UNIVERSIDAD
DE GRANADA

Department of Computer Science and Artificial Intelligence

Interpretable and Effortless Techniques for Social Network Analysis

PhD Dissertation submitted by

Manuel Francisco Aparicio

To obtain the international PhD degree as part of the

Doctoral Programme in Information and Communication Technologies

Under the supervision of

Juan Luis Castro Peña
Department of Computer Science and Artificial Intelligence
University of Granada

November 15, 2022

Editor: Universidad de Granada. Tesis Doctorales
Autor: Manuel Francisco Aparicio
ISBN: 978-84-1117-628-6
URI: https://hdl.handle.net/10481/79151

Manuel Francisco Aparicio
Interpretable and Effortless Techniques for Social Network Analysis
PhD. Dissertation, November 15, 2022
Supervisor: Juan Luis Castro Peña

University of Granada
CITIC-UGR
E.T.S. Ingeniería Informática y Telecomunicaciones
Department of Computer Science and Artificial Intelligence
Rafael Gómez Olmedo 2
18071 Granada

This manuscript was written in LaTeX using Clean Thesis template by Ricardo Langner.

Agradecimientos

> *La tierra prometida pero, ¿para quién?*
> *Un rincón del paraíso pero, ¿con quién?*
>
> **— Javier Ibarra**

No estarían ustedes leyendo este documento si no fuese por **Andrés** y **María Jesús**, mi padre y mi madre, quienes no sólo me han criado sino que han hecho todo lo posible porque nosotros, sus hijos, demos lo máximo a cada paso. Gracias por cuidarnos y animarnos siempre. Os quiero. Gracias a mi hermana **Paqui**, por descubrirnos a todos el esfuerzo del camino universitario; a mi hermana **María**, por su calidez y cariño, y por cuidar tan bien de mi media naranja; a mi hermano **Andrés**, compañero de aficiones y de profesión, y víctima de mis innumerables consultas. A mi abuela **Paqui**, la persona más abierta e inteligente que conozco, que también me ha criado y me ha enseñado que la valentía y el coraje de este mundo es cosa de mujeres. Gracias por enseñarme a regar mis flores. A mis sobrinos, **Gonzalo**, **Andrés**, **Irene** y **Olivia**. Y, por supuesto, a **Limón**, compañero de infinitos paseos e inestimable oyente de mis conversaciones en voz alta. Al resto de mi familia, gracias.

Disculpen mi osadía al no haber empezado por aquí, pero la familia siempre es lo primero. A **Juan Luis**, por su contínua dedicación a mi tesis, por su guía y sus incontables ideas. Gracias por tu esfuerzo y paciencia. A **Encarni**, fuente incansable de energía y entusiasmo, e inestimable ayuda. A **Miguel Ángel**, **Javier**, **Óscar**, **Pascual**, **Azzam**, **M. Ángeles**, **Raquel** y **Fernando**, con quienes he tenido el placer de trabajar de cerca durante estos años. Al resto de miembros del proyecto Nutcracker, tanto por vuestra ayuda como por el gran clima que habéis creado.

A mi predecesor **Álex**, quien ha hecho de mi estancia en Italia una de las experiencias más bellas de mi doctorado. Desde el primer día, has dado lo máximo para enseñarme cuanto ha sido posible. No te haces una idea de lo que te lo agradezco. Grazie mille a **Fabrizio**, per avermi accolto nel suo gruppo e avermi fatto imparare da loro. Grazie al resto dei miei colleghi a l'Italia.

Y, como soy fiel amante de las simetrías, qué mejor que acabar con mis amigos, que son también familia. Cuatro personas me han acompañado cada segundo de ésta tesis: **Salva**, no puedo sino agradecerte que hayas sido mi compañero de alegrías y penas, de trabajo y de alguna que otra carrera de karts; **Fran**, fiel amante de la pachamama, eres una mina de oro. Yo no creía en los dioses hasta que te conocí; **Joaquín**, mi padre en Granada, a quien ni los jienenses le ganan en nobleza ni voluntad; y **Alba**, mi faro particular, no sé cómo te las apañas para ayudarme a crecer tanto y estar siempre a mi lado, aunque vivas tan lejos. Gracias a los cuatro, no habría sido lo mismo sin vosotros. Tampoco pueden faltar los agradecimientos a **Javi** que, con alguna pausa entre capítulos, lleva conmigo toda la vida (y todavía tiene cosas que enseñarme). Y a **Sergio**, nuestro farolillo de Alejandría, que ilumina el camino de la oscuridad de nuestras almas. A **Patri**, mi cocinera vegana y fruto de muchas risas, que ha tenido que aguantarnos tanto a Salva como a mí. A mis otros dos compañeros toscanos, **Jose** (mi musa metafísica) y **Julio** (a quien hay que dejar su espacio, pero que me ha enseñado todo lo que sé de italiano). A **Juanillo**, nuestro cartero de confianza, cuya alegría contagiosa termina convirtiéndose en adicción. A **Isa** (¡y a **Mel**!), con quien he compartido escondite, música, cojera, basura espacial y una cantidad obscena de horas de reflexión. A la **Espe 93½**, aunque a medias, porque la mitad del tiempo ha estado de parranda. A **Fernando**, un sabio en el noble arte de vivir. Al resto de personas que han caminado conmigo en los últimos cuatro años, y en especial a **Ofelia**, **Ramón**, **Nuria**, **el otro Jose**, **Pepe**, **María**, **la otra María**, **Irene**, **Hugo**, **el otro Sergio**, **Pedro**, **Majo** y **Nelson**, compañeros de vacaciones e inquietudes (ENPF).

A **Granada**, que, sin comerlo ni beberlo, me ha brindado a toda esta gente.

Si existe el paraíso, yo ya sé con quién.
Manolo.

Grants and Funding

This work was financially supported by the Spanish Ministry of Economy and Competitiveness (MINECO), project FFI2016-79748-R, and cofinanced by the European Social Fund (ESF).

Manuel Francisco Aparicio was supported by the FPI 2017 predoctoral programme, from the Spanish Ministry of Economy and Competitiveness (MINECO), grant reference BES-2017-081202.

Abstract

Social Networking Sites (SNS) are the most important way of communication nowadays. They have changed how we interact with our friends and family, and even how companies target their clients, conduct market analysis and make business decisions. The amount of data that is being generated every day is virtually unlimited, and it can be used to conduct social media analyses and/or to train Machine Learning (ML) models. However, many handicaps need to be alleviated. SNS data is, typically, unstructured and written in natural language, and it presents misspelled words, contractions, *emojis*, and new semantic units that sometimes are a heavy burden for learning algorithms. A large dataset and multiple preprocessing steps are essential for almost any ML application in SNS.

Unfortunately, there is an inherent cost to gather and build labelled databases (human effort), and it constitutes a major drawback for low- to mid-budget ventures. Additionally, many applications may result in social consequences, thus they need to be audited. Both objectives fall into the interest of a multidisciplinary project called Nutcracker, that aims to detect, track, monitor an analyse radical discourse online. This dissertation is part of the project, and we propose in it *effortless and interpretable* mechanisms to tackle aforementioned disadvantages, using social network's mechanics as leverage. First, we present a reasoning mechanism based on similarity between users, that will allow us to deduce properties of unknown users, hence reducing the effort required to build databases. Then, we present a new kind of feature extraction and selection method whose purpose is to reduce model complexity, thus enhancing model comprehensibility and transparency. Finally, we study the peculiarities of aggregated analysis and, particularly, how well can *class prevalence count* be estimated when working with SNS data.

Our results show that we are able to build large databases in Twitter with a fraction of the effort; that we can train interpretable models as accurate as the baselines but one order of magnitude less complex; and that *quantification* is a novel approach that has much to offer to social network analysis, since it is able to adjust classification bias. We developed a *proof-of-concept* tool for effortless labelling and continuous user tracking, and we tested the platform by producing four high-quality weak-labelled datasets. The proposed techniques, methodologies and tools have been proven useful for disciplines such as computational linguistics, political science and cybersecurity. They are being used by members of our team and they have raised the attention of Spanish Civil Guard. Applications include building (and working with) supervised databases (e.g., social network analysis, market analysis, customer service, user profiling...); reaching full transparency in automatic decision-making algorithms (e.g., preemptive account closing, illegal activity tracking, hiring policies...); measuring overall user opinion or sentiment (e.g., during an event like a political debate); studying mental illnesses, detection of epidemic outbreaks, targeting customers, profiling brand ambassadors, or determining the impact of organised communities, among many others.

Resumen

Las redes sociales son el medio de comunicación más importante hoy en día. Han cambiado la manera que tenemos de interactuar con nuestra familia y amigos, e incluso la manera que tienen las empresas de realizar estudios de mercado, tomar decisiones de negocio o dirigirse a sus clientes. La cantidad de datos que están siendo generados cada día puede considerarse ilimitada, y puede usarse para realizar estudios sociales o para entrenar modelos de aprendizaje computacional (ML). Sin embargo, existen dificultades con las que lidiar. La información recogida de redes sociales es mayormente desestructurada y escrita en lenguaje natural, y puede presentar faltas de ortografía, contracciones, *emojis*, y unidades semánticas nuevas, que pueden resultar una carga para los algoritmos de aprendizaje. Una buena base de datos y varios pasos de preprocesamiento se vuelven requisitos indispensables para casi cualquier aplicación de ML en redes sociales.

