasificación de la polaridad de los segmentos en los que aparecen mediante el empleo auxiliar de la polaridad asociada a la valoración del usuario. Tras estas etapas, realizan una extracción de las oraciones más relevantes basadas en la frecuencia de las mismas.

Más recientemente, se han propuesto técnicas extractivas basadas en *Deep Learning*. En [4] los autores proponen un *framework* para la síntesis de opiniones en el que se realiza una detección de las categorías de aspectos de forma débilmente supervisada y una clasificación de la polaridad de las opiniones. Posteriormente, realizan un ránking de los segmentos de texto más relevantes basándose en la polaridad de los mismos y la probabilidad de aparición de las categorías de aspectos. Por último, realizan una selección de los segmentos para así evitar redundancia en el resumen obtenido. Otro ejemplo de propuesta extractiva para la síntesis de opiniones es la presentada en [62], donde los autores en lugar de realizar una DA realizan una tarea de reconocimiento de similitud de aspectos, una tarea consistente en predecir si una pareja de segmentos de texto comparten al menos la mención de un aspecto. Tras esto, realizan una selección de las oraciones más relevantes calculando la relevancia a partir de la polaridad de la opinión expresada y el número de aspectos mencionados, haciendo también un control de la redundancia entre los segmentos de texto seleccionados.

3.4.2 *Técnicas abstractivas para la síntesis de opiniones*

La síntesis de opiniones por medio de técnicas abstractivas consiste en la generación de texto de manera que el texto generado sea un resumen general de las opiniones expresadas con respecto a una entidad. Estos resúmenes, a diferencia de los obtenidos con técnicas extractivas, pueden contener términos u oraciones no presentes en el conjunto original de opiniones debido a que estas técnicas se basan en la generación de texto. Además, este tipo de técnicas reducen la redundancia de los resúmenes, uno de los principales problemas de las técnicas extractivas.

Existe una gran cantidad de propuestas de estrategias abstractivas para el resumen de texto en contextos no subjetivos (véase [120, 122, 82]). Sin embargo, estas propuestas son supervisadas, por lo que requieren como conjunto de datos pares de {texto,resumen} o {conjuntos de texto, resumen} para resumen de documentos o multi-documentos, respectivamente. En síntesis de opiniones obtener este tipo de conjuntos de datos es realmente costoso, pues supondría el emparejar múltiples documentos (múltiples opiniones por entidad) con un resumen. Existen

algunas propuestas totalmente supervisadas como la presentada en [18], donde los autores presentan primero el conjunto de opiniones con esquema {conjuntos de texto,resumen} más grande existente y posteriormente una técnica para reducir este conjunto mediante la selección de las opiniones más informativas, haciendo que el entrenamiento de un modelo sea computacionalmente admisible.

Debido a la dificultad mencionada relacionada con la obtención de conjuntos de datos para enfoques supervisados, la mayoría de propuestas abstractivas para la síntesis de opiniones son propuestas no supervisadas o débilmente supervisadas. En [3] los autores realizan una propuesta en la que emplean técnicas supervisadas pero con un enfoque no supervisado. En su propuesta, toman una opinión cualquiera la cual consideran como un pseudo-resumen y_i y a partir de este resumen generan diversas versiones ruidosas del mismo, que formarán el conjunto de opiniones X_i de manera que obtenemos un par documentos-resumen $\{X_i,y_i\}$ que, ahora sí, puede ser usado en técnicas supervisadas. En [63] los autores presentan una propuesta similar a la previamente descrita pero en la que emplean *meta-data* de las opiniones, como por ejemplo las imágenes adjuntadas en las opiniones, para generar pseudo-resúmenes.

3.4.3 *Síntesis de opiniones mediante la generación de resúmenes estructurados*

Las estrategias previamente descritas, bien extractivas o bien abstractivas, tienen en común una característica: ambas presentan resúmenes mediante texto. Sin embargo, existen propuestas que van a un nivel mayor de abstracción, que son las propuestas para la generación de resúmenes estructurados. Estas propuestas, además, tienen una mayor relación con la principal hipótesis de esta tesis: es posible obtener resúmenes de opiniones de manera que estos se presenten de forma esquemática o estructurada, haciendo esto que los resúmenes sean más legibles de cara al usuario, más generales y, además, que sean explicables.

Existen diversas propuestas de síntesis de opiniones de forma esquemática, las cuales podemos diferenciar en tres categorías:

- Métodos visuales. Son métodos que se basan en la visualización de determinadas estadísticas tras la DA. En [116] los autores emplean una modificación del algoritmo LDA para obtener los aspectos mencionados y muestran algunas estadísticas relacionadas con las categorías de aspectos más mencionadas así como las polaridades expresadas. En [138] los autores representan los resúmenes de forma visual por medio de un grafo interactivo, dando la opción al usuario de buscar información de su interés.
- Métodos basados en grafos de conocimiento. Son resúmenes obtenidos mediante la representación como gráficos de conocimiento [140], que consolidan

la información semántica de un conjunto de documentos (opiniones en nuestro caso). Por ejemplo en [25] los autores combinan métodos de ABSA, la identificación de relaciones causales e implicación semántica para construir un grafo de conocimiento de opiniones.

- Métodos basados en reglas. Son métodos que muestran los resúmenes en forma de reglas, siendo estos resúmenes fácilmente interpretables y legibles. Por ejemplo, en [132] los autores emplean técnicas de SD para analizar las opiniones negativas en el dominio de opiniones sobre monumentos. En una de nuestra contribuciones, descrita en el capítulo 6, a diferencia de esta propuesta, planteamos una metodología débilmente supervisada en la que la mayor granularidad de los aspectos obtenidos dan lugar a resúmenes más generales.

4

ADAPTACIÓN AL DOMINIO MEDIANTE COMBINACIÓN DE CLASIFICADORES

En este capítulo, presentamos la primera contribución de esta tesis, relacionada con el subobjetivo de esta tesis relacionado con dar una solución al reto de la adaptación al dominio en el campo del Análisis de Opiniones. Esta contribución se trata de una metodología para la clasificación de la opinión en texto cuyo fin es optimizar la clasificación de la opinión por parte de Modelos de Clasificación de Opiniones, que son recursos preentrenados para la clasificación de la opinión en texto. La metodología desarrollada es un modelo de combinación de Modelos de Clasificación de Opiniones que emplea algoritmos evolutivos para encontrar una distribución de pesos que mejore el rendimiento de un modelo de forma individual. Esta distribución de pesos hace que este modelo de combinación consiga una adaptación al dominio del texto de entrada.

4.1 INTRODUCCIÓN

La tarea de clasificación de la polaridad del sentimiento expresado en una opinión puede definirse como un problema de clasificación binaria (clases Positiva y Negativa) o un problema multi-clase con diferentes intensidades de polaridad. Esta tarea es una de las tareas clave en el AO y como tal surgen herramientas o recursos para la clasificación de polaridad preentrenados, los cuales llamamos Modelos de Clasificación de Opiniones (MCO).

Los MCO son métodos, bien comerciales o bien de interés investigador, cuyo fin es proporcionar al usuario una herramienta para la obtención de la polaridad en un texto. Estos MCO son recursos que han sido entrenados o especializados en un determinado dominio (financiero [7], hostelero [32], salud [112]...) y en un determinado género de texto (noticias [68], *micro-blogging* [49], opiniones [81]...). Además, el rendimiento de un MCO está relacionado con la aproximación de aprendizaje seguida, pudiendo esta ser desde métodos supervisados hasta métodos no supervisados (basados en reglas lingüísticas, en diccionarios, etc.).

La gran demanda de recursos para el usuario para el análisis de opiniones provoca un crecimiento del desarrollo de nuevos MCO. Estos MCO son de muy distintas naturalezas: distintos métodos de aprendizaje, distinta caracterización del texto, distintos géneros y dominios sobre los que se han especializado, etc. Esto supone que cada uno de ellos tenga sus ventajas y sus inconvenientes con respecto a su rendimiento en distintos dominios de texto de evaluación. Estos elementos hacen que surja el problema de adaptación al dominio [81], el cual se presenta cuando un método es evaluado en un dominio distinto al dominio en el que el método está especializado, y trata de aportar una mayor capacidad de generalización al método.

La combinación de distintos modelos de clasificación o regresión aporta puntos de vista heterogéneos y una mayor robustez a la hora de la predicción. Sin embargo, a la hora de realizar la agregación de las predicciones de los distintos modelos base, esta no se realiza teniendo en cuenta el comportamiento de cada modelo base en el dominio de estudio. En este capítulo presentamos una contribución motivada por la hipótesis de que una combinación de MCOs, donde cada MCO tiene una contribución en la respuesta final dependiendo del comportamiento de cada MCO en el dominio de interés, ofrecerá mejores resultados en la tarea de clasificación de la opinión que el empleo de estos MCOs de forma individual y que otros operadores de agregación de la literatura. Esta contribución se trata de una metodología cuyo objetivo es la de combinar las respuestas de varios MCOs. Esta combinación se realiza mediante el empleo de un AE que optimiza la distribución de pesos a cada MCO dependiendo de la naturaleza del texto de evaluación.

Este capítulo se estructura de la siguiente manera: en la sección 4.2 hacemos una definición formal de nuestra contribución, en la sección 4.3 detallamos el marco experimental empleado para contrastar nuestra contribución, en la sección 4.4 mostramos y analizamos los resultados de nuestra experimentación y finalmente en la sección 4.5 presentamos las conclusiones obtenidas a partir de la experimentación.

4.2 MÉTODO DE COMBINACIÓN EVOLUTIVO

La capacidad predictiva de un MCO preentrenado tiene una especial dependencia de la aproximación de aprendizaje empleada para su entrenamiento, así como del dominio y género del texto empleado para el aprendizaje. De esta manera, el empleo de un MCO de forma individual puede ofrecer resultados limitados dependiendo del dominio sobre el que se emplee.

Los métodos de combinación de modelos predictivos son métodos que combinan modelos base de distinta naturaleza o que son entrenados en distintas condiciones. Estos métodos han demostrado que en muchas tareas mejoran el emplear estos modelos base de forma individual debido a la unión de varios espacios de búsqueda. Sin embargo, si se realiza una agregación uniforme de las respuestas de los modelos base, podríamos estar deteriorando el rendimiento debido a que asignaríamos el mismo peso de contribución a modelos cuyo rendimiento es muy bueno y a modelos cuyo rendimiento no es tan bueno. De esta manera, surge el desafío de buscar el nivel de contribución de cada modelo base con el fin de optimizar el rendimiento de la agregación final.

El resultado obtenido de la predicción de un método de combinación depende de la función empleada para la combinación de las salidas de cada uno de los estimadores base. En nuestro caso, cada MCO devuelve un valor de polaridad $p_i \in P$ para cada opinión $o_k \in O$, donde P es el conjunto de posibles valores de polaridad y O el conjunto de opiniones. De esta manera, para cada o_k obtenemos un vector de estimaciones de polaridad $p = (p_1, ..., p_s)$ formado por la polaridad dada por cada uno de los estimadores del conjunto de estimadores S. A partir del vector p, se puede definir formalmente un método de combinación como aparece en Ecuación 8.

$$\begin{aligned} f_{ens} &: [0,1]^{|S|} \rightarrow [0,1] \\ (p_1, \ldots, p_s) &\mapsto f_{ens}(p_1, \ldots, p_s) \end{aligned} \tag{8}$$

En esta sección presentamos nuestra contribución E^{2}SAM, una metodología cuyo resultado es una combinación de MCOs de distinta naturaleza y especializados en distintos dominios y géneros. E^{2}SAM se presenta como una metodología

totalmente independiente de los MCOs que el usuario quiera emplear para realizar la combinación. E^{2}SAM realizará una agregación de las respuestas de los MCOs donde cada MCO tendrá una contribución a la respuesta proporcional a la bondad de ese MCO en el dominio de interés. En la ecuación 9 redefinimos la función general f_{ens} de la ecuación 8 como f_{e^2sam}, donde los pesos w_i son aprendidos por un AE.

$$f_{e^2sam}(p_1, \ldots, p_s) = \sum_{i=1}^{|S|} w_i p_i, \tag{9}$$

En la figura 4 se muestra un esquema general del flujo de E^{2}SAM. Este flujo parte de un conjunto de MCO base, descritos en la sección 3.3.2.3, y posteriormente E^{2}SAM realiza una búsqueda del mejor reparto de pesos según el dominio del texto de evaluación. Esta búsqueda se realiza por medio de un AE que optimizará una función objetivo seleccionada por el usuario, proporcionando de esta manera también cierta flexibilidad con respecto a qué aspectos del rendimiento se quieren optimizar. En la sección 4.3 se dan más detalles sobre la experimentación realizada para seleccionar qué AE será usado para implementar E^{2}SAM.

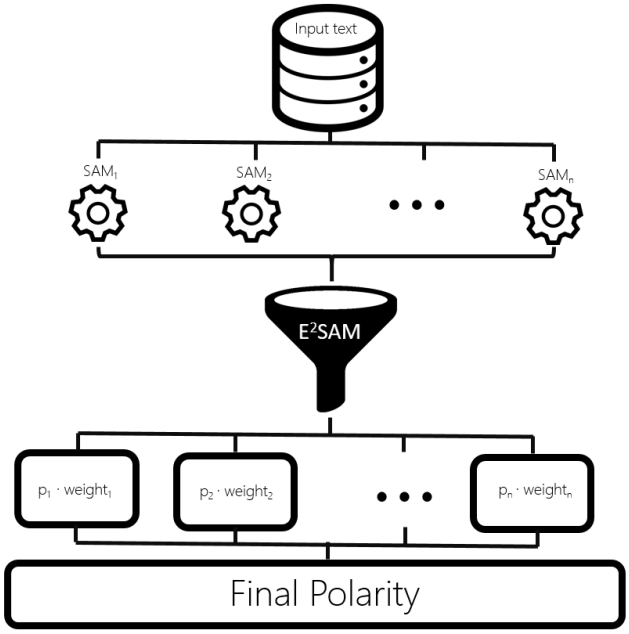

Figura 4: Flujo del proceso de E^{2}SAM, compuesta por un conjunto de MCO y un método evolutivo que calcula la distribución de pesos más óptima. Fuente: [88].

4.3 MARCO EXPERIMENTAL

En esta sección haremos una descripción de todo el marco experimental empleado para la evaluación de las distintas aproximaciones. En la sección 4.3.1 describimos los conjuntos de datos empleados. En la sección 4.3.2 mostramos la configuración de los tres AEs usados para la implementación final de E^2SAM. En la sección 4.3.3 detallamos dos operadores de combinación de MCOs de la literatura empleados como modelos de combinación base para comparar con E^2SAM. En la sección 4.3.4 se describen los distintos aspectos que se han tenido en cuenta a la hora de realizar la evaluación de las aproximaciones. En la sección 4.3.5 describimos los recursos empleados para usar cada MCO base.

4.3.1 *Conjuntos de datos*

Los conjuntos de datos empleados para la evaluación de nuestra propuesta son un subconjunto de los empleados en [117]. Este subconjunto se ha tomado teniendo en cuenta que queremos mostrar cómo la capacidad predictiva de un MCO depende de forma significativa del género y dominio del texto de entrada. Por lo tanto, emplearemos 13 conjuntos de datos de distinto género y dominio. Esto significa que algunos de los dominios y géneros de los datasets empleados para la evaluación coinciden con los empleados para el modelado de algunos de los MCO base, es decir, se evalúan tanto en el dominio en el que han sido especializados (*In-Domain*) como fuera del dominio de especialización (*Off-Domain*). Como ejemplo, hay datasets del dominio *Opiniones de películas* (`pang_movie`, `vader_movie`) o texto de género *micro-blogging* (`debate`, `vader_twitter`, `english_dailabor`, `tweet_semevaltest`, `sentistrength_twitter`).

También se han empleado conjuntos de datos cuyos dominios y géneros no se han empleado en el entrenamiento de ninguno de los MCO. Estos son concretamente datasets que son comentarios de foros científicos y tecnológicos (`sentistrength_digg`), comentarios en distintos sitios webs (`sentistrength_youtube`, `sentistrength_bbc`, `sentistrength_myspace` and `vader_nyt`) y opiniones sobre productos (`vader_amazon`).

En la tabla 1 se muestran algunas de las características de los conjuntos de datos empleados. Concretamente, se muestran el tamaño del conjunto (Tamaño), el número de opiniones Positivas (Pos.), Negativas (Neg.) y Neutras (Neu.), el número medio de frases (Media frases), el número medio de palabras (Media palabras), el género del texto (Género) y el dominio del texto (Dominio).

Como podemos observar en la tabla 1, los conjuntos seleccionados para el estudio son heterogéneos. `Pang_movie` y `vader_movie` contienen un 50 % más de comentarios que el resto. `Sentistrength_bbc`, `sentistrength_digg` y `sentistren-`

gth_myspace son los datasets que menos comentarios contienen. Algunos de los datasets están desbalanceados, es decir, tienen más comentarios de una polaridad que de otras. Esto ocurre, por ejemplo, en los datasets debate y sentistrength_myspace. También está la particularidad de pang_movie, el cual no contiene ninguna opinión neutral. Por lo general, el número de frases media por comentario son similares, a excepción de los datasets sentistrength_bbc y sentistrength_digg.

Conjunto de datos	Tamaño	Pos.	Neg.	Neu.	Media frases	Media palabras	Género	Dominio
debate	3238	730	1249	1259	1.86	14.86	Micro-Blogging	Debate político
english_dailabor	3771	739	488	2536	1.54	14.32	Micro-Blogging	General
pang_movie	10662	5331	5331	-	1.15	18.99	Opiniones	Cine
sentistrength_bbc	1000	99	653	248	3.9	64.39	Comentarios Foro	Noticias
sentistrength_digg	1077	210	572	295	2.50	33.97	Comentarios Foro	Noticias Tecnología
sentistrength_myspace	1041	702	132	207	2.22	21.12	Posts Redes Sociales	General
sentistrength_twitter	4242	1340	949	1953	1.77	15.81	Micro-Blogging	General
sentistrength_youtube	3407	1665	767	975	1.78	17.68	Forum Comments	General
tweet_semevaltest	6087	2223	837	3027	1.86	20.05	Micro-Blogging	General
vader_amazon	3708	2128	1482	98	1.03	16.59	Opiniones	Productos
vader_movie	10605	5242	5326	37	1.12	19.33	Opiniones	Cine
vader_nyt	5190	2204	2742	274	1.01	17.76	Comentarios Foro	Noticias
vader_twitter	4200	2897	1299	4	1.87	14.10	Micro-Blogging	General

Tabla 1: Resumen de los conjuntos de datos empleados en nuestro estudio.

4.3.2 *Algoritmos evolutivos para la implementación de E^{2}SAM*

Los AEs son algoritmos de optimización que se basan en los procesos de evolución natural. Son métodos que parten de una población (conjunto de posibles soluciones). Los individuos de esta población sufren mutaciones y se reproducen (se combinan soluciones), dando lugar a nuevos individuos o soluciones. Finalmente, el algoritmo selecciona qué individuos permanecen en la población basándose en cómo de fuertes son, es decir, el algoritmo selecciona las soluciones que mejores valores ofrecen a una función de coste.