Por desgracia, existen costes nada despreciables para producir dichas bases de datos (esfuerzo humano), y constituye una de las mayores desventajas para empresas de bajo y medio presupuesto. Además, muchas de estas aplicaciones pueden tener repercusiones sociales, por lo que necesitan ser auditadas. Ambos objetivos caen dentro del ámbito de un proyecto multidisciplinar llamado Nutcracker, cuyo objetivo es detectar, rastrear, monitorizar y analizar el discurso radical en Internet. Esta tesis es parte del proyecto, y en ella proponemos diferentes mecanismos interpretables y de esfuerzo reducido para abordar las desventajas existentes, utilizando en nuestro beneficio las propias mecánicas de las redes sociales. Primeramente, presentamos un mecanismo deductivo de razonamiento basado en similitud entre usuarios, que permiten inferir

propiedades de usuarios desconocidos y, por consiguiente, reducir el esfuerzo necesario para producir la base de datos. Posteriormente, presentamos un nuevo tipo de característica cuya finalidad es reducir la complejidad de los modelos una vez entrenados, consiguiendo así una mayor comprensibilidad y transparencia. Finalmente, estudiamos las peculiaridades del análisis agregado y, en especial, cómo de buenos son lo métodos actuales estimando la prevalencia de las clases en muestras de datos de redes sociales.

Nuestros resultados muestran que somos capaces de construir grandes bases de datos de Twitter con una fracción del esfuerzo normal; que podemos entrenar modelos interpretables tan precisos como siempre pero reduciendo su complejidad en un orden de magnitud; y que la cuantificación es una disciplina con mucho que ofrecer al análisis de redes sociales, ya que es capaz de ajustar el sesgo de clasificación. Hemos desarrollado una herramienta como prueba de concepto que es capaz de reducir el esfuerzo de etiquetado de *datasets* y de la monitorización continua de usuarios relevantes, y la hemos puesto a prueba mediante la producción de cuatro bases de datos. Las técnicas, metodologías y herramientas propuestas han demostrado ser efectivas en diferentes ámbitos, como las ciencias políticas, la lingüística y la ciberseguridad. Están siendo usadas por expertos de nuestro proyecto y han llamado la atención de la Guardia Civil por su potencial. Las aplicaciones incluyen la producción de bases de datos supervisadas (por ejemplo, para análisis de redes sociales, estudios de mercado, atención al cliente, caracterización de perfiles de usuarios...); la aplicación de algoritmos de toma de decisiones completamente interpretables (por ejemplo, para el cierre preventivo de cuentas, rastreo de actividades ilegales, políticas de contratación...); la medición de la opinión general de una población (por ejemplo, durante un evento, como un debate político); el estudio de enfermedades mentales, la detección de epidemias, para campañas de atracción de clientes, o para determinar el impacto de comunidades organizadas, entre otras muchas.

Contents

1 Introduction

In the last decade, Social Networking Sites (SNS) have become one of the most important means of communication. They have given a say to everyone regardless of their status, and they influence our society in a way that it even affects our relations, economy and democracy [1]. SNS enables a new way of organisation, because it is easier to make contact and establish relations between people with similar interest all around the world; they are also a new collaboration tool, since it facilitates working and sharing knowledge remotely; they are also being used in education and, in some cases, as a replacement of traditional institutions, since it is possible to offer materials and lessons in an ubiquitous manner; they constitute an unlimited resource for polling and opinion mining, and companies are using them not only to promote their brands but also to take informed business decisions. All in all, it seems like the limits of the potential of SNS are yet to be established.

According to *Digital 2022* [2], there are 4.62 billion active social media users that spend 2 hours and 27 minutes on these platforms every day. *Domo Resource - Data Never Sleeps 9.0* [3] reports that, every minute, there are 575 thousand tweets being posted, 305 thousand photos being shared on Instagram and Facebook, and 694 thousand hours are streamed in Youtube. Only in 2021, we consumed 79 zettabytes of data and it is expected to reach 180 zettabytes by 2025. These overwhelming statistics raises the question of data analysis capabilities. *Is it even possible to learn from all this data?* The problem is even bigger when facing supervised learning environments, in which the data is required to be labelled in order for Machine Learning (ML) models to learn upon.

There is a growing number of research articles and conference papers that are using SNS to conduct their studies. Figure 1.1 🖼 shows keywords for recent research trends related to Social Networks. Since 2020, a lot of research work related to COVID-19 pandemic has been carried out using SNS data (e.g., Ng et

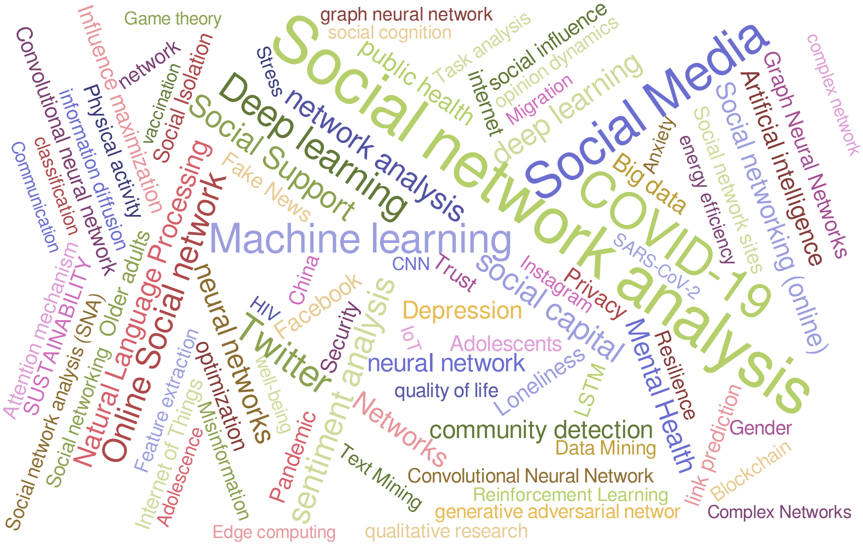


Fig. 1.1.: Wordcloud of frequent keywords co-occurring with *Social Network*. Keywords source: Scopus.

al. [4]). There are other trending health-related issues, such as *depression* [5], *adolescence* [6], *social isolation* [7], *anxiety* [8], and *loneliness* [9], among others.

Of all SNS, Twitter is the most popular one amidst researchers [10]. The peculiarities of the social network (short-text messages, prompt response to events, rapid information diffusion...) make the platform very attractive for certain users that are interested in time-sensitive aspects. Moreover, it offers a public API that is convenient for data retrieval, although with some limitations. Twitter has inspired many studies and applications, some of them related to the structure of the platform itself and the flow of information [11, 12, 13]; or the polarisation of its users [14, 15, 16]; or as a support tool for disaster and crisis management [17, 18, 19]; or to perform brand sentiment analysis [20, 21, 22]; or many others like food poisoning tracking [23], analysing political strategies [24], detecting fraud [25], or stock prediction [26].

But not all use cases are legit. According to Morgan [27], Twitter has hosted at least 46000 ISIS supporters. The Islamic State has taken advantage of the popularity of the social network to draw the attention of sympathetic people susceptible to radicalisation, to spread propaganda and even to coordinate attacks. Propaganda and radical discourse resort to the intended use of emotions to facilitate the engagement through hate and a sense of brotherhood, both at the same time.

This is the motivation of The Nutcracker Project, that aims to *detect, monitor and track* terror-related accounts and *analyse* their discourse. To that end, we have a **multidisciplinary** team composed of political scientists, linguists, mathematicians and computer scientists. The contributions of The Nutcracker Project are multiple. Mainly, the team has developed a theoretical model for language evaluation, whose effectiveness has been proved in several context (i.e., tweets, interviews, discourse, propaganda articles) [28, 29, 30, 31]. The model does not only offers new discoveries in linguistics but also in psychology of emotions. Emotive persuasion appeals to human values in which there is an undisputed consensus, hence the success of radicalisation in ideological (i.e., terrorism) or political values (i.e., populism).

The latter can also be evaluated from the political perspective. The radicalisation process has similar origin in both cases: generalised anger and distrust towards political elites [32]. The characterisation of this process, its origins, and proliferation, are interesting research lines for political scientists, and the analysis of traditional and social media is crucial towards that end [33, 34, 35, 36, 37, 38, 39].

This PhD dissertation is framed into The Nutcracker Project, and it aims to solve the technological needs of a multidisciplinary project. Our work constitutes the first step towards developing tools, methodologies and techniques that can be used to track and mine published content of interest groups, such as terror-related accounts, and it is being used for linguistic and political applications within our project.

However, it is not an easy task. When conducting research on Social Media, there are three things that often entail big handicaps for freelancers or entities with medium to low budget:

1. Data availability. Although there are open data initiatives like *Open Data Watch*[1] or even governmental platforms like *European Data Portal*[2], almost all of SNS data belongs to private companies that offer services to users and their data to anyone who is willing to pay for it (and it is not cheap).

[1] `https://opendatawatch.com/`

[2] `https://data.europa.eu/`

2. Data supervision. Whenever the rules of supervised environments apply, it is necessary to label data. Building well-annotated datasets require experts or trained workers and time. Either labelled in-house or outsourced, data supervision is costly even for medium-sized datasets.

3. Training costs. Resources required to train machine learning models, specially state-of-the-art models based on Deep Artificial Neural Networks (DNN), are expensive to build, maintain and/or rent.