En esta sección mostramos los distintos AEs empleados en la experimentación para seleccionar un AE para la implementación de E^{2}SAM. En nuestra experimentación se emplean tres AEs distintos basados uno de ellos en AMs y dos en

estrategias de ED. Siendo n el tamaño de la población, a continuación se detallan las particularidades de estos AEs en nuestra experimentación:

ALGORITMO MEMÉTICO. Son algoritmos que combinan la capacidad de exploración de un algoritmo genético y la de explotación de una búsqueda local. En la sección 2.3.2 se puede ver una descripción más detallada de este tipo de algoritmos. En nuestra implementación del AM se han tenido en cuenta las siguientes particularidades:

- Para la fase de selección se emplean n torneos binarios, es decir, n comparaciones de 2 individuos aleatorios donde permanece el que ofrece un mejor valor de la función objetivo.
- En la fase de reproducción se generará un máximo de n hijos, ocurriendo con una probabilidad de 0.7. Se emplea como operador de cruce un *Blend Crossover Operator* (BLX-α) [41]. Este operador genera cromosomas hijos siguiendo el siguiente proceso:
  1. Se seleccionan dos padres c^1 y c^2 del conjunto de individuos de forma aleatoria.
  2. Para cada elemento c_i^n del cromosoma hijo c^n, se elige un valor aleatorio uniforme del intervalo $[C_i^1; C_i^2]$ siguiendo la siguiente distribución:

$$\begin{aligned} C_i^1 &= min(c_i^1, c_i^2) - \alpha d_i \\ C_i^2 &= max(c_i^1, c_i^2) + \alpha d_i \\ d_i &= |c_i^1 - c_i^2| \end{aligned} \tag{10}$$

  donde c_i^1 y c_i^2 son los elementos i de los padres c^1 y c^2, respectivamente, y α es un número positivo para expandir proporcionalmente el intervalo del dominio de los parámetros, d_i. En nuestra implementación, empleamos el valor por defecto de 0.1 para α [41].
- En la etapa de mutación cada gen tendrá una probabilidad de 0.001 de ser mutado, consistiendo esta mutación en sumar al gen mutado un valor aleatorio generado según una distribución Normal con media 0 y desviación típica 0.3.
- El reemplazamiento de la población G por la población G+1 se realiza sustituyendo la población anterior por la nueva en su totalidad. Se realiza un reemplazamiento elitista, lo que quiere decir que el peor cromosoma de la nueva población será sustituido por el mejor de la población anterior.
- La búsqueda local empleada será el algoritmo *Hill Climbing* [121] y se realizará cada 10 generaciones, explorando soluciones vecinas a algunas de las existentes en la población. El algoritmo de búsqueda local explorará un máximo de vecinos determinado por 5 veces el tamaño del cromosoma.
- El valor de n (tamaño de la población) será 30 y el criterio de parada será 100.000 evaluaciones de la función objetivo.

EVOLUCIÓN DIFERENCIAL. Son estrategias evolutivas que enfatizan especialmente la mutación. En nuestra experimentación implementamos los contrastados jSO y L-Shade, algoritmos descritos en la sección 2.3.4. Para estos algoritmos, seleccionamos los siguientes parámetros:

- Tamaño de la población máximo e inicial n_{max} = 50 y tamaño mínimo de la población n_{min} = 4.
- Ratio p_{best} = 0.11 en L-Shade y 0.25 en jSO, parámetros por defecto de los algoritmos.
- Tamaño de memoria = 5 para ambos algoritmos.
- Para jSO, los valores p_{min} y p_{max} tienen un valor de 0.5 y 1, respectivamente.

4.3.3 *Otras propuestas de métodos de combinación de modelos predictivos*

Con el fin de validar el rendimiento de E^{2}SAM, comparamos los resultados obtenidos con otros dos operadores disponibles de la literatura también empleados para la combinación de MCOs. Estos modelos u operadores, ambos propuestos en [131], asignan un peso determinado a cada MCO dependiendo de la salida de los mismos. Sin embargo, estos pesos no tienen en cuenta cómo cada MCO se comporta en cada dominio. A continuación describimos los dos operadores usados en esta tesis:

- *Average Based Model* (AVG). Es un modelo de agregación en el que todos los MCO contribuyen con el mismo peso al valor final de la polaridad. f_{ens} se reformula como f_{avg} y se describe en la ecuación 11.

$$\begin{aligned} f_{avg}(p_1,\ldots,p_s) &= \sum_{i=1}^{|S|} w_i p_i \\ w_i &= \frac{1}{|S|},\ \forall i \in \{1,\ldots,s\} \end{aligned} \tag{11}$$

- *Neutral Penalty Based Model* (NEUTY). Da menos importancia a aquellos MCO que devuelven una polaridad más neutra. De esta forma, f_{ens} se reformula como f_{neuty} y se describe en la ecuación 12.

$$\begin{aligned} f_{neuty}(p_1,\ldots,p_s) &= \sum_{i=1}^{|S|} w_i p_i \\ w_i &= \frac{|p_i - 0{,}5|}{\sum_{j=1}^{|S|} |p_j - 0{,}5|},\ \forall i \in \{1,\ldots,s\} \end{aligned} \tag{12}$$

4.3.4 *Entrenamiento y evaluación*

La evaluación realizada en nuestro estudio consta de 2 fases: (1) Una evaluación de los MCO en cada conjunto de datos, así como de los modelos de combinación base descritos en la sección 4.3.3 y (2) una evaluación de los tres AEs descritos en la sección 4.2 con el fin de encontrar el mejor método evolutivo para E^{2}SAM.

El método empleado para la evaluación es una validación *Hold-out*, habiéndose realizado una partición en cada conjunto de datos de un 80 % para *train* y un 20 % para *test*. Como los MCO son métodos ya entrenados, la partición de *train* no se empleará, sino que se evaluará exclusivamente sobre el 20 % de test. En la figura 5 se muestra un esquema gráfico del flujo de evaluación empleado.

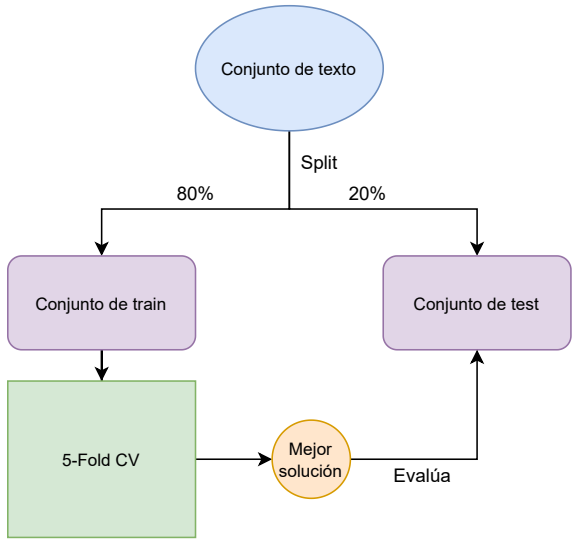

Figura 5: Flujo de experimentación empleado en el estudio. Se divide el conjunto en un conjunto de entrenamiento y otro de evaluación. Sobre el conjunto de entrenamiento, se realiza una 5-Fold *Cross Validation*, de donde se obtiene la mejor solución. Esta mejor solución se evaluará sobre el 20 % de test restante.

E^{2}SAM, como modelo predictivo, debe ser capaz de tener una capacidad de generalización aceptable. Sin embargo, un AE es un método de optimización cuyos hiperparámetros no son configurables con el fin de buscar esta capacidad de generalización. Esto hace que debamos realizar un modelado de E^{2}SAM enfocado en buscar esa capacidad de generalización. Por ello, nuestro modelado se basa en una 5-Fold *Cross Validation* sobre el 80 % de datos de *Train* donde cada AE tendrá un vector solución para la función f_{e^2sam} para cada *Fold* de entrenamiento. A

partir de estos vectores solución, consideramos como mejor solución para un AE a aquella que obtiene el mejor resultado de entre todos los *Folds*, es decir, aquella que ha generalizado mejor en este proceso de modelado. Esta mejor solución, es la empleada para evaluar sobre el 20 % de *test*. Este proceso, debido a la aleatoriedad implícita en los AEs, se repetirá 30 veces, mostrando en la sección 4.4 los resultados promedio de estas 30 ejecuciones.

Como métrica de evaluación hemos empleado el F_1, definido como la media armónica entre las métricas Precision y Recall (ver ecuación 13).

$$\begin{aligned} F_1 &= \frac{2 * Precision * Recall}{Precision + Recall} \\ Precision &= \frac{tp}{tp + fp} \qquad\qquad Recall = \frac{tp}{tp + fn} \end{aligned} \tag{13}$$

donde tp es el número de instancias donde el modelo clasifica como clase positiva y acierta, fn es el número de instancias donde el modelo clasifica como clase negativa de forma errónea y fp el número de instancias donde el modelo clasifica como clase positiva de forma errónea.

En nuestra evaluación, definimos la tarea de clasificación de la polaridad como un problema de clasificación de tres clases clases, donde cada texto es clasificado como Positivo, Negativo o Neutro. Por lo tanto, adaptamos la métrica F_1 al problema de tres clases, calculando el Macro-F_1. Esta métrica consiste en calcular el F_1 para cada una de las clases y aplicar una media aritmética. Nos referiremos al Macro-F_1 directamente como F_1.

4.3.5 *Modelos de Clasificación de Opiniones*

Como ya se ha mencionado, consideramos nuestra tarea como una tarea de clasificación de tres clases (positivo, negativo y neutro). Formalmente, nuestro sistema devolverá un valor de polaridad $p \in [0, 1]$. Con el fin de asignar la etiqueta final, la asignamos según tres umbrales, donde la polaridad es Positiva si $p \in (0{,}66, 1]$, Neutra si $p \in (0{,}33, 0{,}66]$ y Negativa si $p \in [0, 0{,}33]$.

Con el fin de homogenizar la salida de los MCO, se define la polaridad p_i como un valor real tal que $p_i \in [0, 1]$. Por lo tanto, en cada MCO realizamos el posprocesamiento a continuación descrito.

- **Azure.** Se empleó el paquete `mMCOtexta4r`[1] y la salida ya está definida en el intervalo $[0, 1]$.

1 https://cran.r-project.org/web/packages/mscstexta4r/README.html

- **CoreNLP.** Empleamos el paquete CoreNLP[2] de R. La salida de CoreNLP son valores discretos de 5 posibles valores de intensidad de polaridad. Por lo tanto, la clase *Muy Positiva* tendrá valor 1, *Positiva* tendrá valor 0.75, *Neutra* tendrá valor 0.5, *Negativa* tendrá valor 0.25 y *Muy negativa* como 0. Este método ofrece la polaridad a nivel de frase, por lo que la opinión de un comentario será la media de las polaridades de las frases.
- **MeaningCloud.** También proporciona una salida discreta de cinco intensidades de polaridad, por lo que seguimos la misma estrategia que con CoreNLP.
- **SentiStrength**[3]. Devuelve los valores tanto de positividad como de negatividad de un texto de entrada. Para obtener la polaridad final, hacemos una agregación de ambos valores de polaridad. Consideramos polaridad postiva si el valor agregado de la polaridad está en (0.66,1] y negativa si está en el intervalo [0,0.33].
- **Bing y Syuzhet.** Empleamos el paquete syuzhet[4] de R. La polaridad es ofrecida a nivel de frase, por lo que se realiza una agregación mediante una media. Se asigna la polaridad siguiendo el mismo criterio que con Sentistrength.
- **Vader.** Empleamos el script de python[5] presentado por [117]. Hacemos una normalización min-max a la salida.

4.4 RESULTADOS Y ANÁLISIS

En esta sección mostramos y analizamos los resultados de la experimentación realizada para presentar E^{2}SAM. En la sección 4.4.1 justificamos el AE empleado para la implementación de E^{2}SAM, comparando los tres AEs definidos en la sección 4.2. En la sección 4.4.2 presentamos los resultados de cada MCO de forma individual en cada uno de los conjuntos de texto así como de los operadores de combinación de modelos de aprendizaje base empleados y presentados previamente en la sección 4.3.3. Finalmente, en la sección 4.4.3 realizamos un estudio de cómo el empleo de E^{2}SAM supone diferencias significativas en la distribución de errores con respecto a los mejores MCO individuales y cómo E^{2}SAM realiza la ponderación de los distintos MCOs dependiendo de los dominios de los textos.

4.4.1 *Selección de algoritmo evolutivo para la implementación de E^{2}SAM*

Para seleccionar un AE para el desarrollo de la función f_{e^2sam}, evaluamos y comparamos el rendimiento de L-Shade, jSO y un AM, descritos todos ellos en la sección 4.3.2. Como ya se ha mencionado en la sección 4.3.4, (1) empleamos una

2 https://cran.r-project.org/web/packages/coreNLP/index.html
3 http://sentistrength.wlv.ac.uk/
4 https://cran.r-project.org/web/packages/syuzhet/vignettes/syuzhet-vignette.html
5 https://bitbucket.org/matheusaraujo/sentimental-analysis-methods

	F_1 - AE		
Dataset	**AM**	**L-Shade**	**jSO**
debate	0.443	**0.449**	0.448
english_dailabor	**0.725**	0.724	0.724
pang_movie	**0.637**	**0.637**	**0.637**
sentistrength_bbc	0.489	**0.496**	**0.496**
sentistrength_digg	0.552	0.554	**0.555**
sentistrength_myspace	**0.621**	0.610	0.608
sentistrength_twitter	**0.643**	0.637	0.640
sentistrength_youtube	0.627	**0.629**	**0.629**
tweet_semevaltest	0.643	**0.647**	0.645
vader_amazon	0.592	**0.597**	**0.597**
vader_movie	**0.565**	0.556	0.562
vader_nyt	0.543	**0.546**	0.545
vader_twitter	**0.754**	0.753	0.753
Media	**0.603**	**0.603**	**0.603**

Tabla 2: F_1 medio de cada uno de los AEs empleados en los 13 conjuntos de datos de texto. Se observan unos resultados muy similares en todos los datasets, viéndose este hecho reflejado en los resultados medios mostrados.

5-Fold Cross Validation sobre el conjunto de *Train*, (2) seleccionamos los pesos obtenidos en el mejor *Fold* y (3) evaluamos con estos pesos en el conjunto de test. Debido a la naturaleza estocástica de los AEs, se han hecho 30 repeticiones de los experimentos. Por lo tanto, los resultados serán la media de las 30 iteraciones [50]. En la tabla 2 mostramos el F_1 para cada AE.

Como muestra el F_1 medio de cada AE, los resultados son muy similares. Para seleccionar qué método evolutivo emplear como implementación de E^2SAM, se empleó un test Wilcoxon [139] para estudiar si existe alguna diferencia significativa entre cada par de los tres AEs. Este test estadístico se emplea para la comprobación de si existen diferencias significativas en la distribución de dos muestras pareadas. Este test es comúnmente empleado a la hora de comparar el rendimiento de métodos y algoritmos. Tras realizar el test, se concluye que no existe un método evolutivo significativamente mejor que el resto con un $p-$value de 0.05. Por lo tanto, para elegir la mejor aproximación evolutiva, tuvimos en cuenta los valores de Ránking positivo (R+) y Ránking negativo (R-) del propio

	R+	R-
L-Shade vs. MA	43.5	34.5
jSO vs. L-Shade	15	13
jSO vs. MA	49.5	28.5

Tabla 3: Valores de rankings R+ y R- para cada comparación de parejas de los AEs seleccionados. Se observa que jSO obtiene ligeramente un mayor valor R que L-Shade y MA.

test de Wilcoxon. En la tabla 3 se muestran los valores R+ y R- de cada pareja de AEs. Aunque no sean diferencias significativas según el test de Wilcoxon, jSO se muestra como el método con mayor valor R y confirma que los algoritmos de ED son métodos mejores que los AMs con respecto a optimización de parámetros continuos. También podría significar que los cambios que incorpora con respecto a L-Shade para la gestión de distintos parámetros de configuración influyen positivamente.

4.4.2 *Comparación de E^2SAM con los modelos base*

RESULTADOS DE LOS MCO INDIVIDUALMENTE. En la tabla 4 se muestran los resultados obtenidos por cada MCO en cada uno de los conjuntos de texto. En la tabla se aprecia que los MCO entrenados sobre conjuntos de naturaleza similar a la de los conjuntos donde se evalúa, reflejan este hecho en un mejor resultado que otros MCO. Por ejemplo, `CoreNLP` obtiene los mejores resultados en los dos conjuntos de datos de opiniones sobre películas (`pang_movie` y `vader_movie`). Por otro lado, `Sentistrength` obtiene buenos resultados especialmente en los conjuntos de texto de `sentistrength`. Si vemos los resultados de estos MCO en otros conjuntos, se ve que estos resultados empeoran, llevándonos este hecho a concluir que el rendimiento de un MCO depende en gran medida del género y dominio donde estos se han entrenado.

RESULTADOS DE LOS MODELOS DE COMBINACIÓN. En la tabla 5 mostramos los resultados de los dos modelos base de combinación (AVG, NEUTY) y E^2SAM. También se incluye el F_1 del mejor MCO para cada dataset, así como el nombre de este MCO. Lo primero que destacamos, es que el operador AVG solo mejora al mejor MCO en cinco conjuntos de datos mientras que NEUTY solo lo hace en tres, lo que es un resultado sorprendente debido a que un método de combinación generalmente mejora el rendimiento de los sistemas base que lo forman.

	F_1 - MCO						
Dataset	**Azure**	**Bing**	**CoreNLP**	**MCloud**	**SentiStr**	**Syuzhet**	**Vader**
debate	0.436	0.458	0.400	**0.462**	0.405	0.429	0.437
english_dailabor	0.656	0.624	0.428	0.659	0.648	0.581	**0.677**
pang_movie	0.516	0.481	**0.635**	0.489	0.366	0.503	0.439
sentistrength_bbc	0.393	0.471	0.407	0.407	**0.557**	0.429	0.399
sentistrength_digg	0.464	0.490	0.446	0.501	0.518	0.494	**0.521**
sentistrength_myspace	0.512	0.449	0.410	0.536	**0.598**	0.543	0.560
sentistrength_twitter	**0.590**	0.552	0.404	0.584	0.559	0.549	0.588
sentistrength_youtube	0.564	0.540	0.493	0.584	**0.591**	0.521	0.580
tweet_semevaltest	0.521	0.565	0.416	0.594	0.575	0.542	**0.612**
vader_amazon	0.557	**0.586**	0.516	0.564	0.543	0.501	0.571
vader_movie	0.439	0.458	**0.550**	0.453	0.438	0.451	0.445
vader_nyt	0.527	0.522	0.489	**0.543**	0.481	0.526	0.533
vader_twitter	0.547	0.675	0.326	0.686	0.669	0.694	**0.748**

Tabla 4: Resultados F_1 de cada uno de los MCO en todos los conjuntos de datos.