Accounts that violate Twitter's *terms of service*[3] try to stay hidden, with a low number of connections and repercussion outside of their targeted scope (covert networks). Additionally, they use a variety of languages, including English and French but, mainly, Arabic (in multiple dialects). This situation makes particularly hard to gather and analyse content from such accounts, since manually-labelling a database of propaganda and recruitment posts is a hard task that requires a lot of human labour (effort). Current automatic solutions are based on textual features that depend on the language (cross-lingual ML approaches), which are not only difficult to build but also require a large, good quality dataset.

In 2014, Twitter started banning suspected accounts related to terror and, since then, they have continue to do so. Preemptive account closing has raised the debate whether or not it limits free speech, hence suggesting that this line of action may have a social impact. Consequently, algorithms and models use to detect and track these accounts should be auditable, fair and transparent [40].

Arguably, any model whose decision may have a social impact is required to be accurate, transparent and fair. The decision making process on the legitimacy of the accounts that are related to politics and religion needs to be taken with caution (i.e., a false positive may compromise free speech). In other words, it is required to use *interpretable* ML.

In recent years, Deep Artificial Neural Networks (DNN) have become the cutting-edge technology in almost every discipline. Since 2012, *deep learning* is replacing other classical techniques due to its accuracy and capability to extract data from both structured and non-structured data. However, the inherent complexity of dense layers of artificial neurons makes DNN a *black-box* approach, hence it is not interpretable.

[3] `https://twitter.com/en/tos`

Interpretability-related research has focused, mainly, in surrogate models [41]. It consists in training an interpretable model to predict the output of the main black-box model, and then use the interpretation of the former as an explanation for the behaviour of the latter. However, there are several caveats to consider [42]. Mainly, it is possible to find explanations that do not respond to actual knowledge learned from the data but to the inherent bias of the learning process. And, in any case, it is not a valid solution to generally describe (or prove) how the model will behave when facing unknown situations.

Additionally, there is a discussion on how comprehensible are interpretable models, and the trade-off between interpretability and model performance [41, 43]. Once trained, depending on the complexity of the data, resulting model may be interpretable yet impossible to understand due to its numerous ramifications. Despite that most experts agree that interpretability is an important dimension to evaluate, most work often avoid facing the question directly [44].

For these reasons, this document focuses in proposing new interpretable techniques that can be used to analyse social media content with a reduced effort.

Research Hypothesis

Despite that SNS data is usually unstructured, there is a lot of associated metadata that may offer information about the context and help with the interpretation of trained models when performing aggregated analyses.

Social Network's mechanics (such as befriending someone) not only offer explicit information but they also present some implications that may be used to our advantage. For example, in the case of Twitter, it is possible to *retweet* (or republish) some other user's content in you own timeline. It implies not only that you are interested in the content (otherwise you would not be interacting with it) but also that you agree with what is said in it (otherwise you would not share it with your followers)[4].

[4] At the end of 2020, Twitter introduced the possibility of quoting *tweets*. When the user clicks on the *retweet* button, the platform asks for the user's genuine opinion regarding the matter, as if it were a *quote*. Throughout this document, we will distinguish between a *retweet* and a *quote*.

The particulars of these mechanics can be generalised using abstract high-level relations that enable new reasoning mechanisms. In fact, that kind of reasoning may be used to tackle one of the major handicaps presented above: data supervision costs. Our hypothesis is that it is possible to use similarity-based reasoning to reduce the effort required to detect and characterise accounts of interest in an interpretable manner.

For example, imagine a situation in which a given user has been characterised as *sympathetic to liberal ideology*. We cannot assume that their followers will also have *liberal ideology*, since the reasons to follow an account may be arbitrary and/or not related to politics. On the contrary, if we had a high-level relation that implies that connected users *are similar in terms of political thinking* it would be relatively safe to assume that both users have *liberal ideology*.

Deductive reasoning by similarity enables a way to obtain information from uncharacterised user accounts, as long as they are connected to influential users that have already been characterised. Moreover, if the criteria to build such high-level relations are human readable, the reasoning mechanism would be fully interpretable.

However, if automatic mechanism were used to characterise key users, those should be interpretable in order for the conclusions to be auditable, transparent and fair. In fact, they should be not only interpretable but also simple enough to be understandable by humans (low complexity). In this sense, we think that it possible to reduce the complexity of trained interpretable models by using less but more meaningful features, carefully build and selected to help the classifier in its duty. For example, by using a set of input features that are closely related to a particular class (and not the others).

Both fronts combined cover the individual analysis of social network accounts. Yet, what happens with aggregated analysis? Classification algorithms focus in minimising classification error, and thus they will take decisions that, notwithstanding being statistically accurate, present an inherent bias towards the *most probable* class.

This is specially problematic in such cases in which feature and class probability distribution of test sets differ from training sets [45]. Twitter and other SNS present that kind of behaviour when some topic become trending, or when the focus of the users shift to another perspective or matter. For example, hashtags that cover political debates have different clusters of documents, each of them pertaining to one of the subjects discussed during the event (economy, foreign affairs, social politics...).

There are several mechanisms that can be used to deal with drifts between training data and real-world scenarios, i.e., samples that are not independent and identically distributed. Quantification is the task of estimating class prevalence values, that is particularly useful when determining sentiment prevalence values in Twitter [46]. We think that it is possible to perform interpretable aggregated analysis using quantification methods, that are able to improve the estimates by *learning* from the classification bias.

Research Objectives

As its main objective, this PhD dissertation aims to develop techniques, methodologies and tools to detect, track, monitor and analyse groups of interest in Social Networking Sites (SNS) with a reduced effort and high interpretability, within the framework of The Nutcracker Project. This objective focuses on the following landmarks:

Effort Reduction

① Introduce similarity-based reasoning mechanism using high-level relations in social networks.

There are specific interactions in SNS that can be use to build high-level relations. A network of users that are connected through links that represent *common interest* or *similar ideology* enable deductive reasoning (Similarity Semantic Networks). For example, if two given users are interested in politics and have common opinions, it is safe to extend annotated properties from one to another up to a certain degree. This approach yield a field of new possibilities, such as weak-annotation of SNS users.

② Develop an interpretable methodology to effortless supervise datasets that are useful for Social Networking Sites (SNS) analysis and training of Machine Learning (ML) models.

Since opinion mining pipelines are built upon SNS data, a mechanism to retrieve and delimit regions of interest (in the network of users) is a necessary consequence of Similarity Semantic Networks. This methodology should allow the use of trained ML models or human oracles, interchangeably.

3 Implement a *proof-of-concept* solution of this methodology to test whether it is a valid approach to build SNS datasets.

Testing aforementioned methodology requires to put it in practice. A *proof-of-concept platform* not only gives us that possibility but also serves as a stepping stone to develop powerful annotation tools based on similarity.

Improve Interpretability

4 Study current methods for interpretable Machine Learning (ML) pipelines to establish how comprehensible they are.

In order to evaluate interpretable models comprehensibility, it is necessary to establish baselines using current approaches. There are two main ways of addressing the issue, one of them being the use of inherently-interpretable models and the other the use of surrogate models to explain black-box ones. As the latter also require the use of the former, our evaluation is going to be focused on the comprehensibility of interpretable models, such as decision trees.

5 Propose new mechanisms in any of the steps of the classification pipeline to ensure comprehensibility of interpretable models.

During the training process, models try to find patterns in the features that are related to a class. Then, during the prediction phase, it tries to find learned patterns to guess the class of an unknown instance. Preprocessing steps may include extraction and selection of new features that facilitate the learning process of the model with the purpose of obtaining a simpler one.

Reduce Aggregation Bias

6 Understand types of bias that may be present in Social Networking Sites (SNS) data.

Facing real-world scenarios imply dealing with noise or data from different nature. In order to better deal with these situations, it is necessary to study their origin, their implications and their repercussions.

7. Study applicability of quantification models to SNS data.

8. Explore performance of standard quantifiers to establish the influence of different types of bias in the training and validation sets.

   There are different types of quantifiers that rely on different principles to estimate prevalence counts. It is necessary to select the most representative ones to setup a battery of experiments and understand their strength and weaknesses when dealing with different drifts.

9. Propose new adjustments of the prevalence count that take advantage of spatial and/or temporal features.

   In the same manner that similarity reasoning can be used to propagate labels from known users to unknown ones, similarity measures can be used to adjust the prevalence estimates. When dealing with SNS data, particularly with Twitter data, both spatial and time relations between similar instances may be present. For example, tweets with the same hashtag that were published close in time are likely to be related. On the contrary, if they have a two-year difference, it is likely that the topic has diverged. Exploiting these relations may yield better results.

Outline

The present document is structured in five chapters, as follows. Chapter 2 deals with objectives one, two and three, that are related to Similarity Semantic Networks, our methodology to build weak-supervised dataset and our platform, Nutcracker. Chapter 3 tries to improve model comprehensibility while addressing objectives four and five. It introduces *distinguishing features* and our ranking mechanism, *CF-ICF*. Chapter 4 explore quantification literature for Twitter and study the behaviour of several quantifiers facing different kinds of data bias. Chapter 5 offer an exploratory analysis of one of our produced dataset, in terms of feature and class distribution, correlations and frequent patterns. Finally, chapter 6 sums up the principal contributions of this work, as well as future lines of research.