RESULTADOS DE E^{2}SAM. En la tabla 5 se aprecia cómo E^{2}SAM no solo mejora a los dos operadores de combinación base empleados en la experimentación en doce de los trece conjuntos de datos, sino que también se aprecia claramente que E^{2}SAM mejora en once de los trece conjuntos de datos al mejor MCO en cada conjunto. Esto indica que claramente E^{2}SAM es capaz de encontrar una buena distribución de pesos que pondere la contribución de cada MCO de forma que optimice el rendimiento general de la agregación.

4.4.3 *Significancia de los resultados y contribución de los modelos base*

ANÁLISIS DE SIGNIFICANCIA. Para comprobar si el empleo de E^{2}SAM supone diferencias significativas con respecto a usar un MCO de forma individual realizamos un test estadístico McNemar [96]. Este test detecta si hay diferencias significativas entre una muestra binaria antes de una experimentación u observación y esa misma muestra tras la observación. En nuestro caso, estas muestras binarias serán por un lado el 'acierto' o 'no acierto' de la polaridad por parte del mejor MCO y por otro lado, el 'acierto' o 'no acierto' de la polaridad por parte de E^{2}SAM, mostrando de esta manera si existen diferencias significativas en la distribución de los errores.

Considerando un p-value < 0.01, E^{2}SAM presenta un cambio en la distribución de los errores con respecto al mejor MCO en seis conjuntos de datos en más

	F_1				
Dataset	**AVG**	**NEUTY**	**E²SAM**	**M. MCO**	**M. SAM nombre**
debate	0.450	0.468	0.448	**0.462**	MCloud
english_dailabor	0.707	0.597	**0.724**†	0.677	Vader
pang_movie	0.462	0.527	**0.637**†(12)	0.635	CoreNLP
sentistrength_bbc	0.468	0.476	0.496	**0.557**	SentiStr
sentistrength_digg	0.536	0.550	**0.555**	0.521	Vader
sentistrength_myspace	0.529	0.568	**0.608**†	0.598	SentiStr
sentistrength_twitter	0.633	0.593	**0.640**†	0.590	Azure
sentistrength_youtube	0.627	0.586	**0.629**	0.591	SentiStr
tweet_semevaltest	0.635	0.576	**0.645**	0.612	Vader
vader_amazon	0.554	0.557	**0.597**†	0.586	Bing
vader_movie	0.527	0.497	**0.562**†	0.550	CoreNLP
vader_nyt	0.537	0.506	**0.545**†	0.543	MCloud
vader_twitter	0.699	0.589	**0.753**	0.748	Vader

Tabla 5: F_1 obtenido con los modelos *baseline* (AVG y NEUTY) y E²SAM. También se muestran los resultados obtenidos por el mejor MCO en cada conjunto de datos (M. MCO) y el nombre de ese MCO (M. MCO nombre). El símbolo †significa que existen diferencias significativas en los resultados según un test de McNemar con siderando un (p-value < 0.01).

de 15 de las 30 repeticiones. En `pang_movie`, ocurre en 12 de las 30 repeticiones. Por lo tanto, podemos concluir que nuestra propuesta E²SAM mejora el rendimiento de los métodos MCO de forma individual así como otros operadores de combinación de la literatura, mostrando que nuestra hipótesis inicial es correcta.

ANÁLISIS DE LA DISTRIBUCIÓN DE PESOS. En la figura 6 se muestra cómo E²SAM asigna un peso a cada MCO y es capaz de detectar el dominio del texto de entrada. Este hecho se ve reflejado en que E²SAM asigna más peso a aquellos MCO cuyo entrenamiento ha sido realizado en textos de la misma naturaleza que el de evaluación. Por ejemplo, vemos que CORENLP obtiene más del 80 % del peso cuando se evalúa sobre textos de opinión de películas (`pang_movie` y `vader_movie`). Algo similar ocurre cuando el género del texto de entrenamiento y el de evaluación coinciden como ocurre con el MCO Vader cuando se evalúa en `vader_twitter`. En otros conjuntos como en `sentistrength_digg`, `sentistrength_twitter` y `tweet_semevaltest`, vemos que E²SAM distribuye de forma más uniforme los pesos entre todos los MCO.

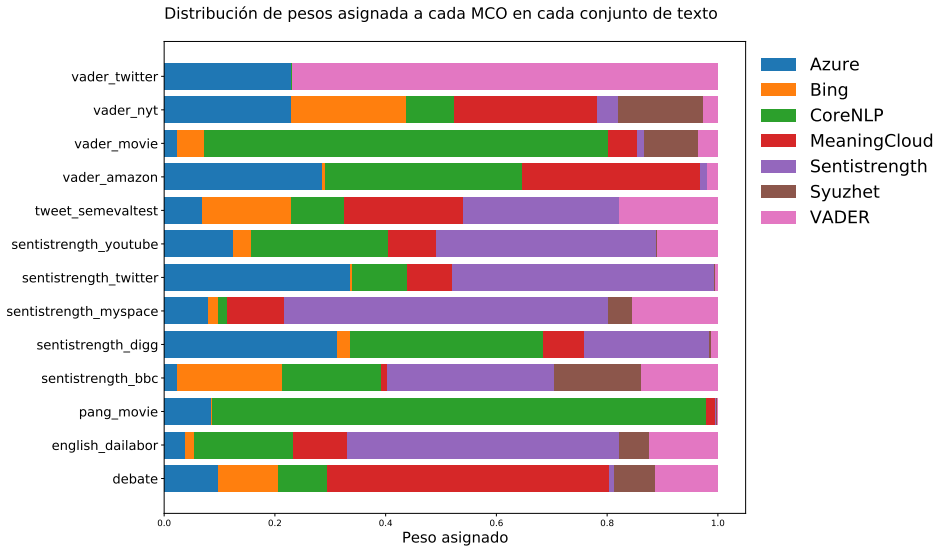

Figura 6: Distribución de pesos asignados a cada MCO en cada uno de los conjuntos de datos por parte de nuestra propuesta E^{2}SAM.

4.5 CONCLUSIONES

Un clasificador de la polaridad de la opinión expresada en un texto, y de forma particular un MCO, es altamente dependiente del dominio en el que es evaluado. Esto significa que el rendimiento es mucho más pobre si se evalúa sobre un dominio o género de texto distintos a los que el MCO ha sido especializado, llevándonos esto a un problema de adaptación al dominio. De esta manera, la hipótesis principal de nuestra propuesta es que el combinar varios MCO y el hacerlo de manera que cada MCO participe en la polaridad final según el dominio del texto de entrada, va a suponer una solución al problema de adaptación al dominio.

Los resultados de los MCO de forma individual hacen patente esta gran dependencia del dominio comentada, mostrando estos métodos rendimientos muy distintos dependiendo del conjunto de datos sobre el que son evaluados. A partir de este hecho, consideramos que el construir un modelo de combinación de clasificadores, donde se aproveche la diversidad de los MCO, puede mejorar el rendimiento de estos.

Sin embargo, en los resultados se muestra cómo las dos aproximaciones de combinación empleadas como modelos base solo superan al mejor MCO en cinco de los trece conjuntos de datos. En este capítulo se ha presentado nuestra con-

tribución E^{2}SAM, una propuesta metodológica que realiza un modelo de combinación en el que cada MCO tiene una participación en la polaridad final proporcional al peso asignado por un AE. Este AE realiza una optimización de la distribución de pesos asignada a los MCO, considerando como función objetivo la métrica de evaluación F_1. Los resultados de nuestra experimentación muestran cómo E^{2}SAM mejora en once de los trece conjuntos de datos al resto de aproximaciones probadas, tanto los modelos de combinación base empleados como al mejor MCO en cada conjunto de texto. De esta manera, se demuestra nuestra hipótesis inicial presentada en la introducción de este capítulo en la sección 4.1 y contribuimos a una de las subtareas dentro de la síntesis de opiniones, dando una solución al problema de adaptación al dominio por parte de los MCOs preentrenados.

E^{2}SAM fue propuesto en *E^{2}SAM: Evolutionary Ensemble of Sentiment Analysis Methods for domain adaptation* [88], artículo publicado en la revista *Information Sciences* con un índice de impacto de 6.795 indexada en la categoría *Computer Science, Information Systems*.

5

CONJUNTO DE DATOS PARA LA SÍNTESIS DE OPINIONES

La Síntesis de Opiniones es un campo de Procesamiento del Lenguaje Natural que trata de sintetizar las opiniones que los usuarios aportan sobre una entidad. Por lo tanto, una metodología o algoritmo de Síntesis de Opiniones, solo puede obtener conclusiones o conocimiento valioso cuando las opiniones sobre las que se aplica hacen referencia a una misma entidad, aspecto u otro nivel de granularidad de bajo nivel. Ante la falta de recursos para poder evaluar todo el flujo de una metodología de Síntesis de Opiniones, en este capítulo presentamos nuestra propuesta de un conjunto de opiniones sobre una única entidad en el dominio de restaurantes anotado a nivel de categorías de aspecto por oración y polaridad por oración. De esta manera, este conjunto de datos es un conjunto que puede emplearse para evaluar distintas posibles etapas dentro de una metodología de Síntesis de Opiniones y extraer conclusiones valiosas.

5.1 INTRODUCCIÓN

La SO es un campo del AO que pretende resumir qué se opina con respecto a una entidad y sus distintos aspectos. Además, es un campo que puede dividirse en varias subtareas, dando esto lugar a la necesidad de evaluar de forma distinta las propuestas a cada una de las tareas en un flujo de trabajo de SO.

Para que se puedan obtener conclusiones válidas a partir de un resumen de opiniones, estas opiniones deben tener en común la entidad, es decir, las conclusiones serán más precisas y valiosas cuanto más fina sea la granularidad de la entidad. Por ejemplo, las conclusiones que se pueden extraer de un conjunto de opiniones sobre un hotel en concreto serán más valiosas que las que se pueden extraer de un conjunto de opiniones sobre varios hoteles.

Existen diversas propuestas de conjuntos de opiniones que se emplean para la evaluación de propuestas para la SO. En [5], los autores presentan un conjunto de opiniones llamado SPACE para SO en el dominio de hoteles agrupado por entidad y cuyo conjunto de evaluación consiste en resúmenes hechos por anotadores. SPACE sin embargo no presenta categorías de aspecto y polaridades anotadas. También los mismos autores presentaron en [4] OPOSUM, un conjunto de opiniones en varios dominios (televisión, botas, teclados, etc.) con categorías de aspectos, polaridades y resúmenes asociados a algunas entidades. Sin embargo, OPOSUM muestra pocas opiniones por entidad.

Ante la escasez de conjuntos de opiniones agrupados por entidad con categorías de aspectos y polaridades anotadas, en esta tesis proponemos un conjunto de opiniones en el dominio de *Restaurantes* en inglés sobre un único restaurante anotado a nivel de categoría de aspecto y de polaridad. Con esta contribución proporcionamos un conjunto de opiniones sobre el cual se pueden evaluar distintas etapas de la SO, como son la clasificación de la polaridad y la detección de aspectos, y además, obtener resúmenes con respecto a una entidad, pudiendo así extraer conclusiones valiosas.

Este capítulo se estructura de la siguiente manera: en la sección 5.2 relacionamos nuestro conjunto de opiniones con otros conjuntos de opiniones del dominio de restaurantes. En la sección 5.3 se muestran distintas características de interés de nuestro conjunto propuesto. En la sección 5.4 se muestra la guía de anotación así como el acuerdo entre anotadores. Finalmente, en la sección 5.5 mostramos las conclusiones obtenidas a partir de nuestra contribución.

5.2 ORCO: CONJUNTO DE OPINIONES DEL DOMINIO DE RESTURANTES PARA SO

El dominio de restaurantes es un dominio de opiniones recurrente en distintas tareas de ABSA. En [48] los autores proponen un conjunto de opiniones en el dominio de restaurantes para la tarea de detección de categorías de aspectos, obteniendo las opiniones de la plataforma *CitySearch*[1]. Por otro lado, el dominio de restaurantes también es muy común en SemEval [113, 114, 142], *workshop* muy popular en el campo de PLN. Este conjunto también recoge opiniones de *CitySearch*, siendo estas opiniones anotadas para diversas tareas de PLN, concretamente de ABSA, como son detección de aspectos a nivel de término y de categoría, clasificación de la polaridad, etc.

Sin embargo, las propuestas de conjuntos de opiniones mencionadas, pertenecen a un grupo heterogéneo de restaurantes, sin existir ningún tipo de caracterización que agrupe por restaurantes las opiniones. Por lo tanto, a pesar de poder evaluar tareas relacionadas con SO, el flujo de trabajo de una metodología o algoritmo de SO no podría ser evaluado al completo debido a que generar resúmenes de un conjunto heterogéneo de restaurantes no daría lugar a conclusiones valiosas.

En esta tesis presentamos nuestra contribución **O**ne **R**estaurant **Co**rpus (ORCo[2]), un conjunto de opiniones en inglés del dominio de restaurantes obtenido a partir de opiniones de TripAdvisor. ORCo presenta opiniones de una única entidad, es decir, de un único restaurante de la ciudad de Londres. ORCo está anotado a nivel de oración y contiene las etiquetas de las distintas categorías de aspecto mencionadas en cada oración así como de la polaridad de la opinión reflejada en la oración.

5.3 CARACTERÍSTICAS DE ORCO

ORCo es un conjunto de opiniones formado por 50 opiniones en total, donde 25 de ellas tienen valoración de 1 estrella en TripAdvisor (opiniones negativas) y 25 de ellas de 5 estrellas (opiniones positivas). Estas 50 opiniones son divididas en 277 oraciones. En cada oración se anotan todas las categorías de aspectos mencionadas en la oración. Las categorías de aspctos definidas son *Food, Desserts, Ambience, Staff, Location, General, Price, Drinks, Desserrts* y *None*. Las oraciones también tienen anotadas la polaridad expresada en las mismas, pudiendo esta ser positiva (1), negativa (-1) o neutra (0). En la tabla 6 se muestra un resumen de estas características.

1 http://www.citysearch.com/world
2 ORCo está disponible en https://github.com/ari-dasci/OD-One-Restaurant-Corpus

Características ORCo	Valor
Número oraciones	277
Número opiniones	50
Número opiniones pos.	25
Número opiniones neg.	25
Aspectos	Comida, Postres, Ambiente, Servicio, Localización, General, Precio, Bebidas, Postres y Ninguno
Polaridades	-1,0,1

Tabla 6: Principales características del conjunto de opiniones ORCo.

ORCo está compuesto por 127 oraciones de polaridad negativa, 117 de polaridad positiva y 33 de polaridad neutra. Por otro lado, como ya se ha mencionado con respecto a las categorías de aspectos anotadas, cada oración está anotada con uno o varios aspectos mencionados, siendo así ORCo un conjunto de opiniones válido para evaluar un enfoque multi-etiqueta. En la tabla 7 se muestra el número de veces que cada aspecto es mencionado en ORCo.

Aspectos	Nº de veces mencionado
General	81
Servicio	82
Comida	57
Ambiente	40
Bebidas	17
Precio	18
Postres	3
Localización	3

Tabla 7: Número de veces que aparece mencionada cada categoría de aspecto en ORCo.

5.4 ANOTACIÓN DE ORCO: GUÍA Y ACUERDO ENTRE ANOTADORES

La anotación de ORCo fue llevada a cabo por tres anotadores, concretamente tres investigadores miembros del *Instituto Andaluz de Ciencia de Datos*. A la hora de realizar la anotación, es necesario establecer una serie de instrucciones y protocolos para que la anotación sea de la mayor calidad posible. En la sección 5.4.1

explicamos las instrucciones que los anotadores debieron seguir. Ante el posible desacuerdo entre anotadores, es necesario proveer métricas que midan el grado de acuerdo en la anotación [108] y, por lo tanto, la calidad de la anotación. En las secciones 5.4.2 y 5.4.3 se describen las dos métricas empleadas para medir el grado de acuerdo en la anotación multi-etiqueta de categorías de aspectos y en la anotación de la polaridad, respectivamente.

5.4.1 *Guía de anotación*

A la hora de realizar la anotación de un conjunto de datos es necesario preestablecer unas indicaciones que faciliten el acuerdo entre anotadores. Con estas pautas se pretende que los anotadores tengan una guía para actuar en caso de ciertas ambigüedades, como puede ser la aparición de polaridades opuestas en una misma oración o la aparición de aspectos de forma implícita. Estas indicaciones son clave para obtener un conjunto de datos de la máxima calidad posible. A continuación, se detallan las indicaciones que los anotadores recibieron para la anotación de ORCo en sus distintos niveles:

- **Anotación de la polaridad.** Las oraciones deben ser anotadas como que emiten una polaridad positiva (1), negativa (-1) o neutra (0). La polaridad positiva debe indicarse cuando se expresa felicidad o tono positivo en la opinión con respecto a la entidad en general o con respecto a algún aspecto de la entidad; la polaridad negativa sigue la misma naturaleza pero cuando se expresa una emoción de insatisfacción o rechazo; la polaridad neutra debe anotarse cuando se realiza una descripción objetiva referente a la entidad. En el caso de que tanto polaridad positiva como negativa aparezcan en la oración, debe anotarse la que presente más intensidad. En caso de empate en la votación por voto mayoritario, los anotadores se reúnen con el fin de llegar a una decisión común.
- **Anotación de las categorías de aspectos.** Se les pide a los anotadores que seleccionen, de un conjunto base de categorías de aspectos, aquellas categorías de aspecto que son mencionadas en cada oración tanto explícita como implícitamente. Cada anotador dará lugar a un conjunto de categorías por oración. El conjunto final de categorías por oración será la unión de los conjuntos de cada uno de los anotadores.

5.4.2 *Evaluación de la calidad de la anotación de categorías de aspectos*

Para evaluar el consenso para la anotación de categorías de aspecto, es necesario tener en cuenta la naturaleza multi-etiqueta de la anotación. Por ello, empleamos para evaluar el consenso el valor α de Krippendorff [76]. Este valor, a diferencia de otras métricas de acuerdo entre anotadores, tiene en cuenta que si dos anota-

dores en una misma instancia coinciden en la anotación de ciertos aspectos y no están de acuerdo en otros, el desacuerdo sea menor que si ambos anotadores no estuviesen de acuerdo en ninguno de los aspectos anotados.

Dado un conjunto R de posibles respuestas (posibles categorías de aspectos en nuestro caso), las unidades u para cada instancia siendo estas formadas por las respuestas de todos los anotadores para esa instancia y el conjunto U como el conjunto de todas las unidades u, el valor α de Kripendorff viene dado por la expresión mostrada en la ecuación 14:

$$\alpha = 1 - \frac{D_0}{D_e} \tag{14}$$

donde D_0 es el desacuerdo observado:

$$D_0 = \frac{1}{n} \sum_{c \in R} \sum_{k \in R} \partial(c,k) \sum_{u \in U} m_u \frac{n_{cku}}{P(m_u, 2)} \tag{15}$$

donde $\partial(c,k)$ es en nuestro caso 1 si c y k son distintos y 0 en caso contrario, m_u es el número de respuestas en una unidad u, n_{cku} es el número de parejas (c,k) en la unidad u y P es la función de permutación. Por otro lado D_e viene dado por la expresión mostrada en la ecuación 16:

$$D_e = \frac{1}{P(n,2)} \sum_{c \in R} \sum_{k \in R} \partial(c,k) P_{ck} \tag{16}$$

siendo P_{ck} el número de veces que la pareja (c,k) puede realizarse y n el número total de elementos emparejables, es decir, en nuestro caso el número de categorías de aspectos posibles.

El valor de α de Krippendorff para ORCo es $\alpha = 0.7311$, siendo considerado según [76] un acuerdo aceptable debido a que $\alpha \geq 0.667$.

5.4.3 *Evaluación de la calidad de la anotación de la polaridad*

Para evaluar el acuerdo entre anotadores en la anotación de la polaridad, se emplea el coeficiente *multi-κ* de Fleiss [31, 108], un coeficiente que, a diferencia del α de Krippendorff, es una métrica comúnmente empleada para el cálculo del acuerdo en anotaciones multi-clase pero no multi-etiqueta. De esta manera, en el caso del valor *multi-κ* los anotadores están de acuerdo o en desacuerdo en el anotado de una misma instancia, no mostrando distintos niveles de acuerdo como sí ocurre con el valor α de Krippendorff. Este coeficiente se calcula según la expresión de la ecuación 17:

$$\kappa = \frac{\overline{P} - \overline{P}_e}{1 - \overline{P}_e} \tag{17}$$

donde $\overline{P}$ viene dado por la expresión mostrada en la ecuación 18:

$$\overline{P} = \frac{1}{Nn(n-1)}(\sum_{i=1}^{N}\sum_{j=1}^{k} n_{ij}^2 - Nn) \tag{18}$$

y $\overline{P}_e$ viene dado por las expresiones:

$$\begin{aligned} p_j = \frac{1}{Nn}\sum_{i=1}^{N} n_{ij}\,,\ 1 = \sum_{j=1}^{k} p_j \\ \overline{P}_e = \sum_{j=1}^{k} p_j^2 \end{aligned} \tag{19}$$

siendo N el número total de instancias, en nuestro caso, de oraciones, n el número de anotadores por oración, k el número de clases, en nuestro caso 3 clases (-1,0,1) y n_{ij} el número de anotadores que han asignado la clase j a la oración i.

ORCo obtiene un valor *multi-*κ=0.9041, considerado como un acuerdo casi perfecto según [79].

5.5 CONCLUSIONES

La poca cantidad de conjuntos de opiniones existentes para la evaluación de una metodología de SO y las posibles etapas que la conforman crea la necesidad de proveer conjuntos de opiniones válidos y de calidad. Estos conjuntos de opiniones, para poder obtener resúmenes con conclusiones valiosas, deben ser de una única entidad o estar agrupados por entidad.

En este capítulo hemos presentado ORCo, un conjunto de opiniones en el dominio de restaurantes, un dominio muy común en tareas típicas de ABSA y, hasta donde sabemos, no representado en recursos para la evaluación de algoritmos o metodologías de SO. ORCo puede emplearse para evaluar la detección de categorías de aspectos y la clasificación de la polaridad, además de dar la posibilidad de extraer resúmenes valiosos debido a que ORCo presenta opiniones de una única entidad. Con ORCo contribuimos a la ampliación de recursos para la evaluación de sistemas de SO, dando una solución al segundo subobjetivo de los tres propuestos inicialmente en esta tesis.

ORCo consigue un acuerdo entre anotadores aceptable para las categorías de aspecto y uno casi perfecto para la polaridad según los coeficientes α de Kripendorff y *multi-*κ de Fleiss, respectivamente. ORCo es empleado en la experimentación de ADOPS, una de las contribuciones de esta tesis y descrita en el capítulo 6, para la evaluación de una metodología para la SO.

ORCo además es un recurso de libre acceso presentado en el artículo *ADOPS: Aspect Discovery Opinion Summarisation methodology based on Deep Learning and Subgroup Discovery for generating explainable opinion summaries* [89], publicado en la revista *Knowledge Based Systems* con índice de impacto de 8.038 indexada en la categoría *Computer Science, Artificial Intelligence*.

6

GENERACIÓN DE SÍNTESIS DE OPINIONES EXPLICABLES

En este capítulo de la tesis mostramos nuestra propuesta metodológica en la cual abarcamos el subobjetivo de esta tesis relacionado con la generación de resúmenes de opiniones estructurados y explicables. Nuestra contribución en una metodología modular que emplea técnicas de distinta naturaleza como son el *Deep Learning* y la minería de datos descriptiva. El fin de nuestra metodología es generar resúmenes de opiniones estructurados formados por un conjunto de reglas interesantes. Estas reglas hacen que el resumen obtenido sea fácilmente interpretable y, gracias al acompañamiento de métricas que cuantifican estas reglas, también explicable de cara al usuario. Nuestra metodología, además, es débilmente supervisada, suponiendo esto que sea una metodología fácilmente adaptable al dominio. De esta manera aportamos una solución al problema de adaptación al dominio.

6.1 INTRODUCCIÓN

En capítulos anteriores se han presentado dos contribuciones que se relacionan con los dos primeros subobjetivos propuestos en el inicio de esta tesis. Con E^{2}SAM realizamos una contribución para la clasificación de la opinión, una potencial subtarea dentro de la SO. Con ORCo, presentamos un recurso para la evaluación de sistemas o metodologías de SO. Tras estos dos subobjetivos, nos enfrentamos al tercer subobjetivo que además se relaciona de forma más directa con el objetivo principal de esta tesis: la obtención de resúmenes de opiniones de forma que estos sean estructurados, interpretables, explicables y mediante métodos fácilmente adaptables a distintos dominios.

El ABSA se compone de un diverso número de tareas y subtareas y el fin de estas es la obtención de información referente a la opinión expresada por distintos sujetos opinadores con respecto a distintos aspectos o características de una entidad. Sin embargo, como ya mencionamos en la introducción de esta tesis, presentada en el capítulo 1, las distintas tareas pertenecientes al ABSA ofrecen una salida dispersa y heterogénea, es decir, el ABSA directamente no aporta un conocimiento lo suficientemente sintetizado como para que sea fácil por parte del usuario extraer conclusiones.

La SO [81] se encarga de acabar con esta información heterogénea y dispersa, pretendiendo dar una visión sintetizada y general de esta información, de manera que esta sea, ahora sí, fácilmente interpretable y general para el usuario. La SO está especialmente centrada en aportar resúmenes que giran en torno a los distintos aspectos de una entidad, existiendo técnicas tanto abstractivas como extractivas que aportan resúmenes en forma de texto. Sin embargo, nosotros planteamos que un resumen estructurado y esquemático dará una visión más interpretable al usuario. Esta interpretabilidad, unida a un conjunto de métricas que cuantifiquen los resúmenes obtenidos, harán que estos resúmenes sean explicables.

En este capítulo presentamos nuestra contribución metodológica para la SO en la que hibridamos el empleo de técnicas de distinta naturaleza en distintas etapas de la metodología: (1) técnicas de *Deep Learning* para la DA y agrupación de los mismos en categorías más generales, y (2) empleo de técnicas de minería de datos descriptiva, concretamente de SD, para la obtención de resúmenes basados en reglas. Estas dos etapas y tipos de técnicas están unidas por una etapa intermedia consistente en la representación de los conjuntos de opiniones como *itemsets*.

Esta metodología además se presenta como una metodología modular, donde cada etapa o fase de la metodología puede ser afrontada con técnicas a gusto del usuario. Sin embargo, en la presentación de la metodología proponemos y su-

gerimos un enfoque débilmente supervisado, lo que aporta a nuestra propuesta metodológica una facilidad para la adaptación al dominio ya que no dependería de la existencia de conjuntos de opiniones etiquetados del dominio de estudio deseado.

En las siguientes secciones presentamos nuestra metodología. En la sección 6.2 hacemos una definición formal de la metodología así como de sus distintas etapas. En la sección 6.3 mostramos la configuración de la experimentación realizada. En la sección 6.4 hacemos un análisis de los resultados obtenidos de nuestra experimentación. Finalmente en la sección 6.5 detallamos las conclusiones principales de nuestro estudio.

6.2 METODOLOGÍA ADOPS

La obtención de conocimiento con respecto a opiniones es esencial a la hora de la toma de decisiones por parte de cualquier sujeto interesado en el producto o entidad sobre el que se opina. El campo de la Síntesis de Opiniones (SO) trata la obtención de resúmenes que ofrezcan conocimiento a partir de un conjunto de opiniones con respecto a una única entidad [81]. La SO, como campo dentro de la IA, también requiere del desarrollo de soluciones que cumplan los nuevos desafíos asociados a la IA, como son la democratización de la misma y la transparencia algorítmica.

En esta sección se presenta la metodología ADOPS (**A**spect **D**iscovery **OP**inion **S**ummarisation), cuyo fin es la construcción de resúmenes de opiniones de forma estructurada con respecto a una única entidad. Para conseguir dicha naturaleza estructurada y explicable, el resumen es representado por medio de un conjunto de reglas junto con una serie de métricas que cuantifican estas reglas.

Las reglas que la metodología ADOPS ofrece están formadas por antecedentes, que se refieren a la mención, tanto explícita como implícita, o 'no mención' de determinados aspectos en las opiniones, y un consecuente, que hace referencia a una propiedad de interés. En nuestro estudio, consideramos la polaridad del sentimiento expresado en la opinión como propiedad de interés. De esta manera, la metodología ADOPS realiza unos resúmenes que explican la relación de la mención de determinados aspectos con la polaridad expresada en la opinión.

Formalmente, la metodología ADOPS parte de un conjunto de opiniones D de distintos usuarios con respecto a una única entidad e, donde cada opinión $d_i \in D$ está asociada a una polaridad del sentimiento $p_i \in \{-1, 1\}$ y un conjunto de características o aspectos mencionados a_{ik} de e. De esta manera, D se puede representar de una forma estructurada considerando cada d_i como un conjunto de a_{ik} mencionados y su polaridad p_i, llamando a esta representación $D_{itemset}$. A partir de $D_{itemset}$, la metodología ADOPS tiene como objetivo construir un resumen de

opiniones empleando técnicas de SD, que representan el resumen de las opiniones como un conjunto de reglas de interés. Básicamente, la metodología ADOPS toma como entrada un conjunto de opiniones con respecto a una única entidad y genera un resumen estructurado empleando técnicas de SD.

La metodología ADOPS se enfrenta a la tarea de resumen de opiniones mediante la combinación de dos tareas de IA muy distintas. Por un lado, la tarea de detección de aspectos (DA) de ABSA, que extrae los aspectos mencionados en un texto. Por otro lado está la tarea de SD, cuyo objetivo es obtener una serie de reglas con respecto a la polaridad expresada. Estas dos tareas pueden aplicarse de forma independiente, lo que hace que nuestra aproximación sea modular.

Con el fin de conectar ambas tareas en un flujo, es necesario añadir un paso intermedio con el que se adapte la salida de la DA a la entrada de la tarea de SD. Por lo tanto, la metodología ADOPS consiste en las tres etapas siguientes, como también se muestra en la figura 7:

1. Detección de los aspectos mencionados en cada oración. La detección de aspectos realizada por nuestra implementación de ADOPS se describe en la sección 6.2.1.
2. Agrupar las oraciones y los aspectos por opinión, con la intención de representar el conjunto de opiniones como un conjunto de *items* tal y como mostramos en la sección 6.2.2. De esta manera, adaptamos la salida de la etapa de DA a la entrada requerida por los métodos de SD.
3. Detección de reglas de interés empleando técnicas de SD, como mostramos en la sección 6.2.3 a partir del *itemset*.

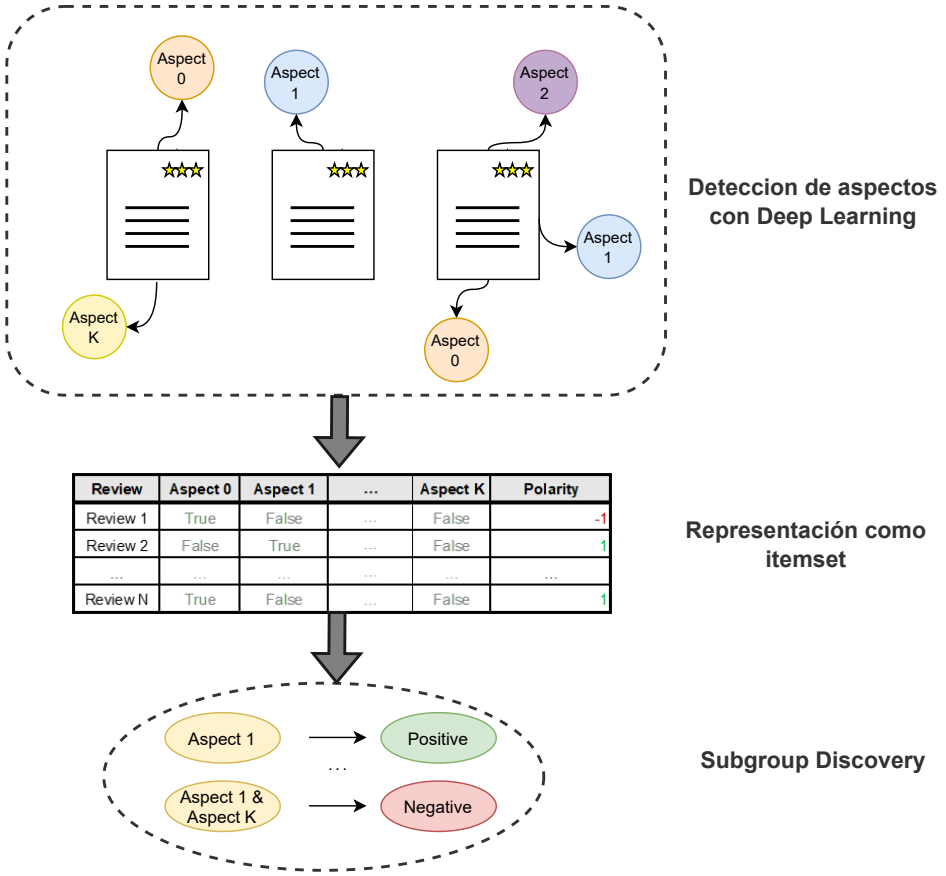

Figura 7: Diagrama del flujo de la metodología ADOPS. Se parte de un conjunto de opiniones con respecto a una entidad a las cuales se les realiza una detección de categorías de aspectos con DL. Posteriormente, las opiniones son representadas como itemsets donde cada opinión se caracterizará como la presencia o no presencia de determinados aspectos. Finalmente, se emplean técnicas de SD para la obtención de reglas interesantes a partir de la representación como itemset.

6.2.1 *Detección de aspectos con $ABAE_{AS}$*

La DA es un paso clave en el campo de *Aspect-Based Opinion Summarisation* debido a que se encarga de obtener los aspectos mencionados en una opinión, componente principal de estos resúmenes.

Como ya se ha mencionado, la metodología ADOPS es modular y se puede implementar con distintos métodos pre-entrenados o *ad-hoc*. La implementación del paso de DA debe cumplir las siguientes condiciones:

1. Debe ser una aproximación fácilmente adaptable al dominio. De esta manera, el modelo será independiente del dominio de las opiniones. En nuestra experimentación empleamos una aproximación débilmente supervisada, haciendo así que nuestro sistema no dependa de la disponibilidad de conjuntos de datos anotados.
2. Se deben categorizar los términos aspecto en un conjunto reducido e informativo de categorías de aspectos, dando lugar así a un resumen más genérico y significativo combinando estas categorías de aspecto a la polaridad de la opinión.

En nuestra propuesta implementamos la fase de DA con ABAE_{AS} basándonos en [4], una adaptación del método ABAE [55] para la extracción de términos de aspecto y categorización de aspectos. La descripción de ABAE y ABAE_{AS} se puede ver en la sección 3.3.1.3.

En nuestra implementación de ABAE, la DA se realiza a nivel de oración, dividiendo de esta manera cada opinión en un conjunto de oraciones. Por lo tanto, ABAE_{AS} devuelve una distribución de probabilidad p_t para cada una de las oraciones de una opinión, clasificando una oración como que un determinado aspecto es mencionado. En nuestra implementación, solo consideramos un aspecto por oración, asumiendo que la probabilidad corresponde al único aspecto mencionado en la oración.

La figura 8 muestra un ejemplo de cómo la DA funciona en la metodología ADOPS. La opinión de entrada al modelo es dividida en oraciones. Considerando que la matriz de aspetos se ha inicializado con los aspectos *Servicio*, *Bebidas* y *Comida*, ABAE_{AS} devuelve una distribución de probabilidad sobre cada una de las oraciones. En este caso particular, la oración *La comida estaba bastante buena* obtiene una distribución de probabilidad tal que $p_{servicio} = 0.05$, $p_{bebida} = 0.1$ y $p_{comida} = 0.85$, indicando que el aspecto *Comida* es el aspecto mencionado en la oración debido a su mayor probabilidad.

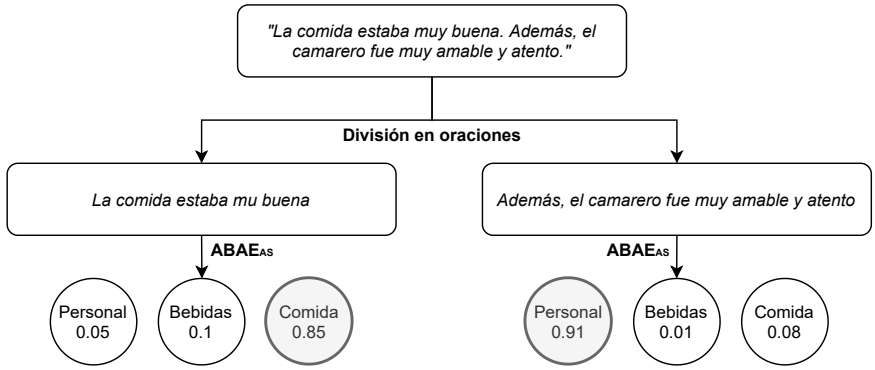

Figura 8: En la fase de DA de ADOPS, cada opinión es dividida en oraciones, a las que mediante ABAE_{AS} se les asocian las categorías de aspectos detectadas.

6.2.2 *Representación como itemset*

Tras haber extraído los aspectos mencionados en cada una de las oraciones, debemos adaptar esta información para que sirva de entrada al método de SD. Para hacerlo, consideramos que las categorías de aspectos son *items* que aparecen o no aparecen en una opinión. Por lo tanto, nuestro objetivo es representar el conjunto de opiniones como un conjunto de *items* o *itemset*.