Introducción

1

Durante la última década, las redes sociales (Social Networking Sites) se han convertido en uno de los medios de comunicación más importantes. Han dado voz a todo el mundo, con independencia de su estatus socio-económico, y han conseguido cambiar nuestra sociedad de tal manera que afecta, incluso, a nuestras relaciones, economía y forma de gobierno [1]. Las posibilidades de las redes sociales son múltiples: constituyen una nueva manera de organización, ya que facilita el contacto entre personas de intereses similares a lo largo del mundo; también se han convertido en una herramienta de colaboración, ya que facilita el trabajo remoto y la difusión del conocimiento; pueden ser usadas en educación y, en algunos casos, como sustitutas de instituciones educativas tradicionales, ya que es posible ofrecer materiales y contenido docente de manera ubicua; constituyen una fuente ilimitada de recursos para encuestas y *opinion mining*, por lo que hay empresas que las usan para tomar decisiones de negocio informadas. En resumen, parece que los usos potenciales de las redes sociales están aún por delimitar.

Según *Digital 2022* [2], existen 4,62 mil millones de usuarios activos en redes sociales, que dedican unas 2 horas y 27 minutos al día a consultar las distintas plataformas. *Domo Resource - Data Never Sleeps 9.0* [3] cifra, por cada segundo, 575 mil tweets publicados, 305 mil fotos compartidas en Instagram y Facebook, y 694 mil horas de vídeos transmitidos en Youtube. Se han consumido 79 zettabytes de información en 2021, y se espera alcanzar los 180 zettabytes en 2025. Estas estadísticas suscitan preguntas acerca de las capacidades de análisis de datos actuales. *¿Es posible manejar todos estos datos y extraer conocimiento de ellos?* El problema es incluso mayor en entornos de aprendizaje computacional supervisados, donde la información ha de estar etiquetada para que los modelos de aprendizaje computacional (Machine Learning) puedan aprender.

Hay una tendencia creciente de artículos y conferencias en las que se utilizan las redes sociales para realizar investigaciones. En la figura 1.1 mostramos las palabras clave de trabajos recientes relacionadas. Fíjese que, desde 2020, se

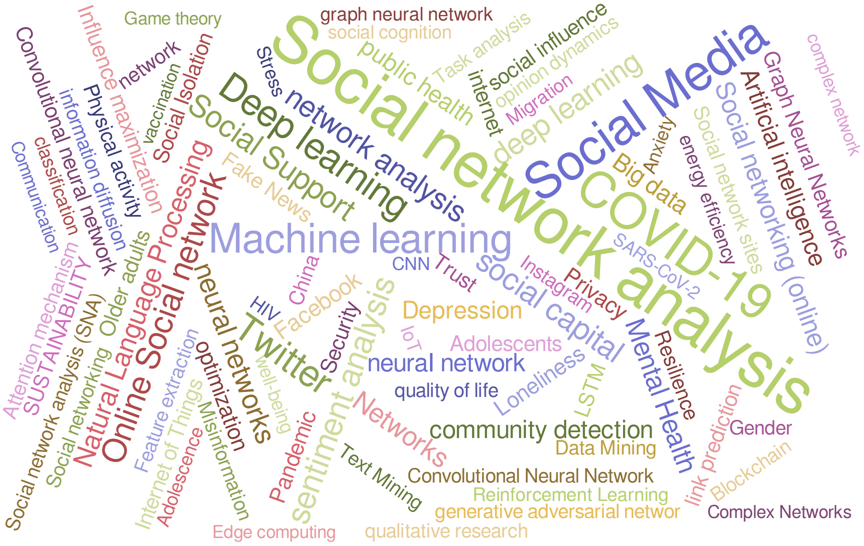


Figura 1.1.: Nube de términos frecuentes que se dan junto a la palabra clave *Social Network*. Fuente: Scopus.

han llevado a cabo una gran cantidad de investigaciones en redes sociales en el contexto de la pandemia de la COVID-19 (por ejemplo, Ng y col. [4]). También existen otras tendencias relacionadas con la salud, como la *depresión* [5], *adolescencia* [6], *aislamiento social* [7], *ansiedad* [8], y *soledad* [9], entre otras.

De todas las redes sociales, Twitter es la más popular entre investigadores [10]. Las peculiaridades de la plataforma (mensajes cortos, respuesta temprana a eventos, propagación rápida de la información...) hacen que sea muy atractiva para ciertos tipos de usuarios interesados en cuestiones donde el tiempo es un factor importante. Además, ofrece una API pública que facilita la recuperación de la información, aunque con ciertas limitaciones. La red social ha inspirado numerosos estudios y aplicaciones, algunos de ellos relacionados con la estructura de la plataforma en sí misma y el flujo de información dentro de ella [11, 12, 13]; o con la polarización de sus usuarios [14, 15, 16]; o del uso de la red como una herramienta de soporte ante desastres naturales [17, 18, 19]; o para realizar estudios sobre la opinión de los usuarios respecto a una marca [20, 21, 22]; y para muchas otras aplicaciones, como el seguimiento de intoxicaciones alimentarias [23], análisis de estrategias políticas [24], detección del fraude [25], o predicción del *stock* [26].

Sin embargo, no todos los casos de uso son legítimos. Morgan [27] asegura que Twitter ha hospedado al menos 46000 cuentas de simpatizantes del Estado Islámico (ISIS). El grupo ha usado en su beneficio la popularidad de la red social con el objetivo de llamar la atención de simpatizantes y personas susceptibles de ser radicalizadas, para divulgar propaganda e, incluso, para coordinar ataques.

Tanto la propaganda como el discurso radical hacen uso de las emociones para facilitar el compromiso de sus víctimas, a través del odio y el sentimiento de hermandad. Esta es, precisamente, la motivación del proyecto Nutcracker, cuyo objetivo es la detección, monitorización y rastreo de cuentas relacionadas con el terrorismo, y el análisis de su discurso. Con tal fin, disponemos de un equipo **multidisciplinar** compuesto de politólogos, lingüistas, matemáticos e informáticos. Las contribuciones del proyecto Nutcracker son diversas. Principalmente, el equipo ha desarrollado un modelo teórico del lenguaje evaluativo, cuya efectividad ha sido probada en diferentes contextos (por ejemplo, en tweets, entrevistas, discursos y artículos propagandísticos) [28, 29, 30, 31]. El modelo no solo ofrece avances en lingüística, sino también en la psicología de la emoción. La persuasión emocional apela a valores humanos firmemente consensuados, de ahí el éxito de la radicalización ideológica (terrorismo) y/o política (populismo).

Este último punto puede ser abordado también desde una perspectiva política. El proceso de radicalización tiene un origen similar en ambos casos: enfado generalizado y desconfianza hacia las élites políticas [32]. La caracterización de este proceso, sus orígenes y proliferación, son líneas de investigación interesantes, y el análisis de los medios tradicionales y de las redes sociales es crucial de cara a ello [33, 34, 35, 36, 37, 38, 39].

Esta tesis se enmarca dentro del proyecto Nutcracker, y su objetivo es resolver las necesidades tecnológicas de un equipo multidisciplinar. Nuestro trabajo es el primer paso de cara a desarrollar herramientas, metodologías y técnicas que puedan ser usadas para rastrear y minar el contenido publicado por grupos de interés, como aquellos que tengan relación con el terrorismo. Actualmente, nuestras propuestas están siendo usadas por lingüistas y politólogos de nuestro proyecto.

No obstante, ésta no es una tarea sencilla. Cuando llevamos a cabo investigaciones en redes sociales, hay tres puntos que normalmente suponen graves dificultades para investigadores independientes o instituciones con presupuestos bajos o medios:

1. Disponibilidad de datos. Aunque hay iniciativas abiertas como *Open Data Watch*[1] e incluso alternativas institucionales como *European Data Portal*[2], la mayoría de información relativa a redes sociales pertenece a compañías privadas que ofrecen servicios a sus usuarios y los datos generados a quien pueda pagarlos (y no es barato).

2. Etiquetado de los datos. En entornos supervisados, es necesario etiquetar la base de datos. La producción de *datasets* bien etiquetados requiere de la colaboración de expertos o trabajadores previamente entrenados y, sobre todo, de tiempo. Con independencia de que el etiquetado sea externalizado o no, el proceso es costoso incluso para bases de datos medianas.

3. Costes de entrenamiento. Especialmente en el caso de redes neuronales artificiales profundas (Deep Artificial Neural Networks), que actualmente representan el estado del arte, los recursos necesarios para el entrenamiento son costosos.

Las cuentas que violan los términos de uso de Twitter[3] intentan permanecer ocultas, con un número reducido de conexiones y baja repercusión fuera del entorno objetivo (estas redes se conocen como *covert networks*). Además, usan varios idiomas, entre los que se encuentran el inglés y el francés pero, principalmente, el árabe en múltiples dialectos. Esta situación hace que sea especialmente difícil obtener y analizar contenido de dichas cuentas, ya que el etiquetado manual es costoso y requiere mano de obra humana. Las soluciones automáticas se basan en características textuales que dependen del lenguaje que, además de costosas de producir, requieren una base de datos grande y de calidad.

Por otro lado, Twitter comenzó en 2014 a eliminar cuentas sospechosas de estar relacionadas con el terrorismo y, desde entonces, ha continuado haciéndolo. El cierre preventivo de cuentas ha abierto el debate de si dicha práctica limita

[1] https://opendatawatch.com/

[2] https://data.europa.eu/

[3] https://twitter.com/en/tos

la libertad de expresión, lo cual suscita que tal acción tiene impacto social. En consecuencia, los algoritmos y modelos que se usan para detectar y rastrear dichas cuentas deberían ser auditables, justos y transparentes [40].

Cualquier modelo que pueda tener consecuencias sociales debe ser preciso, transparente y justo. El proceso de toma de decisiones que determina si una cuenta está legítimamente relacionada con política y/o religión debe ser cauteloso (un falso positivo puede comprometer la liberad de expresión) o, en otras palabras, requiere el uso de modelos interpretables.

Desafortunadamente, el estado del arte no contempla esta dimensión y no es válido de cara a dichas aplicaciones. Últimamente, las redes profundas (DNN) se han convertido en la tecnología más innovadora en casi todas las disciplinas y, desde 2012, han venido reemplazando a otras técnicas clásicas debido a su precisión y capacidad de extraer conocimiento de información estructurada y no estructurada. Sin embargo, la complejidad inherente a estos modelos con múltiples capas densas de neuronas hace que esta solución se comporte como una caja negra y, por consiguiente, no pueda ser usada en procesos que resulten en un impacto en nuestra sociedad.

La literatura relacionada con la interpretabilidad de los modelos se ha venido enfocando en modelos *ad-hoc* [41]. Esta técnica consiste en entrenar un modelo interpretable para predecir la salida del modelo principal, de tal manera que las explicaciones aportadas por el modelo ad-hoc sirvan para documentar el modelo principal. Sin embargo, existen cuestiones que hay que tener en cuenta [42]. Principalmente, es posible que aparezcan explicaciones que no respondan a conocimiento real aprendido de los datos sino al sesgo inherente del proceso de aprendizaje y, en cualquier caso, este tipo de soluciones no realizan (ni prueban) una descripción general del modelo, por lo que es imposible saber cómo se comportará éste ante situaciones desconocidas.

Adicionalmente, existe un debate sobre cómo de comprensibles son, en realidad, los modelos interpretables y sobre el equilibrio entre interpretabilidad y precisión [41, 43]. Una vez entrenados, dependiendo de la complejidad de los datos, los modelos resultantes pueden ser incomprensibles con independencia de que sean interpretables, debido a las numerosas ramificaciones que pueden presentar. A pesar de que los expertos están de acuerdo en que la interpretabilidad es una dimensión más a evaluar, la mayoría de trabajos evitan hacer frente directamente a esta cuestión [44].

Por estas razones, este documento se centra en proponer nuevas técnicas interpretables que puedan ser utilizadas para el análisis de redes sociales con un esfuerzo (mano de obra humana) reducido.

Hipótesis de Investigación

A pesar de que la mayoría de información de redes sociales es contenido no estructurado, hay muchos metadatos asociados que pueden arrojar información sobre el contexto y ayudar con la interpretación de los modelos ya entrenados.

Las mecánicas presentes en las redes sociales (como añadir un amigo o seguir a alguien) no solo ofrecen información explícita (conocer a una persona o estar interesado en el contenido que genera, respectivamente), sino que también suponen *implicaciones* que pueden usarse para facilitar la consecución de nuestros objetivos. Por ejemplo, en el caso de Twitter, es posible *retweetear* (o re-publicar) contenido de otros usuarios como si fuera nuestro. Esto implica que (1) estamos interesados en la temática que se trata (en otro caso, no estaríamos consumiendo dicho contenido) y (2) estamos de acuerdo con lo que se dice en dicho tweet (de lo contrario, no lo compartiríamos con nuestros seguidores)[4].

Las particularidades de estas mecánicas pueden ser generalizadas utilizando relaciones abstractas de alto nivel que habilitan nuevos procedimientos de razonamiento. De hecho, este tipo de razonamiento puede usarse para lidiar con una de nuestras mayores dificultades: los costes de etiquetado. Nuestra hipótesis es que podemos usar razonamiento por similitud para detectar y caracterizar cuentas de interés de manera interpretable.

Por ejemplo, imagine una situación en la que un usuario ha sido caracterizado como *empático hacia ideologías liberales*. No es correcto asumir que sus seguidores también compartirán *ideología*, ya que hay muchas razones para seguir a otra cuenta y no tienen por qué estar relacionadas con política. Por el

[4] A finales de 2020 y con motivo de la elecciones de Estados Unidos, Twitter introdujo una mecánica de citas en los *retweets*. Cuando el usuario hace clic sobre el botón de *retweet*, la plataforma ofrece la posibilidad de que el usuario exprese su opinión sobre el contenido que quiere compartir, como si de una cita se tratase. En este documento, diferenciaremos el *retweet* puro de la *cita* (o *quote*).

contrario, si tenemos una relación de alto nivel que implica que los usuarios conectados *son parecidos respecto a su pensamiento político*, sería relativamente seguro asumir que ambas cuentas tienen *ideología liberal*.

El razonamiento deductivo por similitud permite obtener información de usuarios no caracterizados, siempre y cuando estén conectados a usuarios influyentes que ya hayan sido caracterizados previamente. Además, si el criterio para construir dichas relaciones de alto nivel puede ser fácilmente comprendido por humanos, los mecanismos de razonamiento serán completamente interpretables.

La caracterización de los usuarios clave, de ser realizada mediante métodos automáticos, también debería ser interpretable y fácilmente comprensible por humanos (baja complejidad de los modelos). En este sentido, creemos que sería posible reducir la complejidad de los modelos interpretables entrenados usando un menor número de características siempre que nos aseguremos que son útiles, mediante su extracción y selección usando el criterio de *utilidad* para el clasificador. Esto sería posible, por ejemplo, si seleccionamos características de entrada estrechamente relacionadas con una clase concreta (pero no con las otras).

La combinación de ambas hipótesis cubren el análisis de cuentas de redes sociales, pero ¿qué pasa con el análisis agregado? Los algoritmos de clasificación buscan minimizar el error, y por consiguiente toman decisiones que, siendo estadísticamente correctas, presentan sesgos hacia la clase *más probable*.

Esto es especialmente problemático en aquellos casos en los que las distribuciones de las características y/o de las clases difieren entre los conjuntos de entrenamiento y de validación [45]. Twitter y otras redes sociales presentan dichas diferencias cuando las temáticas se vuelven *tendencias*, o cuando el centro de atención de los usuarios cambia de perspectiva. Esto es común, por ejemplo, en *hashtags* que cubren eventos, como debates políticos, donde se discute sobre diferentes temáticas (economía, asuntos exteriores, política social...).

Hay diferentes mecánicas para lidiar con *derivas* entre los conjuntos de entrenamiento y lo observado en el mundo real (muestras que no respetan el principio i.i.d., *independent and identically distributed samples*). La cuantificación es la tarea que aborda la estimación de la prevalencia de las clases en una muestra, y es particularmente útil para determinar la distribución de *sentiment* en Twitter [46]. Nuestra tercera hipótesis establece que es posible realizar análisis agregados interpretables utilizando cuantificadores, que son capaces de mejorar las estimaciones ajustando el sesgo de clasificación.

Objetivos de Investigación

Como objetivo principal, esta tesis pretende **desarrollar técnicas, metodologías y herramientas** para detectar, rastrear, monitorizar y analizar **grupos de interés** en redes sociales, con **esfuerzo reducido y alta interpretabilidad**, en el contexto del proyecto Nutcracker. Este objetivo se concreta en los siguientes:

Reducción del Esfuerzo

1. Presentar un mecanismo de razonamiento basado en similitud utilizando relaciones de alto nivel en redes sociales.

   Hay interacciones específicas en las redes sociales que se pueden usar para construir relaciones de alto nivel. Una red semántica que conecta a usuarios usando enlaces que representen *intereses en común* o *ideas similares* habilita el razonamiento deductivo (Similarity Semantic Networks), y serviría para extrapolar características de usuarios conocidos a otros de los que no tenemos información.

2. Desarrollar una metodología interpretable para etiquetar con menos esfuerzo bases de datos que sean útiles para el análisis de redes sociales y/o para el entrenamiento de modelos de aprendizaje computacional.

   Teniendo en cuenta que los modelos de minería de opiniones se construyen sobre datos de redes sociales, la posibilidad de recuperar información y delimitar regiones de interés dentro de la red de usuarios es una consecuencia necesaria de las *Similarity Semantic Networks*. Esta metodología serviría para usar oráculos automáticos o humanos indistintamente.

3. Implementar una prueba de concepto de dicha metodología para comprobar su validez de cara a la producción de *datasets* etiquetados.

   La validación de la metodología requiere de su puesta en marcha. Una plataforma prueba de concepto nos brinda esa posibilidad y, además, establecería cimientos para desarrollar herramientas de etiquetado más potentes basadas en similitud entre usuarios.

Mejora de la Interpretabilidad

4 Estudiar los modelos interpretables existentes para determinar cómo de comprensibles son en realidad.

Para evaluar cómo de comprensibles son los modelos interpretables, se hace necesario establecer *baselines* utilizando técnicas actuales. Hay dos vias principales de abordar este problema, siendo una de ellas el uso de modelos interpretables y la otra el uso de modelos *ad-hoc* para explicar aquellos que sean de caja negra. Teniendo en cuenta que estos últimos precisan del uso de los primeros (y que presentan ciertas limitaciones), centraremos nuestros esfuerzos únicamente en el primer caso.