Para esto, agrupamos las oraciones y los aspectos mencionados en estas frases por opiniones. De esta manera, representamos una opinión d_i como un un vector de características discretas binarias $(a_{i1}, a_{i2}, ..., a_{ik})$ y una polaridad p_i, donde $a_{ij} \in \{0, 1\}$, $\forall j \in \{1, 2, ..., k\}$ indica si el aspecto a_j es mencionado en la opinión d_i y $p_i \in \{-1, 1\}$ corresponde a la polaridad del sentimiento expresado en la opinión. La tabla 8 muestra varias opiniones representadas en la forma de *itemset*, donde la columna Polaridad de opinión se refiere a la polaridad global de la opinión. El resto de las columnas se refieren a los aspectos de interés.

6.2.3 *Generación de resúmenes de opiniones estructurados con SD*

El principal objetivo de la metodología ADOPS es resumir un conjunto de opiniones con respecto a una entidad de una manera estructurada. Para ello, consideramos que existen relaciones entre la mención de determinados aspectos que explican la polaridad del sentimiento de una opinión.

Estas particularidades hacen que los algoritmos de SD sean ideales para este análisis. Estas técnicas extraen reglas de interés, que explican una propiedad objetivo por medio de patrones y relaciones entre variables de un subgrupo de

Opinión	Servicio	Bebidas	Comida	Polaridad opinión
Food was really good. What's more, the waitress was really nice and very attentive.	1	0	1	1
Wine tasted like dirty water. I have never drunk anything worse.	0	1	0	-1
Wine list was very limited. Also, waitress treated us like we were children.	1	1	0	-1
You must try the steak.	0	0	1	1
The soup was disgusting. I told the waitress to take it away and she looked at me like I was a criminal. In addition, wine was not very tasty.	1	1	1	-1

Tabla 8: Ejemplo de una representación como *itemset* de varias opiniones ejemplo en inglés.

individuos. Estos algoritmos proporcionan un conjunto de reglas que, junto a una serie de métricas de calidad, dan informaciónn explicable y descriptiva.

En nuestra propuesta, consideramos que la polaridad del sentimiento de las opiniones es la propiedad objetivo o de interés y que los aspectos mencionados en las opiniones son las variables cuyas relaciones explican el sentimiento expresado. Como resultado, se obtiene un resumen estructurado consistente en un conjunto de reglas y métricas de calidad que sintetizan el sentimiento con respecto a una entidad específica.

Los resúmenes proporcionados por la metodología ADOPS son en su esencia un conjunto de reglas. Los antecedentes de estas reglas se refieren a los aspectos mencionados o no mencionados en el conjunto de opiniones. Por otro lado, el consecuente es el sentimiento asociado a las opiniones donde se mencionan los aspectos. La figura 9 muestra un ejemplo específico de reglas que cubren los 2 primeros ejemplos de la tabla 8. Estas reglas quieren decir que cuando los aspectos *Comida* y *Servicio* se mencionan de forma conjunta, entonces la opinión es positiva y que cuando el aspecto *Bebidas* es mencionado, entonces la opinión es probablemente negativa. Además, estas reglas están acompañadas por métricas que cuantifican la cobertura de las reglas. Por ejemplo, el valor de confianza $Conf. = 0.73$ de la regla {*Comida*=1,*Servicio*=1} $\rightarrow$ {Positivo} significa que el 73 % de las veces que ambas categorías de aspecto son mencionadas, el sentimiento asociado es positivo. Por lo tanto, *Comida* y *Servicio* normalmente reciben opiniones positivas de los clientes del restaurante.

Nuestro problema de SD se compone de características discretas binarias, que representan la presencia o ausencia de un aspecto en una opinión y una propie-

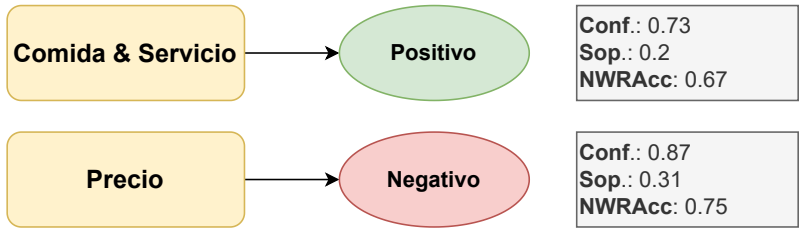

Figura 9: Dos ejemplos de la estructura de las reglas obtenidas por ADOPS. Los resúmenes se mostrarán como la relación entre la mención de ciertas categorías de aspectos y la polaridad de las opiniones donde estos aspectos son mencionados. Los resúmenes además están acompañados de métricas que representan la calidad de la regla y dan soporte estadístico a los resúmenes así como una mayor explicabilidad.

dad objetivo o de interés binaria, que se refiere a la polaridad del sentimiento de una opinión. Para dar solución a este problema, debido a la naturaleza modular de la metodología ADOPS, se puede emplear cualquier algoritmo de SD que permita el empleo de características discretas binarias. En nuestro estudio se han considerado dos algoritmos de SD de distinta naturaleza, Apriori-SD y NMEEF-SD, ambos descritos en la sección 2.4.2.

Por un lado, Apriori-SD es un algoritmo cuya entrada por defecto deben ser características binarias. Por esta razón, además del hecho de que sea un método de SD clásico y de calidad, se ha elegido Apriori-SD como uno de los métodos de SD empleados en nuestro estudio.

NMEEF-SD está enfocado en el tratamiento de características continuas que son convertidas a etiquetas lingüísticas difusas. A pesar del hecho de que la representación de las opiniones de nuestra propuesta son características binarias discretas, empleamos NMEEF-SD en nuestro estudio porque: (1) extrae un conjunto de reglas más pequeño pero más preciso que otras aproximaciones, lo que da mejor interpretabilidad al conjunto de reglas; (2) acepta características categóricas como entrada (3) es una aproximación multi-objetivo por lo que optimiza varias métricas de SD; y (4) es un algoritmo de naturaleza no determinista por lo que permite explorar en umbrales de confianza y soporte más bajos con unos tiempos de ejecución menores que otras aproximaciones como Apriori-SD.

6.3 MARCO EXPERIMENTAL

En esta sección describiremos la configuración de los experimentos de las distintas fases de experimentación de nuestra propuesta científica. En la sección 6.3.1 haremos una descripción de los distintos conjuntos de opiniones empleados en nuestra experimentación y sus requisitos. En la sección 6.3.2 mostraremos una descripción de los modelos base y la configuración de estos, así como de $ABAE_{AS}$. Finalmente, en la sección 6.3.3 describimos la experimentación para la obtención de reglas con SD.

6.3.1 *Conjuntos de datos*

Para la evaluación de la metodología ADOPS, necesitamos conjuntos de datos de opiniones de distintas naturalezas. Primero, necesitamos un conjunto de tamaño significativo de opiniones, sin tener que estar anotadas, para generar los *word embeddings* y para el entrenamiento de $ABAE_{AS}$. Después de esto, se necesitan conjuntos de opiniones con categorías de aspectos etiquetadas para evaluar el paso de DA de la metodología ADOPS. Finalmente, para evaluar los resúmenes generados por las técncias de SD, se necesitan conjuntos de opiniones con respecto a una única entidad.

Los conjuntos de opiniones empleados los clasificamos en 3 categorías distintas dependiendo de la fase en la que se empleen estos. Estas categorías son:

1. **Conjuntos del dominio de entrenamiento (*InDomainData*).** Con el fin de entrenar las representaciones vectoriales de palabras (*word embeddings*) y el modelo $ABAE_{AS}$ para la DA, necesitamos conjuntos de opiniones grandes del dominio de interés. En la tabla 9 se muestran los conjuntos de datos de dominio empleados en nuestro estudio, así como el número de oraciones y el dominio de cada conjunto de opiniones. Los conjuntos de opiniones empleados son de los dominios de televisión (TV Oposum Train [4]), botas (Boots Oposum Train [4]), restaurantes (Restaurant CitySearch Train [48]) y monumentos (Monuments Train) formado por la unión de conjuntos de opiniones de La Alhambra y Sagrada Familia propuestos en [132].
2. **Conjuntos de test para la evaluación de la DA (*DAData*).** Son conjuntos de test con las categorías de aspectos etiquetadas. En la tabla 10 se muestran de forma resumida las principales características de estos conjuntos, mostrándose el número de oraciones (instancias), número de categorías de aspectos, dominio y las categorías de aspectos. En los *DAData* aparecen los dominios de televisión (TV Oposum Test [4]), Botas (Boots Oposum Test [4]) y restaurantes (Restaurant CitySearch Test [48] y ORCo [89]). Para evaluar los modelos em-

Conjuntos - Entrenamiento	Oraciones	Dominio
TV Oposum Train	597,007	TV
Boots Oposum Train	423,754	Botas
Restaurant CitySearch Train	279,885	Restaurantes
Monuments Train	234,246	Monumentos

Tabla 9: Conjuntos de opiniones de entrenamiento del dominio empleados para generar los *word embeddings* así como para la fase de DA por medio de $ABAE_{AS}$.

Conjuntos - Test	Oraciones	Dominio	Etiquetas de aspectos
TV Oposum Test	349	Televisiones	Image, Sound, Connectivity, Ease of use, Price, Apps Interface, Customer Service, Size/Look.
Boots Oposum Test	325	Botas	Color, Comfort, Durability, Look, Materials, Price, Size, Weather resistance.
Restaurant CitySearch Test	3,315	Restaurantes	Food, Staff, Anecdotes, Ambience, Price, Miscellaneous.
ORCo	247	Restaurantes	Food, Desserts, Drinks, General, Ambience, Staff, Location, Price.

Tabla 10: Conjuntos de evaluación de la etapa de DA de la meotolodgía ADOPS. Se muestran el número de oraciones o instancias, número de etiquetas, el dominio y las categorías de aspectos existentes.

pleando estos conjuntos de opiniones, se han suprimido todas las *stop words* y las oraciones con el aspecto *None*.

3. **Conjuntos de datos para la evaluación de la etapa SD (*SDData*).** Empleamos conjuntos de opiniones sobre una única entidad para así obtener resúmenes significativos. La tabla 11 muestra un resumen de las características de estos conjuntos, mostrando el número de opniones totales, positivas y negativas, número total de oraciones y dominio de las opiniones. Trabajamos con opiniones de los dominios de monumentos (SagradaFamilia y The Alhambra [132]) y restaurantes (ORCo [89]). Hemos considerado las valoraciones de 5 estrellas según TripAdvisor como positivas y las de 1 estrella como negativas, haciendo filtrado del resto de valoraciones para evitar ambigüedad en las mismas. También se ha reducido el desbalanceo de los conjuntos de La Alhambra y Sagrada Familia, realizando un inframuestreo de las opiniones de 5 estrellas.

Datasets	Nº Opiniones	Op. Pos.	Op. Neg.	Oraciones	Dominio
SagradaFamilia	382	199	183	1723	Monumentos
The Alhambra	239	120	119	1950	Monumentos
ORCo	50	25	25	247	Restaurantes

Tabla 11: Estadísticas de los conjuntos de opiniones SDData.

6.3.2 *Detección de aspectos: Modelos Baseline, $ABAE_{AS}$ y configuración*

Para la etapa de DA, evaluamos los métodos experimentados sobre los conjuntos de Evaluación descritos en la sección 6.3.1. Las métricas empleadas para la evaluación son dos métricas comúnmente usadas en problemas de clasificación como son el *Accuracy* y Macro-F_1. En nuestro estudio, comparamos $ABAE_{AS}$ con dos aproximaciónes base que también pueden ser inicializadas por medio de palabras semilla.

- **K-Means** [90] es un algoritmo no supervisado clásico usado para *clustering* cuyos centroides pueden ser fácilmente guiados mediante la inicialización de estos con los valores deseados. En nuestro caso, cada centroide representará un *embedding* de aspecto. A una oración se le asigna el aspecto cuyo *word embedding* es más cercano al *word embedding* promedio de la oración.
- **Guided-LDA** [64] se trata de una modificación de *Latent Dirichlet Allocation*. Incorpora elementos léxicos a priori que sesgan el modelo y lo guían a obtener *topics* o aspectos según palabras semilla. En nuestro estudio, empleamos 200 iteraciones de este método.

En nuestra experimentación entrenamos un modelo *word2vec* [98] para obtener las representaciones vectoriales de las palabras empleando para ello los conjuntos *InDomainData* previamente descritos. En el entrenamiento del modelo *Word2Vec* usamos un tamaño de vectores de 200, una ventana de 10 y un tamaño de muestras negativas de 5. Estas representaciones vectoriales de las palabras se emplean para modelar $ABAE_{AS}$ y K-Means. El vocabulario es reducido a las 9000 palabras más frecuentes, como hacen en [55]. Para K-Means y $ABAE_{AS}$, las representaciones vectoriales iniciales de los aspectos se obtienen mediante el cálculo del vector medio de los vectores de las palabras semillas empleadas para ese aspecto. Para todos los algoritmos se emplearon las mismas palabras semilla. En la tabla 12 se muestran los hiperparámetros empleados en $ABAE_{AS}$.

A la hora de evaluar los modelos para la DA, algunas de las oraciones en los conjuntos de *DAData* están categorizadas como que aparecen varias categorías de aspectos. En estos casos, consideramos que una predicción es correcta si la categoría predicha por el modelo es alguna de las que aparecen en la oración.

$ABAE_{AS}$ Parameters	**Value**
Épocas	15
Tamaño de batch	50
Muestras negativas	20
Learning rate	0.001

Tabla 12: Hiperparámetros empleados para el entrenamiento de $ABAE_{AS}$.

6.3.3 *Obtención de reglas con Descubrimiento de Subgrupos*

Para la evaluación de las reglas extraídas con SD se emplean los conjuntos de opiniones *SDData* tras haber hecho una detección de categorías de aspectos sobre los mismos con $ABAE_{AS}$.

Las categorías de aspectos de interés para cada conjunto de opiniones son las siguientes:

- ORCo: *Staff*, *Location*, *Ambience*, *Food*, *Generral*, *Dessert*, *Price* y *Drinks*.
- Alhambra y Sagrada Familia: *Architecture*, *Location*, *Staff*, *Ticketing*, *Price*, *Tourism Resources*, *General* y *Queues*.

En este estudio se han empleado los algortimos de SD Apriori-SD y NMEEF-SD, descritos en la sección 6.2.3. En la tabla 13 se muestra la configuración de parámetros de Apriori-SD. Estos parámetros han sido configurados de esta manera para mantener un equilibrio entre los tiempos de ejecución del algoritmo y la calidad de las reglas.

En cuanto a NMEEF-SD, usamos el paquete SDEFSR[1] de R y en la tabla 14 mostramos la configuración de parámetros empleada para la ejecución. Igual que con Apriori-SD, se ha tenido en cuenta mantener un equilibrio entre los tiempos de ejecución y la calidad de las reglas. Por otro lado, se han seleccionado las métricas de *Unusualness* y Significancia como métricas para la optimización objetivo tras haberse probado varias combinaciones de otras métricas.

6.4 RESULTADOS Y ANÁLISIS

En esta sección mostramos los resultados de la experimentación detallada en la sección 6.3. En la sección 6.4.1 mostramos la comparación en la etapa de DA de $ABAE_{AS}$ frente a los modelos *baseline* K-Means y Guided-LDA. En la sección 6.4.2 mostramos una comparación de la calidad de las reglas obtenidas por

1 https://CRAN.R-project.org/package=SDEFSR

Parámetros Apriori-SD	Valor
Min. Confidence	0.3
Min. Support	0.2
Modified WRAcc Threshold	0.25

Tabla 13: Configuración de parámetros para la ejecución de Apriori-SD.

Parámetros NMEEF-SD	Valor
Number Evaluation	10,000
Population Size	100
Mutation Probability	0.05
Crossover Probability	0.6
Min. Confidence	0.1

Tabla 14: Configuración de parámetros para la ejecución de NMEEF-SD.

APriori-SD y NMEEF-SD. Tras esta comparación, en la sección 6.4.3 exploramos cómo las reglas representan la realidad presente en los conjuntos de opiniones. En la sección 6.4.4 hacemos un análisis de los puntos débiles de la metodología ADOPS analizando sus errores. Finalmente, en la sección 6.4.5 comparamos nuestra propuesta con una propuesta similar presentada en [132].

6.4.1 *Resultados de la detección de las categorías de aspectos*

En la tabla 15 se muestran los resultados obtenidos con los modelos *baseline* K-Means y GuidedLDA, así como con $ABAE_{AS}$ sobre los conjuntos de datos de evaluación de la DA y entrenando los modelos sobre los conjuntos de entrenamiento de dominio correspondientes. Vemos cómo $ABAE_{AS}$ mejora generalmente los otros dos algoritmos a excepción del caso del Macro-F1 en TV Oposum, donde Guided-LDA obtiene un resultado ligeramente mejor. Estos resultados nos confirman que $ABAE_{AS}$ parece un modelo ideal para la implementación de la DA de forma débilmente supervisada de la metodología ADOPS.

	$ABAE_{AS}$		**K-Means**		**GuidedLDA**	
Test Sets	**Acc.**	**M. F1**	**Acc.**	**M. F1**	**Acc.**	**M. F1**
Restaurant CitySearch Test	**0.5427**	**0.3676**	0.4449	0.2793	0.5143	0.3384
TV Oposum Test	**0.6762**	0.6160	0.4699	0.4623	0.6446	**0.6166**
Boots Oposum Test	**0.4953**	**0.4643**	0.4461	0.3947	0.4769	0.4591
ORCo	**0.5263**	**0.3649**	0.4008	0.3024	0.4331	0.3198

Tabla 15: Resultados de $ABAE_{AS}$ y los dos modelos *baseline* de la etapa de DA.

En los resultados mostrados se aprecia que hay una precisión de los modelos limitada, lo que puede afectar negativamente a los resúmenes obtenidos por la metología ADOPS. Por lo tanto, es necesario estudiar si las categorías de aspectos aprendidas por ABAE_{AS} tienen sentido semánticamente hablando.

En las figuras 10, 11 y 12 mostramos, respectivamente de los dominios de *Botas*,*TV* y *Restaurantes*, las representaciones vectoriales de las palabras más cercanas del vocabulario a la representación vectorial de las categorías de aspectos obtenidas por el modelo. Esta representación gráfica representa las representaciones vectoriales tras una reducción de dimensionalidad con *Principal Component Analysis* a una dimensión de tres componentes. Se aprecia que los puntos están agrupados según la categoría de aspecto y que los grupos más parecidos semánticamente están más cerca entre ellos que los que son más distintos entre sí.