5 Proponer nuevos mecanismos en cualquiera de las etapas de clasificación para asegurar la comprensibilidad de los modelos interpretables.

Durante la etapa de entrenamiento, los algoritmos intentan encontrar patrones entre las características que estén relacionados con una clase. Posteriormente, durante la etapa de inferencia, intentan encontrar los patrones aprendidos para predecir la clase de la instancia desconocida. Las etapas de preprocesamiento pueden incluir extracción y selección de nuevas características que faciliten el proceso de aprendizaje del modelo y permitan obtener uno menos complejo.

Reducción del Sesgo en la Agregación

6 Comprender los tipos de sesgos que pueden presentarse en los datos recogidos de redes sociales.

Enfrentarnos a escenarios reales implica lidiar con ruido o información de distinta naturaleza. Para abordar mejor dichas situaciones, se requiere el estudio del origen, las implicaciones y las repercusiones de dichas diferencias.

7 Estudiar la aplicabilidad de los cuantificadores a bases de datos de redes sociales.

8 Explorar la precisión de los cuantificadores estándares para establecer la influencia de los diferentes tipos de variabilidad entre los conjuntos de entrenamiento y validación.

Hay diferentes tipos de cuantificadores construidos sobre principios diversos con el objetivo de estimar el recuento de cada clase. Es necesario seleccionar los más representativos y ejecutar una batería de experimentos que nos permita entender sus fortalezas y debilidades a la hora de predecir sobre conjuntos de datos de naturaleza diferente a aquellos con los que fueron entrenados.

9 Proponer nuevos tipos de ajustes sobre las estimaciones de prevalencia de clases que se basen en características espaciales y/o temporales de los datos.

De la misma manera que el razonamiento por similitud se puede usar para propagar etiquetas desde usuarios conocidos a otros no conocidos, las medidas de similitud pueden usarse para ajustar las estimaciones de prevalencia de clases. Cuando trabajamos con datos de redes sociales, y en particular con Twitter, pueden presentarse tanto características espaciales como relaciones temporales entre instancias similares. Por ejemplo, aquellos *tweets* con el mismo *hashtag* que fueron publicados en instantes temporales cercanos son más propensos a estar relacionados que aquellos que presenten dos años de diferencia, ya que es esperable que la temática haya divergido. La explotación de dichas relaciones puede arrojar mejores resultados.

Esquema del Documento

El presente documento se estructura en cinco capítulos. El capítulo 2 se enfoca en los objetivos uno, dos y tres, que están relacionados con las *Similarity Semantic Networks*, nuestra metodología para producir *weak-supervised datasets* y nuestra plataforma, Nutcracker. El capítulo 3 intenta mejorar la comprensibilidad de los modelos, desarrollando los objetivos cuatro y cinco: presenta las *expresiones diferenciadoras* y nuestro sistema de ponderación, *CF-ICF*. El capítulo 4 explora la literatura de cuantificación para Twitter y estudia el comportamiento de varios cuantificadores en diferentes situaciones donde los datos han sido alterados artificialmente. El capítulo 5 ofrece un análisis exploratorio de uno de las bases de datos producidas usando nuestra metodología, en materia de distribución de características y clases, correlaciones y patrones frecuentes. Finalmente, el capítulo 6 resume los principales aportes del presente trabajo, al igual que las futuras líneas de investigación.

2

Building Weak-Supervised Datasets with Minimum Effort

Objectives

1. Develop an interpretable methodology to effortless supervise datasets that are useful for Social Networking Sites (SNS) analysis and training of Machine Learning (ML) models.

2. Implement a *proof-of-concept* solution of this methodology to test whether it is a valid approach to build SNS datasets.

2.1 Introduction

Machine Learning is a subset of Artificial Intelligence that uses data in order to identify hidden patterns which can be used to predict a target variable. There are mainly two types of ML tasks according to the existence of labels in the data used to learn upon [47]:

- Supervised learning relies on algorithms that use labelled datasets to generalise knowledge in order to make predictions for unlabelled instances. They are normally used for classification or regression.

- Unsupervised learning tries to model the stochastic relations between features to obtain clusters of unlabelled instances with similar characteristics.

Both types are extensively used, and both of them present different advantages. In recent years, models trained over large-scale unlabelled datasets (e.g., language models) are becoming popular. However, to this moment, supervised models are the most popular approach and the *de facto* standard for industrial solutions.

Supervised learning requires good quality datasets that are difficult to put together. The process of dataset labelling presents a major handicap, which is related to the cost of building such databases. Labelling datasets requires that an oracle (normally a human expert or a committee of them) assigns accurate labels to every instance. This is called *data supervision*, *dataset labelling* or *dataset annotation*, and it is costly in two aspects:

- Economic budget, in such cases in which experts are hired or outsourced.
- Time budget, in the event of in-house labelling.

Dataset supervision is a complex and time-consuming task, when carried out in-house; and expensive, when outsourced. The process of building ground-truth datasets may take years depending on the complexity of the task and the volume of data required. In order to reduce the effort, current solutions take advantage of automatic or semi-automatic approaches, also known as weak supervision techniques. They consist on training models on large amounts of low quality data that is easy to put together (e.g., data programming, synthetic data, and non-expert freelance annotators). Alternative, Continuous Active Learning (CAL) may be used along with *relevance sampling* to speed up the manual annotation, however it results in well differentiated documents since *relevance* criterion yields those documents that are far from the decision boundary.

In NLP, weak supervision methods are usually based on textual features. In the case of SNS, documents present many classification handicaps like contractions or misspelled words, and/or short-context documents. However, SNS offer more features apart from text. Along with the typical metadata (timestamp, author, location...), there is also information regarding which entities (i.e., user, post, or both) have interacted or are related to each other. There are implications behind these interactions:

- Mention: the mentioned user is involved in the discussion.
- Reply: the content is relevant for the reader.

- Favourite: the content is relevant and they explicitly agree with it.
- *Retweet*: the content is relevant for the reader and they subscribe every word.
- Quote: the content is relevant for the reader, although they would like to clarify some points (may be *for* or *against*).

Our hypothesis is that it is possible to reduce the effort required to build a SNS dataset using reasoning by similarity. It is based on the idea that it is possible to infer qualities of unknown users by using known features of similar users. We introduce *deep relations*, that are high-level relations deduced from basic relations, as similarity measures. But not all similarity measures can be used to propagate knowledge. For instance, if two users have similar ideas with respect to politics, we can assume that there is a certain possibility that they support the same political party; however, *being similar with respect to politics* is completely unrelated to the food they like, therefore it would not be possible to infer *eating habits* from a similarity measure related to *politics*.

In order to maintain a network of similar users, we propose the concept of *Similarity Semantic Networks*. In these networks, entities are connected to one another using simple semantic relations. We use a second level relation to link basic relations with particular aspects. Knowledge inference happen by propagating values of basic relations through edges of similar users, when both the similarity measure and the basic relation are bounded to the same aspect.

Similarity Semantic Networks offer a full range of possibilities, but abstract relations need to be materialised for particular contexts. In this chapter, we implement an interpretable *proof-of-concept* solution that enables similarity reasoning in SNS, particularly in Twitter. We built a micorservice-based platform with several capabilities:

- Document gathering, using Twitter API to retrieve data in order to store it later on a database with the necessary abstractions, so it is possible to change the source of the documents (i.e. using another SNS API) without requiring further changes in the platform.
- Document labelling, using both automatic techniques and human oracles.
- Semantic Network building, using the relation and/or interaction specified from the ones that are available in the stored data.

- Label propagation, from labelled documents to users in the semantic neighbourhood.

- Review of propagated label, both with automatic rules and human oracles.

The rest of this chapter is organised as follows. Section 2.2 reviews related work, approaches and platforms used for dataset labelling. Section 2.3 introduces *Similarity Semantic Networks* and Section 2.4 explains what *deep relations* are. Our proposed methodology using these techniques is explained in Section 2.5, and our *proof-of-concept* implementation is described in Section 2.6. Section 2.7 illustrates the dataset production process. Section 2.8 describes the experimental setup, and results are discussed in Section 2.9. We conclude this chapter in Section 2.10.

2.2 Related Work

In general terms, there is a basic rule when building supervised datasets: **quality**. A consistent, well-curated dataset will result in better models than weakly-annotated ones [48]. There are several techniques currently being used to produced supervised and weak-supervised datasets. In this section, we review their advantages and limitations. Section 2.2.1 discusses the different approaches when labelling supervised datasets. Section 2.2.2 explores open-source and enterprise-grade platforms.

2.2.1 Approaches

There are several approaches that can be used to obtain supervised datasets, each of them with different weaknesses.

Automatic Approaches

The first division is related to automatic labelling techniques, that are synthetic labelling and data programming. Despite that both are valid methods, resulting datasets are not suitable to perform opinion mining, since instances are artificial. However, they can be used to train weakly supervised models, hence the reason to include them in this review [49].

Synthetic Labelling It is the process of automatically generating data that complies with several rules that try to preserve real-life restrictions [50]. Traditionally, instances were created with random values that followed each feature's distribution. Recently, more elaborated methods are being used. While they are cheaper, synthetic datasets may have less quality since generative models do not fully represent the possible casuistries that actual data presents.