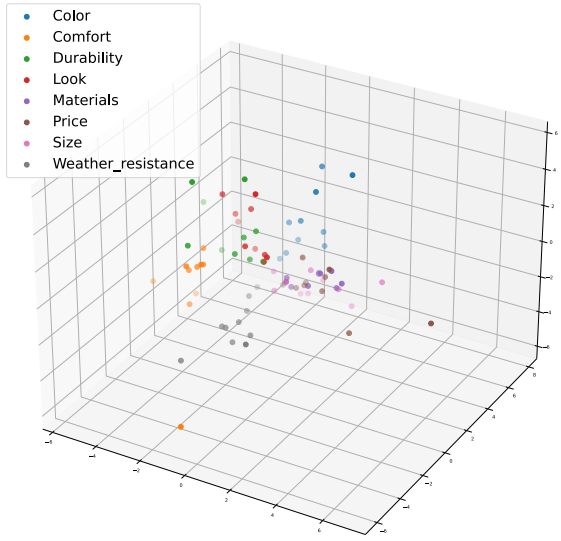

Figura 10: Gráfico 3D de las representaciones vectoriales de las palabras más cercanas a la representación vectorial de los aspectos tras una reducción de dimensionalidad en el dominio de *Botas*.

Por ejemplo, en la figura 10 se muestran grupos cercanos que pueden referirse a "estética", como son *Color* y *Look*. En la figura 11 también se aprecia algo similar, donde aspectos técnicos como *Image* tienen otros aspectos técnicos cercanos como *Sound*. En la figura 12 también se ve que *Desserts* y *Food* están bastante juntos a diferencia de otros aspectos como *Ambience*. También se aprecia que el aspecto *General* se mantiene en una posición intermedia, indicando que se trata de un aspecto muy neutro.

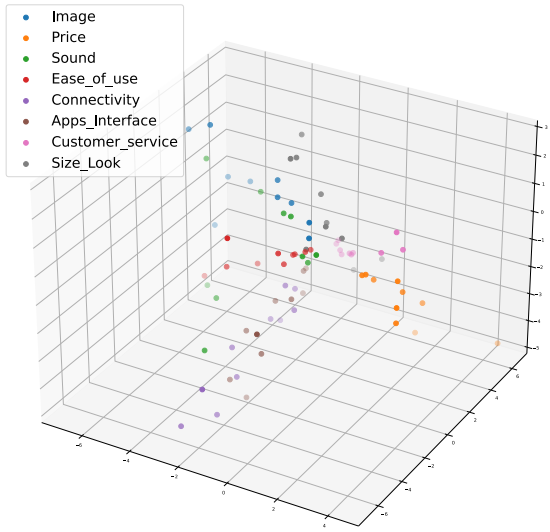

Figura 11: Gráfico 3D de las representaciones vectoriales de las palabras más cercanas a la representación vectorial de los aspectos tras una reducción de dimensionalidad en el dominio de *TV*.

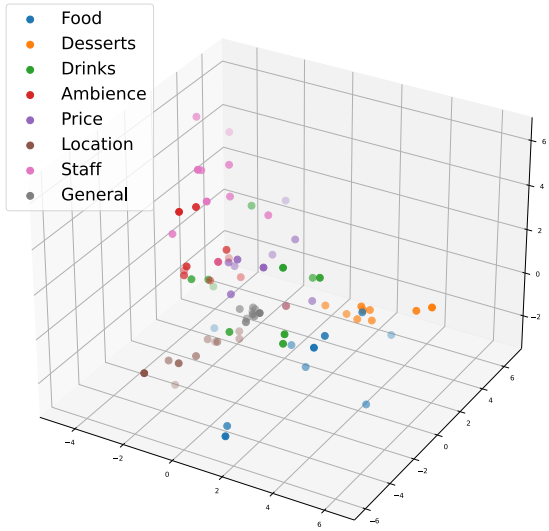

Figura 12: Gráfico 3D de las representaciones vectoriales de las palabras más cercanas a la representación vectorial de los aspectos tras una reducción de dimensionalidad en el dominio de *Restaurantes*.

Tras este análisis podemos concluir que, a pesar de la limitación en cuanto a métricas de la etapa de DE en la metodología ADOPS, se muestra que las representaciones vectoriales de los aspectos modeladas por $ABAE_{AS}$ tienen sentido semánticamente hablando.

6.4.2 *Resultados de los métodos de SD en la etapa de obtención de reglas*

En los experimentos realizados, Apriori-SD extrae unas 74 reglas en ORCo, 70 en La Alhambra y 75 en Sagrada Familia. En el caso de Apriori-SD se obtiene un conjunto de reglas grande, donde algunas de las reglas son de valor como {Drinks=0, Price=1} $\rightarrow$ {Negative} en ORCo, con valores de confianza y NWRAcc altos. Sin embargo, también se obtienen muchas reglas de baja calidad como por ejemplo la regla {Location=0} $\rightarrow$ {Negative} con un valor de confianza y NWRAcc de 0.42 en ORCo.

Por otro lado, NMEEF-SD obtiene 7, 8 y 6 reglas en los conjuntos ORCo, La Alhambra y Sagrada Familia, respectivamente. Este bajo número de reglas se debe a que NMEEF-SD tiene una tendencia a extraer conjuntos de reglas más pequeños pero de buena calidad. Por ejemplo, NMEEF-SD extrae la regla {General = 1 , Staff = 0} $\rightarrow$ {Positive} con un valor de confianza de 1 y un NWRAcc de 0.625.

En la tabla 16 se resumen las estadísticas más interesantes de las reglas obtenidas para tanto Apriori-SD como NMEEF-SD, mostrando medidas de interés medias de las reglas obtenidas. Como ya hemos mencionado, Apriori-SD extrae un conjunto de reglas más grande que NMEEF-SD. Esto significa que Apriori-SD obtiene unas medidas medias más bajas debido al exceso de reglas de poca calidad en el conjunto de reglas obtenidas. Por lo tanto, es necesario hacer un filtrado manual con el fin de obtener las reglas más significativas a partir de estos conjuntos grandes de reglas. Por otro lado, además de ofrecer un conjunto de reglas menor, NMEEF-SD también obtiene reglas con una media de antecedentes menor, lo que hace que estas reglas sean más interpretables y explicables. Por lo tanto, consideramos que los resultados obtenidos por NMEEF-SD son de una mayor calidad que los mostrados por Apriori-SD.

	Sop. Medio	**Conf. Media**	**NWRAcc Medio**	**Sign. Media**	**Núm. Reglas**
NMEEF-SD	0.2778	0.7984	0.6872	9.806	21
APriori-SD	0.2588	0.6291	0.5832	2.5954	219

Tabla 16: Número total de reglas obtenidas por ambos algoritmos de SD y métricas medias (*Soporte*, *Confianza*, *NWRAcc* y *Significancia*).

6.4.3 *Análisis de cómo las reglas reflejan las opiniones*

En los resultados mostrados en la sección 6.4.2 se muestra la comparativa de los dos métodos de SD considerados en este estudio. En esta sección, haremos un análisis más profundo de las reglas obtenidas y comprobaremos cómo estas reglas representan la realidad de los conjuntos de opiniones.

En la tabla 17 se muestran algunas reglas interesantes obtenidas en ORCo, como por ejemplo las tres primeras reglas, que indican que el aspecto *Drinks* no aparece mencionado en opiniones negativas. Esto puede significar que *Drinks* es un aspecto comúnmente positivo del restaurante. Por otro lado, el aspecto *Price* está muy relacionado con las opiniones negativas, por lo que podemos concluir que es uno de los aspectos más negativos reflejados en el conjunto de opiniones.

En las tablas 18 y 19 se muestran las reglas más significativas de los conjuntos de opiniones de La Alhambra y Sagrada Familia, respectivamente. Se aprecian varias reglas de interés. Por ejemplo, en ambos conjuntos el aspecto *Staff* parece especialmente mencionado en opiniones negativas. También, el hecho de que *Architecture* no aparezca en reglas negativas, indica calramente que este aspecto está más relacionado con opiniones positivas. Se observan también que las reglas positivas están especialmente sesgadas a la 'no mención' de determinados aspectos ({*General = 0*}, {*Staff = 0 , Price = 0*}). La regla de la tabla 18 {*General=1, Staff=0*} $\rightarrow$ {*Positive*}, muestra cómo NMEEF-SD puede extraer reglas de calidad en umbrales de soporte bajos, permitiendo la exploración de espacios de búsqueda que serían inasumibles en tiempos de ejecución por modelos deterministas como APriori-SD.

Las métricas mostradas en las tablas muestran generalmente valores altos. Por ejemplo, en el conjunto ORCo todos los valores de confianza son más altos de 0.6. Además, se obtienen reglas con un valor alto de NWRAcc, con todas las reglas mostrando un valor de NWRAcc $\geq$ 0.55. Esto significa que, según [26], obtenemos *conjuntos de contraste*, definidos por [37] como conjuntos de *items* que difieren de forma significativa en sus distribuciones entre grupos de individuos. Estos grupos, en nuestro caso, son definidos por el valor de la polaridad.

Como se ha mostrado, la metodología ADOPS es capaz de generar un resumen estructurado y explicable a partir de un conjunto de opinininones con respecto a una entidad. Además, estas reglas están acompañadas por métricas, que proporcionan un alto valor interpretativo a los resúmenes debido a que podemos asociar las reglas a medidas de frecuencia que cuantifican estas reglas. Sin embargo, también se tiene que comprobar si las reglas obtenidas reflejan realmente la realidad.

Reglas	Alg. SD	Sop.	Conf.	NWRAcc	Sign.
{Drinks=0, Price=1} → {Negative}	Ambos	0.30	0.94	0.78	6.38
{Drinks=0, Staff=1} → {Negative}	Ambos	0.38	0.76	0.76	3.08
{Drinks=0, General=1} → {Negative}	APriori-SD	0.30	0.83	0.74	3.79
{Price = 1} → {Negative}	Ambos	0.32	0.76	0.72	2.63
{Drinks=0, General=1, Staff=1} → {Negative}	APriori-SD	0.24	0.92	0.72	4.76
{Price=0} → {Positive}	Ambos	0.4	0.69	0.72	1.86
{Drinks=1} → {Positive}	Ambos	0.26	0.81	0.68	2.92
{Price=0 , Desserts=1} → {Positive}	APriori-SD	0.2	0.83	0.66	2.53
{Price=0 , Location=0} → {Positive}	APriori-SD	0.2	0.83	0.66	2.53

Tabla 17: Reglas obtenidas de ORCo y métricas de las mismas. La columna Alg. SD indica qué algoritmo ha obtenido cada regla. El resto de columnas se refieren al Soporte, Confianza, NWRAcc y Significancia.

Reglas	Alg. SD	Sop.	Conf.	NWRAcc	Sign.
{Staff=1} → {Negative}	Ambos	0.43	0.70	0.75	10.35
{Staff=1, General=0} → {Negative}	Ambos	0.34	0.76	0.73	13.09
{Staff=1, Ticketing=1} → {Negative}	Ambos	0.34	0.74	0.72	11.65
{Archit.=0, Staff=1} → {Negative}	Ambos	0.35	0.72	0.72	10.36
{Archit.=0, Staff=1, General=0} → {Negative}	APriori-SD	0.29	0.79	0.71	10.2
{Staff=0} → {Positive}	Ambos	0.31	0.82	0.75	17.82
{Staff=0 & Price=0} → {Positive}	APriori-SD	0.27	0.84	0.72	16.78
{Ticketing=0 & Price=0} → {Positive}	APriori-SD	0.20	0.74	0.63	6.58
{Ticketing=0} → {Positive}	APriori-SD	0.25	0.68	0.63	4.79
{General=1, Staff=0} → {Positive}	NMEEF-SD	0.13	1.0	0.63	17.95

Tabla 18: Reglas obtenidas a partir del conjunto La Alhambra y métricas de estas reglas. Archit. y T. Res. se refieren a las categorías de aspectos *Architecture* y *Tourism Resources*.

En la tabla 20 mostramos oraciones que han sido reflejadas en algunas de las reglas obtenidas. Por ejemplo, la regla {*Staff* = 1} → {*Negative*} de los conjuntos La Alhambra y Sagrada Familia, reflejan algunas oraciones donde claramente aparece el aspecto *Staff* y la polaridad de las opiniones es negativa. Además, mostramos también un ejemplo de una opinión cubierta por una regla con mención de varios aspectos, como es la regla {*Staff = 1 , Ticketing = 1*} → {*Negative*}, donde vemos que ambos aspectos son mencionados.

Las reglas del conjunto ORCo también muestran algunas oraciones donde la mención de los aspectos es clara. Por ejemplo, la regla {*Drinks* = 1} → {*Positive*}, a pesar de que el aspecto *Drinks* no es muy común en el conjunto de opinio-

Reglas	Alg. SD	Sop.	Conf.	NWRAcc	Sign.
{Staff=1} → {Negative}	Ambos	0.27	0.68	0.66	11.44
{Staff=1, General=0} → {Negative}	Ambos	0.21	0.74	0.65	12.94
{Staff=1, T.Res.=0} → {Negative}	APriori-SD	0.21	0.70	0.63	9.91
{Staff=1, Price=0} → {Negative}	APriori-SD	0.20	0.70	0.63	9.87
{General=0, Archit.=0} → {Negative}	APriori-SD	0.20	0.62	0.59	4.37
{Staff=0} → {Positive}	Ambos	0.40	0.65	0.66	7.14
{Staff=0, Price=0} → {Positive}	Ambos	0.31	0.68	0.65	7.91
{Staff=0 , Archit.=1} → {Positive}	APriori-SD	0.25	0.68	0.62	6.11
{Staff=0, Price=0, Archit.=1} → {Positive}	APriori-SD	0.21	0.70	0.61	6.56
{Staff=0, T.Res.=0, Price=0} → {Positive}	APriori-SD	0.24	0.65	0.59	3.78

Tabla 19: Reglas obtenidas del conjunto Sagrada Familia y sus métricas. Archit. y T. Res. se refieren a las categorías de aspectos *Architecture* y *Tourism Resources*.

nes, refleja opiniones de menciones del aspecto *Drinks*, como es la oración *"A wonderful selection of wines and some good advice"*.

En este análisis confirmamos que la metodología ADOPS es capaz de generar, por medio de reglas y métricas de calidad, un resumen explicable e interpretable que reflejan de forma general qué opinan los usuarios con respecto a una entidad y sus distintos aspectos.

6.4.4 *Análisis de errores de la metodología ADOPS*

En la sección 6.4.3 hemos mostrado cómo la metodología ADOPS genera conocimiento de valor por medio de las reglas producidas por las técnicas de SD. Sin embargo, como ya hemos observado en la sección 6.4.1, ABAE_{AS} detecta en muchos casos categorías de aspectos incorrectas debido principalmente a su naturaleza débilmente supervisada. Esto implica que este error se propague hacia la etapa de SD de la metodología ADOPS, suponiendo que algunas de las reglas generadas no reflejen fielmente el conjunto de opiniones.

SIMILITUD SEMÁNTICA ENTRE ASPECTOS. En la tabla 15 se muestra que los resultados obtenidos en el conjunto ORCo no son especialmente buenos en la etapa de DA, lo que significa que existen algunas oraciones clasificadas de forma errónea. Por ejemplo, en la oración *"Dessert was really good though."* se detecta la categoría de aspecto *Drinks*, debido probablente a la similitud semántica entre las categorías de aspectos *Food*, *Drinks* y *Desserts*, cuyos contextos son similares en muchos casos.

ORCo	
"The gin and tonic was very small for the price we paid and presented very averagely."	
"Nobody would choose to pay these prices to sit in a box room upstairs."	{Price = 1} → {Negative}
"Our meal was over £100 despite deciding to cut it short because of our upset."	
"We had the 'Tu me manques' as a sharing cocktail before the tasting menu, all so good!"	
"A wonderful selection of wines and some good advice."	{Drinks = 1} → {Positive}
La Alhambra	
"I wish I had captured the name of the woman behind the ticket off but she took the biscuit for being so rude."	
"How is it possible that staff in one of Spain's largest tourist attractions get offended for being asked to speak English?"	
"The staff told us that it was our fault that we come so late - how brazen!"	{Staff = 1} → {Negative}
"Tickets were almost sold out so we had to use the 19:00 slot... The staff worked so slowly that we entered..."	{Staff = 1, Ticketing = 1}→ {Negative}
Sagrada Familia	
"The lady at the ticket office just told us that there will be no time for an audio tour but you can go around."	{Staff = 1} → {Negative}

Tabla 20: Ejemplos de segmentos de texto cubiertos por algunas reglas.

TENDENCIA A LA DETECCIÓN DE DETERMINADOS ASPECTOS. En los conjuntos de La Alhambra y Sagrada Familia, se encuentran algunas opiniones que son erróneamente cubiertas por algunas reglas. Sin ir más lejos, en el conjunto de Sagrada Familia la opinión *"Wasted 1/2 hour tring to throw at this purchasing tickets online. No luck for me. No bucks for them. Look at the outside only, move on to something functional"* aparece cubierta por la regla { Architecture = 1 } → { Negative } . Este error podría deberse a un sesgo encontrado en ABAE_{AS} con respecto a la detección del aspecto *Architecture*, ya que en el 23 % de las oraciones se detecta esta categoría de aspecto.

APARICIÓN DE ASPECTOS IMPLÍCITOS. Otros errores son debidos a la presencia de aspectos implícitos, lo que supone una mayor complejidad especialmente para un modelo débilmente supervisado como es ABAE_{AS}. Por ejemplo, la oración *"overall not coming back and only recommended if you want to see fake plants in a box room."* claramente evalúa el restaurante en general. Sin embargo,

$ABAE_{AS}$ detecta la categoría *Ambience*, probablemente sesgado por la aparición de algunas palabras explícitas como *room* o *plants* que se relacionan fuertemente con la categoría *Ambience*. Este error provoca que esta oración aparezca como cubierta por la regla { Ambience = 1, Staff = 1 } → { Negative }.

Este claro déficit en el rendimiento de la DA indica que existe bastante margen de mejora y nos abre la puerta a la exploración de otras alternativas para la etapa de DA de manera que esta sea semi-supervisada o débilmente supervisada, manteniendo así la fácil adaptación al dominio de la metodología ADOPS.

6.4.5 *Estudio comparativo con una propuesta similar*

Existen otras aproximaciones cuyo objetivo es la construcción de resúmenes sobre conjuntos de opiniones. En esta sección, comparamos la metodología ADOPS con una aproximación similar presentada en [132].

En [132] los autores realizan una DA y técnicas de SD con el fin de construir reglas que expliquen un conjunto de opiniones. Emplean un modelo ATE (Aspect Term Extraction) supervisado para realizar la DA y emplean el método APriori-SD para la obtención de reglas. Sin embargo, la diversidad del lenguaje les supone un gran problema en su estudio. Esto se ve reflejado en la fina granularidad de los aspectos obtenidos a pesar del empleo de un K-Means para realizar una agrupación de estos. Esta granularidad tan fina hace que el conjunto de aspectos sea más heterogéneo y se reduzca la generalización deseada en un resumen de opiniones. En la tabla 21 se muestran los parámetros empleados en la publicación original.

Parámetros DA en Valdivia et al.	**Valor**
Conv1 feature map	100
Conv1 filter size	2
Conv2 feature map	50
Conv2 filter size	2
Pool size	2
Parámetros Apriori-SD en Valdivia et al.	**Valor**
Min. support	0.001
Min. confidence	0.01

Tabla 21: Configuración de parámetros empleada en [132].

En su estudio, emplean los conjuntos de opiniones de La Alhambra y Sagrada Familia. La dispersión mencionada además del gran desbalanceo de los datos, provoca que las reglas no muestren unas métricas lo suficientemente altas. Además, su aproximación es supervisada, por lo que hace que dependan de la existencia de conjuntos anotados del dominio de estudio. Esto provoca que esta aproximación no sea fácilmente adaptable al dominio.

La tabla 22 muestra una comparación entre algunas de las reglas más significativas obtenidas en [132] y algunas reglas similares obtenidas por la metodología ADOPS. Se muestran las métricas de Soporte, Confianza y WRAcc de cada regla. El símbolo '-' indica que no existe medida de la regla debido a que la aproximación correspondiente no generó dicha regla. Vemos cómo en el conjunto de La Alhambra, se obtienen reglas que cubren aspectos similares. En concreto, obtienen reglas que relacionan los aspectos *guard* y *staff* como aspectos comúnmente negativos. Como la metodología ADOPS hace una detección de aspectos a un nivel de granularidad más abstracto, la categoría *Staff* encapsula ambos aspectos, dando lugar de esta manera a una regla más general. En Sagrada Familia, las reglas obtenidas por la metodología ADOPS y [132] son muy distintas. Sin embargo, los bajos valores de las métricas de las reglas obtenidas por [132] hace que sea difícil extraer conclusiones destacables.

Mostramos por lo tanto cómo la metología ADOPS extrae reglas más generales que facilitan de esta manera la interpretación del resumen al usuario. Además de esta mayor capacidad de síntesis, es destacable que nuestra propuesta sea fácilmente adaptable al dominio deseado, debido a que no depende de la existencia de un conjunto anotado para la etapa de DA y su funcionamiento se reduce a la existencia de un conjunto pequeño de palabras semilla y un conjunto de opiniones sin anotar del dominio deseado.

6.5 CONCLUSIONES

La SO es un campo dentro del ABSA que trata la generación de información sintetizada sobre qué se opina con respecto a los distintos aspectos de una entidad. Sin embargo, los algoritmos o estrategias de SO tienen, de forma general, dos carencias significativas: (1) la explicabilidad e interpretabilidad de los modelos, y (2) la dependencia del dominio de las opiniones (adaptación al dominio). En este capítulo, hemos presentado la metodología ADOPS, una metodología para la SO de manera que damos solución a estos dos desafíos mencionados.

La metodología ADOPS es una metodología modular que parte de un conjunto de opiniones con respecto a una entidad y genera un conjunto de reglas que asocian la presencia de determinados aspectos con la polaridad expresada en las opiniones. Para ello, la metodología ADOPS consta de dos etapas. La primera

La Alhambra						
	Metología ADOPS			Valdivia et al. [132]		
Regla	*Sop*	*Conf*	*WRAcc*	*Sop*	*Conf*	*WRAcc*
{Staff = 1, Ticketing = 1} → {Negative}	0.34	0.74	0.11	-	-	-
{Queues = 1, Ticketing = 1} → {Negative}	0.24	0.58	0.04	-	-	-
{Staff = 1} → {Negative}	0.43	0.7	0.12	<0.01	0.28	<0.01
{Guard = 1} → {Negative}	-	-	-	<0.01	0.38	<0.01
{Queues = 1} → {Negative}	-	-	-	<0.01	0.15	<0.01
Sagrada Familia						
	Metodología ADOPS			Valdivia et al. [132]		
Regla	*Sop*	*Conf*	*WRAcc*	*Sop*	*Conf*	*WRAcc*
{Staff = 1} → {Negative}	0.27	0.68	0.08	-	-	-
{Architecture = 0} → {Negative}	0.24	0.57	0.04	-	-	-
{Location = 0} → {Negative}	0.32	0.51	0.02	-	-	-
{Ceiling = 1} → {Negative}	-	-	-	<0.01	0.1	<0.01
{Natural = 1} → {Negative}	-	-	-	<0.01	0.07	<0.01
{Entry = 1} → {Negative}	-	-	-	<0.01	0.05	0

Tabla 22: Comparación de las reglas y sus métricas extraídas por NMEEF-SD in nuestro estudio y en [132] empleando APriori-SD.

estapa consiste en la detección de las categorías de aspectos presentes en cada opinión mediante un modelo DL débilmente supervisado ($ABAE_{AS}$) que solo requiere un conjunto *In-Domain* sin etiquetar y una serie de palabras semilla que constituirían la parte supervisada del modelo. La segunda etapa genera, a partir de una representación como *itemsets* de las opiniones, una serie de reglas mediante técnicas de SD que serían el resumen generado.

Como se ha expuesto a lo largo del capítulo, la metodología ADOPS genera resúmenes formados por reglas, lo que facilita la interpretabilidad por parte del usuario de los resúmenes obtenidos. Además, las reglas generadas son acompañadas por una serie de métricas que cuantifican el interés estadístico de estas y hacen que los resúmenes sean explicables. Por otro lado, nuestra propuesta de emplear un método debilmente supervisado para la DA, hace que nuestro enfoque tenga independencia del dominio debido a que no requiere opiniones etiquetadas *In-domain* para esta etapa de la metodología ADOPS.

En nuestro estudio, la etapa de DA se realiza a nivel de categoría de aspecto. Esto hace que, a diferencia de si se hiciera a nivel de término, obtengamos re-

súmenes más generales y significativos. Para confirmar este hecho, realizamos una comparación con una propuesta similar [132] en la que la etapa de detección de aspectos se realiza con un modelo supervisado a nivel de término y un posterior *clustering* para la agrupación de estos aspectos en categorías. En esta comparación, vemos que las reglas generadas por la metodología ADOPS son reglas más contundentes en cuanto a las métricas obtenidas, facilitando de esta manera la extracción de conclusiones. Además, nuestra propuesta, a diferencia de la realizada en [132], no depende de la existencia de datos anotados para la DA.

Nuestra contribución ADOPS fue presentada en el artículo *ADOPS: Aspect Discovery Opinion Summarisation methodology based on Deep Learning and Subgroup Discovery for generating explainable opinion summaries* [89], publicado en la revista *Knowledge Based Systems* con índice de impacto de 8.038 indexada en la categoría *Computer Science, Artificial Intelligence*.

7

CONCLUSIONES Y TRABAJO FUTURO

Las contribuciones de E^{2}SAM, ORCo y ADOPS suponen la consecución del objetivo principal marcado al inicio de esta tesis así como de los distintos subobjetivos. En este capítulo exponemos las conclusiones obtenidas a partir del desarrollo de esta tesis. Además, las distintas contribuciones realizadas no solo dan lugar a conclusiones y resultados, sino que abren nuevas vías de estudio que serán detalladas en este capítulo.

Una de las formas más comunes de expresión mediante lenguaje en la Web es mediante opiniones, las cuales muestran la valoración subjetiva por parte de sujetos con respecto a entidades y sus distintos aspectos o características. La agregación de la subjetividad expresada en un conjunto de opiniones es útil para cualquier sujeto interesado en estas entidades para la toma de decisiones en cuanto a, por ejemplo en el caso de comercio electrónico, comprar un producto por parte de un cliente o cambiar ciertas características del mismo por parte del vendedor. El campo de la síntesis de opiniones tiene como objetivo el proporcionar una síntesis sobre qué se opina con respecto a una entidad. Este campo del Procesamiento del Lenguaje Natural puede encapsular distintas tareas, como son la detección de aspectos, la clasificación de la opinión o la generación de texto.

Como ya se ha destacado a lo largo de esta tesis, la síntesis de opiniones, y la IA en general, requiere de nuevos enfoques y desafíos que la hagan más accesible y democrática, así como de desarrollar enfoques más interpretables o explicables con el fin de apoyar de una forma justificada la toma de decisiones a partir del conocimiento aportado por los sistemas. En esta tesis nos planteamos el objetivo principal de contribuir al campo de la síntesis de opiniones y, además, hacerlo con contribuciones que se adapten a esta necesidad de crear métodos de IA más democráticos y transparentes. Por ello, en esta tesis hemos enfocado nuestras contribuciones hacia (1) contribuciones para la adaptación al dominio de otros recursos (2) contribuciones fácilmente adaptables a un dominio y (3) contribuciones interpretables y explicables. Siguiendo estas características, a lo largo de esta tesis se han presenatdo las contribuciones de E^{2}SAM, ORCo y ADOPS.

En este capítulo, mostraremos en primer lugar en la sección 7.1 las principales conclusiones extraídas a lo lago del desarrollo de las distintas contribuciones de esta tesis. Posteriormente, en la sección 7.2 enumeraremos posibles trabajos futuros que surgen a partir de nuestras contribuciones.

7.1 CONCLUSIONES

A partir del desarrollo de esta tesis y las contribuciones realizadas, se han obtenido las siguientes conclusiones:

- Con E^{2}SAM nos enfrentamos a una de las potenciales subtareas dentro de la SO como es la clasificación de la opinión. E^{2}SAM es una metodología que parte de un conjunto base de modelos preentrenados para la clasificación de la opinión o MCOs y los combina con el fin de dar una respuesta agregada. En la experimentación, mostrada en el capítulo 4, vimos como E^{2}SAM confirmaba que realiza una distribución de la contribución de cada MCO asociada a la bondad de cada uno de ellos en el dominio de interés. Analizando los resultados vimos como E^{2}SAM mejoraba en la mayor parte de conjuntos de

texto el rendimiento de dos operadores de combinación de la literatura, así como del mejor MCO en cada conjunto de opiniones. De esta manera, E^{2}SAM se confirmó como una contribución que realiza una adaptación al dominio de recursos preentrenados mejorando el rendimiento de estos.

- Como ya se comentó en la introducción de esta tesis, realizada en el capítulo 1, existen pocos recursos para la evaluación de sistemas o metodologías para la síntesis de opiniones. Para poder simular un entorno real, es necesario tener conjuntos de opiniones agrupados por entidad, permitiendo así mostrar conocimiento significativo y valioso con un sistema de síntesis de opiniones. Con nuestra contribución ORCo, presentada en el capítulo 5, proporcionamos un conjunto de opiniones con múltiples categorías de aspectos y polaridades anotadas a nivel de oración, siendo además ORCo un conjunto de opiniones en el dominio de *Restaurantes* de un solo restaurante. De esta manera, ORCo da la posibilidad de evaluar distintas tareas asociadas a la síntesis de opiniones y la obtención de resúmenes significativos. Además, se evaluó la calidad de la anotación del conjunto de datos, llegando a niveles de acuerdo bastante altos en la anotación de tanto las categorías de aspecto como de polaridad.
- Finalmente, con nuestra contribución de ADOPS descrita en el capítulo 6, cumplimos el principal objetivo de esta tesis: desarrollar una metodología para la obtención de resúmenes a partir de opiniones, siendo estos resúmenes interpretables y explicables. ADOPS devuelve un conjunto de reglas, cuya interpretación es sencilla y, además, estos resúmenes son explicados por medio de métricas que cuantifican el interés estadístico de estas reglas. Además, a pesar de que ADOPS se presenta como una metodología modular, en nuestra experimentación empleamos y sugerimos técnicas débilmente supervisadas. Esta independencia con respecto a la necesidad de conjuntos de opiniones anotados, hacen que el desarrollo de un sistema para la SO mediante la metodología ADOPS sea un sistema fácilmente adaptable a distintos dominios.

Con las tres contribuciones realizadas en esta tesis cumplimos los tres subobjetivos marcados al inicio y el objetivo general de contribuir al campo de la SO de manera que seamos capaces de generar resúmenes de opiniones estructurados, interpretables, explicables y mediante una metodología flexible al dominio. Además, las tres contribuciones realizadas están avaladas por publicaciones en revistas científicas de impacto, siendo E^{2}SAM una contribución propuesta en [88] y ORCo y ADOPS propuestas en [89].

7.2 TRABAJOS FUTUROS

Además de cumplir los objetivos propuestos al inicio de esta tesis, el desarrollo de la misma abre puertas a otros desafíos que pueden mejorar nuestras contribuciones, así como el campo de la síntesis de opiniones en general.

- Asociados a nuestra contribución E^2SAM, surgen posibles trabajos futuros como pueden ser el empleo de técnicas de balanceo de datos para mejorar el rendimiento de E^2SAM en conjuntos desbalanceados. También surge como posible trabajo futuro el emplear enfoques multi-objetivo con el fin de optimizar múltiples métricas.
- Con nuestra contribución de ORCo surgen posibles mejoras al propio ORCo como puede ser el ampliar el número de opiniones anotadas o el anotar los aspectos a nivel de término. También se plantea el crear nuevos recursos para la evaluación de sistemas de síntesis de opiniones en otros dominios distintos al de *Restaurante*.
- Durante la experimentación realizada con ADOPS, vimos cómo la calidad de las reglas obtenidas se vinculaba directamente al rendimiento del modelo empleado para la detección de aspectos. De esta forma, una detección de aspectos poco efectiva puede llevar a la obtención de reglas engañosas. Por lo tanto, uno de los trabajos futuros más desafiantes es el desarrollar un sistema para la detección de aspectos que sea débilmente supervisado, mejore los existentes en la literatura y, además, incorpore la posibilidad de realizar una detección de aspectos multi-aspecto a nivel de oración, característica que ningún método semi-supervisado o débilmente supervisado incorpora en la literatura.
- Explorar otros niveles de granularidad en los resúmenes, relacionando, por ejemplo, la mención de determinados términos con la mención de ciertas categorías de aspectos en opiniones con una polaridad determinada de interés. Esto daría lugar a, por ejemplo en el dominio de restaurantes, obtener reglas que asociaran la mención de una comida concreta como puede ser *pizza* a la mención de la categoría *Comida* en una oración con polaridad negativa.
- Crear un sistema *End-to-End* que encapsule todas las subtareas posibles de la síntesis de opiniones y la generación de resúmenes estructurados.

BIBLIOGRAFÍA

[1] Ahmed Abbasi, Hsinchun Chen, and Arab Salem. Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. *ACM Trans. Inf. Syst.*, 26(3):12:1–12:34, 2008.

[2] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. *SIGMOD Rec.*, 22(2):207–216, 1993.

[3] Reinald Kim Amplayo and Mirella Lapata. Unsupervised opinion summarization with noising and denoising. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1934–1945. Association for Computational Linguistics, 2020.

[4] Stefanos Angelidis and Mirella Lapata. Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3675–3686, 2018.

[5] Stefanos Angelidis, Reinald Kim Amplayo, Yoshihiko Suhara, Xiaolan Wang, and Mirella Lapata. Extractive Opinion Summarization in Quantized Transformer Spaces. *Transactions of the Association for Computational Linguistics*, 9:277–293, 2021.

[6] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82 – 115, 2020.

[7] Mattia Atzeni, Amna Dridi, and Diego Reforgiato Recupero. Using frame-based resources for sentiment analysis within the financial domain. *Progress in Artificial Intelligence*, 7(4):273–294, 2018.

[8] Martin Atzmueller and Frank Puppe. Sd-map – a fast algorithm for exhaustive subgroup discovery. In *Knowledge Discovery in Databases: PKDD 2006*, pages 6–17, 2006.

[9] Anthony Aue and Michael Gamon. Customizing Sentiment Classifiers to New Domains: a Case Study. In *Submitted to RANLP-05, the International Conference on Recent Advances in Natural Language Processing*, 2005.

[10] E. Bauer and Ron Kohavi. An empirical comparison of voting classification algorithms : Bagging, boosting, and variants. *Machine Learning*, 36:1–38, 1996.

[11] Stephen D. Bay and Michael J. Pazzani. Detecting Group Differences: Mining Contrast Sets. *Data Mining and Knowledge Discovery*, 5(3):213–246, 2001.

[12] Philip Beineke, Trevor Hastie, Christopher Manning, and Shivakumar Vaithyanathan. An exploration of sentiment summarization. In *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*, 2003.

[13] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1):151–175, 2010.

[14] John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447. Association for Computational Linguistics, 2007.

[15] Christian Blum and Andrea Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Comput. Surv.*, 35(3): 268–308, 2003.

[16] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[17] Arthur Bražinskas, Mirella Lapata, and Ivan Titov. Unsupervised opinion summarization as copycat-review generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5151–5169. Association for Computational Linguistics, 2020.

[18] Arthur Bražinskas, Mirella Lapata, and Ivan Titov. Learning opinion summarizers by selecting informative reviews. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9424–9442. Association for Computational Linguistics, 2021.

[19] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[20] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[21] J. Brest, M. S. Maučec, and B. Bošković. Single objective real-parameter optimization: Algorithm jso. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, pages 1311–1318, 2017.

[22] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[23] Hongjie Cai, Yaofeng Tu, Xiangsheng Zhou, Jianfei Yu, and Rui Xia. Aspect-category based sentiment analysis with hierarchical graph convolutional network. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 833–843. International Committee on Computational Linguistics, 2020.

[24] Erik Cambria, Daniel Olsher, and Dheeraj Rajagopal. Senticnet 3: A common and common-sense knowledge base for cognition-driven sentiment analysis. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, pages 1515–1521. AAAI Press, 2014.

[25] Nofar Carmeli, Xiaolan Wang, Yoshihiko Suhara, Stefanos Angelidis, Yuliang Li, Jinfeng Li, and Wang-Chiew Tan. ExplainIt: Explainable review summarization with opinion causality graphs. abs/2006.00119, 2020.

[26] C. J. Carmona, M. J. del Jesus, and F. Herrera. A unifying analysis for the supervised descriptive rule discovery via the weighted relative accuracy. *Knowledge-Based Systems*, 139:89 – 100, 2018.

[27] Cristóbal J. Carmona, Pedro González, María José Del Jesus, and Francisco Herrera. Nmeef-sd: Non-dominated multiobjective evolutionary algorithm for extracting fuzzy rules in subgroup discovery. *IEEE Transactions on Fuzzy Systems*, 18:958 – 970, 2010.

[28] Zhuang Chen and Tieyun Qian. Bridge-based active domain adaptation for aspect term extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 317–327. Association for Computational Linguistics, 2021.

[29] Amitava Das and Sivaji Bandyopadhyay. Subjectivity detection using genetic algorithm. In *Proceedings of the 1st Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, pages 14–21, 2010.

[30] Swagatam Das and Ponnuthurai Nagaratnam Suganthan. Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, 15(1):4–31, 2011.

[31] Mark Davies and Joseph L Fleiss. Measuring agreement for multinomial data. *Biometrics*, pages 1047–1051, 1982.

[32] Sophie de Kok, Linda Punt, Rosita van den Puttelaar, Karoliina Ranta, Kim Schouten, and Flavius Frasincar. Review-aggregated aspect-based sentiment analysis with ontology features. *Progress in Artificial Intelligence*, 7(4): 295–306, 2018.