There are many domains in which this technique is very useful. Frid-Adar et al. [51] presents an approach that relies on Generative Adversarial Network (GAN) combined with other data augmentation techniques to generate training images. They managed to significantly improve classification performance and they checked whether radiologist could differentiate between the actual set of images and the synthetic one, once again with good results. However, GANs require a complex training phase and their outputs are not guaranteed to be realistic. This would be specially problematic in text.

There is still room for improvement in Natural Language Processing (NLP) tasks. These are considered to be one of the most difficult ones, since creating synthetic text datasets requires text generation, which is considered a hard task [52]. Guan et al. [53] developed a GAN that is able to generate text for synthetic electronical medical records. Results are promising, although authors recognise some lack of cohesion that needs to be addressed.

Recently, the use of pre-trained language models (LM) is becoming the main approach, and models like GPT-2 [54] show astonishing quality. The main drawback of LMs are (1) the amount of time requiered to train, (2) the complexity of the models (therefore lack of interpretability), and that (3) most powerful models are not released by their authors (e.g. GPT-2, and later GPT-3, were released in the form of a demo version, with the fully-capable models restricted to selected researchers [55]).

Data Programming It consists in delegating the labelling task to a few weak-labelling algorithms (e.g. heuristics or scripts) [56]. These are programmed to respond to specific features of the instances and trigger labels that experts have studied to be correlated with the features [57]. In this case, data is realistic but quality is not ensured, since there are tasks, like Natural Language Processing (NLP), that are difficult to model with simple rules and, often, require interpretation.

Heo et al. [58] present a tool that combines manual annotation with data augmentation and data programming to build weak-annotated image datasets. They show promising results, but the tool not only requires good matching rules but also a fine pattern augmentation model and low-noise data. Moreover, each research topic would require an adapted version of the rule set, which would be unfeasible compared to other methodologies.

In NLP, Mallinar et al. [59] propose a simple yet elegant solution to iterative train models from a small labelled dataset using weak supervision. They expand the initial dataset by selecting relevant instances in batches and performing a classification consulting an oracle. They avoid much overhead in the training process since the expansion is independent from the downstream classifier.

Main drawback of this approach is that the quality of the dataset may suffer. ML models need that the accuracy is better than a random classifier to start learning. The learning and dataset generation process should be iteratively refined, until required performance is met. It is necessary to notice that, although there are many cases in which data programming improves results, some problems cannot be addressed with precision using this technique.

Manual Approaches

Internal Annotation It consists in using the available human resources to label the data. Quality is assured because the annotators are familiarised with the problem, but huge amount of resources and time are required. It is also possible to establish common criteria and to monitor the process at any time, which result in consistent datasets.

Outsourced Labelling Outsourcing the labelling task to freelancers or specialised companies. When using freelancers, it is possible to hire them directly or to use crowdsourcing platforms. These platforms provide tools and an organised workflow in order to obtain faster and cheaper results. However, the quality of the dataset may suffer due to multiple reasons (like language, disinterest and/or urgency, because of the fact that their income depends on the number of labelled instances). When outsourcing to specialised companies, the quality improves but so does the price. Depending on the nature of the dataset, privacy may be an issue since it involves sharing content with external people.

Semi-automatic Approaches

Weak supervision is normally performed with small, well-annotated datasets that are used to train a model that would later be employed to annotate a bigger dataset. The quality of the latter would depend on the performance of the model, and therefore it may input noise that would affect the learning process of subsequent models [57].

Classic algorithms such as Support Vector Machines (SVM) and Random Forests (RF), but also state-of-the-art solutions like deep neural networks (DNN), can be used as weak models [60]. In order to maximise the results with the minimun human effort, algorithms may be combined with Active Learning (AL) [61]. AL choose specific instances that are best suited to teach models during the training phase. Commonly, these are the instances that are closer to the decision boundary.

Haldenwang et al. [62] survey different AL selection criteria for tweets that are then fed to a DNN that will label a bigger dataset. Their best result was 0.55 in $f1$-score when labelling 800 instances after training with 100 annotated tweets. However, it drops when labelling bigger datasets.

Helmstetter and Paulheim [63] use a semi-supervised approach in which they constructed a large dataset of more than 400k instances of fake and real news. They labelled them by their source (whether it was known to spread fake news or not), and then enriched datasets with more than 100 additional features that significantly increase classification performance. Despite that this approach is feasible, it still requires to compile a list of sources.

Pohl et al. [64] propose an algorithm called *active online multiple prototype classifier (AOMPC)* that uses active learning to help dealing with crisis in Social Media. They obtained good results, however they measured performance using their own metric, which is not directly comparable to other algorithms.

2.2.2 Platforms

We review below different platforms that can be used to build supervised datasets. It is necessary to notice that these are tools rather than methodologies. They should implement one, or even more, of the methodologies presented above.

Open-Source Platforms

There are a huge amount of open-source platforms dedicated to annotate data in different forms. Their main advantage is that they can be maintained by the community and they may be customised to fit specific needs. In this section, we review a selection of those dedicated to text labelling.

SMART [65] is conceived around the idea of "*helping data scientists and research teams efficiently build labelled training datasets for supervised machine learning tasks*"[1]. To assist the learning process, it relies in active learning. Active learning tries to improve accuracy by 1) letting the model choose the data instances that are more useful to learn from and 2) asking an *oracle* to annotate it so the algorithm can learn.

YEDDA [66] was nominated to ACL 2018 Best Demo Paper[2]. It is a lightweight tool that allows annotation of text spans and multi-annotator analysis and pairwise comparison. However, as of this day, last commit is from July 2019 with suggest that the project is no longer maintained.

[1] https://github.com/RTIInternational/SMART

[2] https://github.com/jiesutd/YEDDA

doccano [67] features team collaboration and the possibility to work with any language. It is designed to deal with NER, Sentiment Analysis, Translation, Text to SQL and also image labelling. It is developed in Django and it exposes an API so it can be easily integrated with already-designed workflows.

PIAF It[3] is based on *doccano* and it is focused on Question Answering (QA) annotation. It is able to score each user so it is possible to detect *trolls* and *bots*.

Despite that these are powerful tools, they require some expertise in order to install and maintain them, and they are not as complete or intuitive as private solutions while still requiring a lot of manual work.

Industry-standard Solutions

Building supervised datasets is a required practice in many situations. It is a time and resource-consuming task, therefore it is possible to resort to platforms or companies dedicated to carefully put together labelled datasets. In this section, we review a selection of enterprise-grade solutions to this process.

Amazon Mechanical Turk Also known as MTurk[4]. Arguably, it is the most popular platform to obtain labelled datasets. It is a crowdsourcing platform, which they claim that *"is a good way to break down a manual, time-consuming project into smaller, more manageable tasks to be completed by distributed workers over the Internet"*[5].

They offer an *efficient, flexible and cheap* solution, where the most important features are (1) integration with the AWS SDK, (2) possibility to fully customise templates, and (3) management and qualification of the workers. However, despite that they offer an API which can be used to perform automatic tasks and reduce the cost of the labelling process, everything needs to be implemented. For non-expert users, this may be an overcomplicated situation.

[3] `https://github.com/etalab/piaf`

[4] `https://www.mturk.com/`

[5] Quoted from their landing page

Lionbridge AI They offer a very popular platform as well as workforce in order to label text, audio, image and video. They claim that their clients have full control of the task, which allows to run quality checks.

They focus on offering good quality and a simple, intuitive yet powerful interface, as well as the capacity to scale the labelling task at any time since they have more than 500 thousand qualified annotators[6].

TagTog This company main selling point is their ability to annotate data automatically in collaboration with humans in the loop. The core of the platform is a NLP toolkit that give clients the ability to perform semisupervised tagging.

It is possible to use a custom ML model or to let the tool learn from your annotations and give you suggestions for new ones. They have ready-to-use models in order to perform tasks, such as Named Entity Recognition (NER), and simpler tools like dictionaries. They acknowledge that their platform has some limitations and it is optimised to work with a small number of tokens per annotation in task without many entities. However, there is no detailed documentation that explains the actual models and architecture they use.

While all of these platforms are very popular and used on a daily basis, they still require loads of work (if used for in-house labelling) or a dedicated budget (if outsourced). Moreover, when analysing opinion in Social Media, automatically-generated datasets would not resort necessarily into genuine results. Our proposal aims to reduce the cost of putting together a supervised dataset for opinion mining in Social Media. It is compatible with most techniques presented above, therefore it can enrich (and be complemented with) them.

2.3 Similarity Semantic Networks

Social Media profiles open the door to similarity-based reasoning in order to extract conclusions to characterise unknown users in the network. Yet, not every label can be propagated using the same similarity measure, therefore it

[6] https://lionbridge.ai/services/data-labeling/

is necessary to use a mechanism that enables high-level similarity reasoning. In the following sections, we introduce the concept of *Similarity Semantic Network* as a tool to infer knowledge by similarity reasoning.

2.3.1 Introduction to Semantic Networks

Semantic Networks are one of the first models proposed for Knowledge Representation, and they have been effectively applied over the years [68, 69]. Semantic Networks represent knowledge with directed labelled graphs, where:

- Vertices are concepts, such as *individuals* or *classes* (that are sets of individuals).
- Edges are semantic relations between concepts, that can be:
  - Hierarchical, such as *instance-of* (an individual is an instance of a class; and *is-a* (a class is a subclass of another class). These are universal relations that are present in all Semantic Networks
  - Domain-specific, such as *is-friend-of* or *has-interacted-with*. These relations are tailored to the specific problem and they only make sense in such context.