[33] M. J. del Jesus, P. Gonzalez, and F. Herrera. Multiobjective genetic algorithm for extracting subgroup discovery fuzzy rules. In *2007 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making*, pages 50–57, 2007.

[34] Ayça Deniz, Merih Angin, and Pelin Angin. Evolutionary Multiobjective Feature Selection for Sentiment Analysis. *IEEE Access*, 9:142982–142996, 2021.

[35] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.

[36] Shizhe Diao, Ruijia Xu, Hongjin Su, Yilei Jiang, Yan Song, and Tong Zhang. Taming pre-trained language models with n-gram representations for low-resource domain adaptation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3336–3349. Association for Computational Linguistics, 2021.

[37] Guozhu Dong and James Bailey. *Contrast Data Mining: Concepts, Algorithms, and Applications*. Chapman &; Hall/CRC, 1st edition, 2012.

[38] Guozhu Dong and Jinyan Li. Efficient mining of emerging patterns: Discovering trends and differences. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 43–52, 1999.

[39] M. Dorigo and G. Di Caro. Ant colony optimization: a new meta-heuristic. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99*, volume 2, pages 1470–1477 Vol. 2, 1999.

[40] Hady Elsahar, Maximin Coavoux, Jos Rozen, and Matthias Gallé. Self-supervised and controlled multi-document opinion summarization. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1646–1662. Association for Computational Linguistics, 2021.

[41] Larry J. Eshelman and J. David Schaffer. Real-coded genetic algorithms and interval-schemata. In L. DARRELL WHITLEY, editor, *Foundations of Genetic Algorithms*, volume 2 of *Foundations of Genetic Algorithms*, pages 187 – 202. Elsevier, 1993.

[42] Hongjian Fan and Kotagiri Ramamohanarao. A bayesian approach to use emerging patterns for classification. In *Proceedings of the 14th Australasian database conference-Volume 17*, pages 39–48. Australian Computer Society, Inc., 2003.

[43] Christiane Fellbaum, editor. *WordNet: an electronic lexical database*. MIT Press, 1998.

[44] J. R. Firth. A synopsis of linguistic theory 1930-55. In *Studies in Linguistic Analysis (special volume of the Philological Society)*, volume 1952-59, pages 1–32. The Philological Society, 1957.

[45] PA Flach and D Gamberger. Subgroup evaluation and decision support for a direct mailing marketing problem. In *Proceedings of the 12th European conference on machine learning and 5th European conference on principles and practice of knowledge discovery in databases*, pages 45 – 56, 2001.

[46] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*, ICML'96, page 148–156. Morgan Kaufmann Publishers Inc., 1996.

[47] Dragan Gamberger and Nada Lavrac. Expert-guided subgroup discovery: Methodology and application. *Journal of Artificial Intelligence Research - JAIR*, 17, 2011.

[48] Gayatree Ganu, Noémie Elhadad, and Amélie Marian. Beyond the stars: Improving rating predictions using review text content. In *WebDB*, 2009.

[49] Klaifer Garcia and Lilian Berton. Topic detection and sentiment analysis in Twitter content related to COVID-19 from Brazil and the USA. *Applied Soft Computing*, 101:107057, 2021.

[50] Salvador García, Daniel Molina, Manuel Lozano, and Francisco Herrera. A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the cec'2005 special session on real parameter optimization. *Journal of Heuristics*, 15(6):617, 2008.

[51] Athanasios Giannakopoulos, Claudiu Musat, Andreea Hossmann, and Michael Baeriswyl. Unsupervised aspect term extraction with B-LSTM & CRF using automatically labelled datasets. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 180–188. Association for Computational Linguistics, 2017.

[52] Fred Glover. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13(5):533–549, 1986.

[53] David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.

[54] Zellig Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

[55] Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. An unsupervised neural attention model for aspect extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 388–397, 2017.

[56] Francisco Herrera, Cristóbal J. Carmona, Pedro González, and María José Del Jesus. An overview on subgroup discovery: Foundations and applications. *Knowledge and Information Systems*, 29:495–525, 2011.

[57] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[58] Minqing Hu and Bing Liu. Mining opinion features in customer reviews. In *Proceedings of the 19th National Conference on Artifical Intelligence*, AAAI'04, page 755–760. AAAI Press, 2004.

[59] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177, 2004.

[60] Jiaxin Huang, Yu Meng, Fang Guo, Heng Ji, and Jiawei Han. Weakly-supervised aspect-based sentiment analysis via joint aspect-sentiment topic embedding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6989–6999. Association for Computational Linguistics, 2020.

[61] Clayton J Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth International Conference on Weblogs and Social Media (ICWSM-14)*, 2014.

[62] Nguyen Huy Tien, Le Tung Thanh, and Nguyen Minh Le. Opinions Summarization: Aspect Similarity Recognition Relaxes The Constraint of Predefined Aspects. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 487–496, 2019.

[63] Jinbae Im, Moonki Kim, Hoyeop Lee, Hyunsouk Cho, and Sehee Chung. Self-supervised multimodal opinion summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 388–403. Association for Computational Linguistics, 2021.

[64] Jagadeesh Jagarlamudi, Hal Daumé III, and Raghavendra Udupa. Incorporating lexical priors into topic models. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 204–213, 2012.

[65] Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.

[66] Martin Joos. Description of language design. *Journal of the Acoustical Society of America*, 22:701–707, 1950.

[67] Dan Jurafsky and James H. Martin. *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition*. Pearson Prentice Hall, 2009.

[68] Eunike Andriani Kardinata, Nur Aini Rakhmawati, and Nurrida Aini Zuhroh. Ontology-based sentiment analysis on news title. In *2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCIT)*, pages 360–364. IEEE, 2021.

[69] Branko Kavšek and Nada Lavrač. Apriori-sd: Adapting association rule learning to subgroup discovery. *Applied Artificial Intelligence*, 20(7):543–583, 2006.

[70] Jacqueline Kazmaier and Jan H. van Vuuren. The power of ensemble learning in sentiment analysis. *Expert Systems with Applications*, 187:115819, 2022.

[71] Alistair Kennedy and Diana Inkpen. Sentiment classification of movie reviews using contextual valence shifters. *Computational Intelligence*, 22(2): 110–125, 2006.

[72] Talaat Khalil and Samhaa R. El-Beltagy. NileTMRG at SemEval-2016 task 5: Deep convolutional neural networks for aspect category and sentiment extraction. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 271–276. Association for Computational Linguistics, 2016.

[73] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016.

[74] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[75] Willi Klösgen. Explora: A multipattern and multistrategy discovery assistant. In *Advances in knowledge discovery and data mining*, pages 249–271. American Association for Artificial Intelligence, 1996.

[76] K. Krippendorff. *Content Analysis: An Introduction to Its Methodology*. Content Analysis: An Introduction to Its Methodology. Sage, 2004.

[77] Lun-Wei Ku, Yu-Ting Liang, and Hsin-Hsi Chen. Opinion extraction, summarization and tracking in news and blog corpora. In *Proceedings of AAAI-CAAW-06, the Spring Symposia on Computational Approaches to Analyzing Weblogs*, 2006.

[78] Avinash Kumar, Pranjal Gupta, Raghunathan Balan, Lalita Neti, and Aruna Malapati. Bert based semi-supervised hybrid approach for aspect and sentiment classification. *Neural Processing Letters*, 53, 2021.

[79] J. Richard Landis and Gary G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1), 1977.

[80] Nada Lavrač, Branko Kavšek, Peter Flach, and Ljupčo Todorovski. Subgroup discovery with cn2-sd. *Journal of Machine Learning Research*, 5:153–188, 2004.

[81] Bing Liu. *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*. Cambridge University Press, 2015.

[82] Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences. In *International Conference on Learning Representations ICLR*, 2018.

[83] Sisi Liu, Kyungmi Lee, and Ickjai Lee. Document-level multi-topic sentiment classification of Email data with BiLSTM and data augmentation. *Knowledge-Based Systems*, 197:105918, 2020.

[84] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[85] C. Lonjarret, C. Robardet, M. Plantevit, R. Auburtin, and M. Atzmueller. Why should i trust this item? explaining the recommendations of any model. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 526–535, 2020.

[86] Yue Lu, Cheng Xiang Zhai, and Neel Sundaresan. Rated aspect summarization of short comments. In *WWW'09 - Proceedings of the 18th International World Wide Web Conference*, WWW'09 - Proceedings of the 18th International World Wide Web Conference, pages 131–140, 2009.

[87] H. P. Luhn. A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development*, 1(4):309–317, 1957.

[88] Miguel López, Ana Valdivia, Eugenio Martínez-Cámara, Maria Victoria Luzon, and Francisco Herrera. E2sam: Evolutionary ensemble of sentiment analysis methods for domain adaptation. *Information Sciences*, 480:273–286, 2019.

[89] Miguel López, Eugenio Martínez-Cámara, M. Victoria Luzón, and Francisco Herrera. Adops: Aspect discovery opinion summarisation methodology based on deep learning and subgroup discovery for generating explainable opinion summaries. *Knowledge-Based Systems*, 231:107455, 2021.

[90] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297, 1967.

[91] R. Mallipeddi, P. N. Suganthan, Q. K. Pan, and M. F. Tasgetiren. Differential evolution algorithm with ensemble of parameters and mutation strategies. *Applied Soft Computing*, 11(2):1679–1696, 2011.

[92] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, 2014.

[93] Eugenio Martínez-Cámara, Yoan Gutiérrez-Vázquez, Javi Fernández, Arturo Montejo-Ráez, and Rafael Muñoz Guillena. Ensemble classifier for twitter sentiment analysis. In *Proceedings of the Workshop on NLP Applications: Completing the Puzzle co-located with the 20th International Conference on Applications of Natural Language to Information Systems (NLDB 2015)*, pages 1–10, 2015.

[94] María-Teresa Martín-Valdivia, Eugenio Martínez-Cámara, Jose-M. Perea-Ortega, and L. Alfonso Ureña-López. Sentiment polarity detection in Spanish reviews combining supervised and unsupervised approaches. *Expert Systems with Applications*, 40(10):3934–3942, 2013.

[95] Eugenio Martínez-Cámara, Maria Martín-Valdivia, M. Dolores González, and José Perea-Ortega. Integrating spanish lexical resources by meta-classifiers for polarity classification. *Journal of Information Science*, 40:539–554, 2014.

[96] Quinn McNemar. Optnote on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

[97] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013.

[98] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 3111–3119. Curran Associates Inc., 2013.

[99] Melanie Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, 1998.

[100] Daniela Moctezuma, Sabino Graff, Mario amd Miranda-Jiménez, Eric S. Tellez, Abel Coronado, Claudia N. Sánchez, and José Ortiz-Bejar. A genetic programming approach to sentiment analysis for twitter: TASS'17. In *Proceedings of TASS 2017: Workshop on Sentiment Analysis at SEPLN co-located with 33rd SEPLN Conference (SEPLN 2017)*, pages 23–28, 2017.

[101] Daniel Molina, Javier Poyatos, Javier Del Ser, Salvador García, Amir Hussain, and Francisco Herrera. Comprehensive Taxonomies of Nature- and Bio-inspired Optimization: Inspiration Versus Algorithmic Behavior, Critical Analysis Recommendations. *Cognitive Computation*, 12(5):897–939, 2020.

[102] Pablo Moscato. On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts - Towards Memetic Algorithms, 1989.

[103] Ruba Obiedat, Osama Harfoushi, Raneem Qaddoura, Laila Al-Qaisi, and Ala' M. Al-Zoubi. An Evolutionary-Based Sentiment Analysis Approach for Enhancing Government Decisions during COVID-19 Pandemic: The Case of Jordan. *Applied Sciences*, 11(19):9080, 2021.

[104] Aytuğ Onan and Serdar Korukoğlu. A feature selection model based on genetic rank aggregation for text sentiment classification. *Journal of Information Science*, 43(1):25–38, 2017.

[105] D. Opitz and R. Maclin. Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11:169–198, 1999.

[106] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundation and Trends in Information Retrieval*, 2(1–2):1–135, 2008.

[107] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, pages 79–86. Association for Computational Linguistics, 2002.

[108] Silviu Paun, Ron Artstein, and Massimo Poesio. Statistical methods for annotation analysis. *Synthesis Lectures on Human Language Technologies*, 15 (1):1–217, 2022.

[109] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.

[110] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics, 2018.

[111] Mohammad Taher Pilehvar and Jose Camacho-Collados. *Embeddings in Natural Language Processing: Theory and Advances in Vector Representations of Meaning*. 2020.

[112] Flor Miriam Plaza del Arco, M. Teresa Martín Valdivia, Salud María Jiménez Zafra, M. Dolores Molina González, and Eugenio Martínez Cámara. COPOS: Corpus of patient opinions in spanish. application of sentiment analysis techniques. *Procesamiento del Lenguaje Natural*, 57:83–90, 2016.

[113] Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, 2014.

[114] Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. SemEval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 486–495. Association for Computational Linguistics, 2015.

[115] Soujanya Poria, Erik Cambria, and Alexander Gelbukh. Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems*, 108:42–49, 2016.

[116] Vineeth Rakesh, Weicong Ding, Aman Ahuja, Nikhil Rao, Yifan Sun, and Chandan K. Reddy. A sparse topic model for extracting aspect-specific summaries from online reviews. In *Proceedings of the 2018 World Wide Web Conference*, page 1573–1582, 2018.

[117] Filipe N Ribeiro, Matheus Araújo, Pollyanna Gonçalves, Marcos André Gonçalves, and Fabrício Benevenuto. Sentibench - a benchmark comparison of state-of-the-practice sentiment analysis methods. *EPJ Data Science*, 5(1): 1–29, 2016.

[118] Nuria Rodríguez-Barroso, Antonio R. Moya, José A. Fernández, Elena Romero, Eugenio Martínez-Cámara, and Francisco Herrera. Deep Learning Hyper-parameter Tuning for Sentiment Analysis in Twitter based on Evolutionary Algorithms. In *2019 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 255–264, 2019.

[119] Lior Rokach. Ensemble methods for classifiers. In *Data Mining and Knowledge Discovery Handbook*, pages 957–980. Springer US, 2005.

[120] Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389. Association for Computational Linguistics, 2015.

[121] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall Press, 3rd edition, 2009.

[122] Abigail See, Peter Liu, and Christopher Manning. Get to the point: Summarization with pointer-generator networks. In *Association for Computational Linguistics*, 2017.

[123] Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, Christopher Potts, et al. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, pages 1631–1642, 2013.

[124] Rainer Storn and Kenneth Price. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11:341–359, 1997.

[125] Chi Sun, Luyao Huang, and Xipeng Qiu. Utilizing BERT for Aspect-Based Sentiment Analysis via Constructing Auxiliary Sentence. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 380–385. Association for Computational Linguistics, 2019.

[126] El-Ghazali Talbi. *Metaheuristics - From Design to Implementation.* Wiley, 2009.

[127] R. Tanabe and A. S. Fukunaga. Improving the search performance of shade using linear population size reduction. In *2014 IEEE Congress on Evolutionary Computation (CEC)*, pages 1658–1665, 2014.

[128] Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. Sentiment strength detection for the social web. *J. Am. Soc. Inf. Sci. Technol.*, 63(1): 163–173, 2012.

[129] Peter D. Turney. 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*, ACL '02, pages 417–424, 2002.

[130] Lan Umek and Blaz Zupan. Subgroup discovery in data sets with multi-dimensional responses. *Intell. Data Anal.*, 15:533–549, 2011.

[131] Ana Valdivia, M. Victoria Luzón, Erik Cambria, and Francisco Herrera. Consensus vote models for detecting and filtering neutrality in sentiment analysis. *Information Fusion*, 44:126 – 135, 2018.

[132] Ana Valdivia, Eugenio Martínez-Cámara, Iti Chaturvedi, M. Victoria Luzón, Erik Cambria, Yew-Soon Ong, and Francisco Herrera. What do people think about this monument? understanding negative reviews via deep

learning, clustering and descriptive rules. *Journal of Ambient Intelligence and Humanized Computing*, 11(1):39–52, 2020.

[133] L. G. Valiant. A theory of the learnable. In *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*, STOC '84, page 436–445. Association for Computing Machinery, 1984.

[134] David Valle-Cruz, Vanessa Fernandez-Cortez, Asdrúbal López-Chau, and Rodrigo Sandoval-Almazán. Does Twitter Affect Stock Market Decisions? Financial Sentiment Analysis During Pandemics: A Comparative Study of the H1N1 and the COVID-19 Periods. *Cognitive Computation*, 2021.

[135] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[136] Sebastián Ventura and José María Luna. *Subgroup Discovery*, pages 71–98. Springer International Publishing, 2018.

[137] Hai Wan, Yufei Yang, Jianfeng Du, Yanan Liu, Kunxun Qi, and Jeff Z. Pan. Target-Aspect-Sentiment Joint Detection for Aspect-Based Sentiment Analysis. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05): 9122–9129, 2020.

[138] Xiaolan Wang, Yoshihiko Suhara, Natalie Nuno, Yuliang Li, Jinfeng Li, Nofar Carmeli, Stefanos Angelidis, Eser Kandogann, and Wang-Chiew Tan. Extremereader: An interactive explorer for customizable and explainable review summarization. In *Companion Proceedings of the Web Conference 2020*, page 176–180, 2020.

[139] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

[140] Rachel Wities, Vered Shwartz, Gabriel Stanovsky, Meni Adler, Ori Shapira, Shyam Upadhyay, Dan Roth, Eugenio Martinez Camara, Iryna Gurevych, and Ido Dagan. A consolidated open knowledge representation for multiple texts. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, pages 12–24, 2017.

[141] David H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.

[142] Dionysios Xenos, Panagiotis Theodorakakos, John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. AUEB-ABSA at SemEval-2016 task 5: Ensembles of classifiers and embeddings for aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic*

Evaluation (SemEval-2016), pages 312–317. Association for Computational Linguistics, 2016.

[143] N. Xiong, Daniel Molina, Miguel Leon, and Francisco Herrera. A Walk into Metaheuristics for Engineering Optimization: Principles, Methods and Recent Trends. *International Journal of Computational Intelligence Systems*, 8: 606–636, 2015.

[144] Cha Zhang and Yunqian Ma. *Ensemble Machine Learning: Methods and Applications*. Springer Publishing Company, Incorporated, 2012.

[145] Albrecht Zimmermann and Luc De Raedt. Cluster-grouping: From subgroup discovery to clustering. *Machine Learning*, 77:125–159, 2009.

[146] Felipe Zschornack Rodrigues Saraiva, Ticiana Linhares Coelho da Silva, and José Antônio Fernandes de Macêdo. Aspect Term Extraction Using Deep Learning Model with Minimal Feature Engineering. In *Advanced Information Systems Engineering*, pages 185–198. Springer International Publishing, 2020.

UNIVERSIDAD
DE GRANADA