Given two concepts, A and B, and a relation $\mathscr{S}$, the simplest semantic network would be:

$$\mathbf{A} \xrightarrow{\mathscr{S}} \mathbf{B} \equiv \mathbf{A}\mathscr{S}\mathbf{B} \tag{2.1}$$

With a third concept C, it is possible to perform reasoning in its simplest form, *inference by inheritance*:

$$\begin{array}{l} \mathbf{A}\ \textit{is-a}\ \mathbf{B}\ \vee \mathbf{A}\ \textit{instance-of}\ \mathbf{B} \\ \mathbf{B}\mathscr{S}\mathbf{C} \\ \hline \therefore \mathbf{A}\mathscr{S}\mathbf{C} \end{array} \tag{2.2}$$

Later, graduations were introduced to obtain Fuzzy Semantic Networks, that have interesting and relevant applications [70, 71]. These models represent knowledge as graded labelled directed graphs. Classes are now defined as fuzzy sets of individuals, and the degree of the universal relation *instance-of* would be the membership function of the corresponding fuzzy set.

$$\mathbf{A} \xrightarrow{\mathscr{S}^{\alpha}} \mathbf{B} \equiv \mathbf{A}\ \mathscr{S}^{\alpha}\ \mathbf{B} \tag{2.3}$$

represent the fuzzy assertion $\mathbf{A}\ \mathscr{S}\ \mathbf{B}$ in α degree.

The *fuzzy inference by inheritance* rule can be defined as a generalisation of the (crisp) *inference by inheritance*, using a function t that is usually a t-norm, such that $t(1, 1) = 1$.

$$\begin{array}{l} \mathbf{A}\ \textit{is-a}^{\alpha}\ \mathbf{B} \vee \mathbf{A}\ \textit{instance-of}^{\,\alpha}\ \mathbf{B} \\ \mathbf{B}\ \mathscr{S}^{\beta}\ \mathbf{C} \\ \hline \therefore \mathbf{A}\ \mathscr{S}^{t(\alpha,\beta)}\ \mathbf{C} \end{array} \tag{2.4}$$

It is possible to obtain, after applying any reasoning method, the same semantic relation between two given concepts but with different degrees. It would be necessary to combine both assertions using the *combining inference* rule:

$$\begin{array}{l} \mathbf{A}\ \mathscr{S}^{\alpha}\ \mathbf{B} \\ \mathbf{A}\ \mathscr{S}^{\beta}\ \mathbf{B} \\ \hline \therefore \mathbf{A}\ \mathscr{S}^{g(\alpha,\beta)}\ \mathbf{B} \end{array} \tag{2.5}$$

where g is an aggregation function, normally extrema functions.

2.3.2 Similarity Semantic Relations

Fuzzy Semantic Networks opens a wide range of possibilities. In fact, it is possible (and effective) to use reasoning by similarity in fuzzy systems [72]. Similarity semantic relations are fuzzy semantic relations that represent that two individuals or classes are similar with respect to an aspect:

$$\mathbf{A}\ \textit{is-similar-wrt-}D^{\alpha}\ \mathbf{B}, \tag{2.6}$$

where D is any topic or aspect, and it represents the assertion A and B are similar with respect to the topic D in α degree. Additionally, for every sense D, each concept will have a fuzzy neighbourhood of similar concepts in sense D.

However, the reasoning method for this kind of network needs to take into account that *is-similar-wrt-D* may be only propagated using semantic relations that are related to the aspect D. Therefore, it is necessary to introduce *meta relations*.

Meta relations are *relations between relations*, thus they can be considered *second order relations*.

$$\mathscr{S} \textit{ is-related-to}^{\gamma} \ \mathbf{D}, \tag{2.7}$$

that stands for $\mathscr{S}$ is related to the aspect D with degree γ and thus it can be propagated by *is-similar-wrt-D* relation. The semantic similarity relation specifies a correspondence between concepts in the context of a specific aspect D, while *is-related-to* delimits the domain in which similarity relations apply.

Since these are second order relations, it can be said that, in fact, we are defining a new higher level semantic network, whose individuals are similarity relations of the main semantic network. The new relations enable a new kind of reasoning based on similarity. Knowledge can be extracted upon propagation of semantic relations through the *is-similar-wrt-D* using the *similarity inference rule*.

$$\begin{array}{l} \mathbf{A} \textit{ is-similar-wrt-}D^{\alpha} \ \mathbf{B} \\ \mathbf{B} \ \mathscr{S}^{\beta} \ \mathbf{C} \\ \hline \therefore \mathbf{A} \ \mathscr{S}^{\gamma * t(\alpha,\beta)} \ \mathbf{C} \end{array} \tag{2.8}$$

where t is a *triangular norm* (t-norm).

The property values can then be deduced by fuzzy inheritance and/or by similarity inference. The inference strategy requires choosing between the classical Z (first similarity, then inheritance) or N (first inheritance, then similarity inference) models. Moreover, this kind of systems is prone to be iterative. In each step, the degree of every semantic relation is updated by applying inheritance and similarity reasoning rules (in the chosen order), and then applying the combining inference rule to solve graduation conflicts.

2.3.3 Similarity Semantic Network in SNS

In this section, we propose an example of how Similarity Semantic Relation can be materialised to work with SNS.

SNS consist of three major components [73]:

- User profiles.
- A list of connected profiles.
- A set of possible interactions between users.

Users can (1) define their profile on the Social Network as a set of attributes and content shared (publicly or private), (2) add or remove connections to other users, (3) interact with other user's list of connections. Additionally, they can (4) interact with other people's profiles (including their shared content).

However, it is necessary to distinguish between the part of the profile that users can edit to fit their interest (attributes such as name, biography details, location, profile picture), which we will refer to as *account details*; and the part composed by their shared content, connections and interactions. Thus, it is precise to say that the concept of user profile has a bigger scope than the *account details*, despite that they are used as synonyms.

Let $\mathbb{U}$ be a set of users and $\mathbb{P}$ a set of possible attributes. We will define *user profile* as a set A of triplets (u, P, v), where $u \in \mathbb{U}$, $P \in \mathbb{P}$ and v is a specific value in the domain of $\mathbb{P}$. Frequently, triplets are stored as *rating matrices* $\mathbb{R}^{|U| \times |A|}$, which enables efficient algebraic operations [74].

The list of properties $\mathbb{P}$ may be edited by the user (in the case of *account details*) and calculated (number of interactions per day, number of followers, mean length of the posts...) or estimated through their content and interactions (affinity to other users, affinity to topics, literacy skills...). There are user profile attributes that depend on other properties, that require an analysis of the interactions between users or that are inherently imprecise (fuzzy). We will refer to them as *complex* or *deep* properties (non-superficial, hidden at first sight) [75].

Let $\mathbb{M}$ be a set of messages authored by a subset of $\mathbb{U}$, and $\mathbb{T}$ a set of topics that we are interested in studying. We will define the basic mechanics of SNS as:

$$\forall u, v \in \mathbb{U}, \mathit{follows}(u, v) = \begin{cases} \mathit{true} & \text{if } u \text{ is subscribed to } v \text{ updates} \\ \mathit{false} & \text{in any other case} \end{cases} \tag{2.9}$$

$$\forall u \in \mathbb{U}, \forall m \in \mathbb{M}, \mathit{author}(u, m) = \begin{cases} \mathit{true} & \text{if } u \text{ is the author of } m \\ \mathit{false} & \text{in any other case} \end{cases} \tag{2.10}$$

$$\forall u \in \mathbb{U}, \forall m \in \mathbb{M}, \mathit{favourite}(u, m) = \begin{cases} \mathit{true} & \text{if } u \text{ likes the message } m \\ \mathit{false} & \text{in any other case} \end{cases} \tag{2.11}$$

$$\forall m \in \mathbb{M}, \forall u \in \mathbb{U}, \mathit{mention}(m, u) = \begin{cases} \mathit{true} & \text{if } m \text{ names user } u \\ \mathit{false} & \text{in any other case} \end{cases} \tag{2.12}$$

$$\forall m, n \in \mathbb{M}, \mathit{copy}(m, n) = \begin{cases} \mathit{true} & \text{if } m \text{ is a verbatim copy of } n \\ \mathit{false} & \text{in any other case} \end{cases} \tag{2.13}$$

$$\forall m, n \in \mathbb{M}, \mathit{reply}(m, n) = \begin{cases} \mathit{true} & \text{if } m \text{ is an answer to } n \\ \mathit{false} & \text{in any other case} \end{cases} \tag{2.14}$$

such that

$$u \xrightarrow{\mathit{follows}} v \equiv u \; \mathit{follows} \; v \tag{2.15}$$

$$u \xrightarrow{\mathit{author}} m \equiv u \; \mathit{author} \; m \tag{2.16}$$

$$u \xrightarrow{\mathit{favourite}} m \equiv u \; \mathit{favourite} \; m \tag{2.17}$$

$$m \xrightarrow{\mathit{mention}} m \equiv u \; \mathit{author} \; m \tag{2.18}$$

$$m \xrightarrow{\mathit{copy}} n \equiv m \; \mathit{copy} \; n \tag{2.19}$$

$$m \xrightarrow{\mathit{reply}} n \equiv m \; \mathit{reply} \; n \tag{2.20}$$

These cover the basic mechanics between *users*, between *users* and *posts*, and between *posts* and *posts*. However, until now, we have not payed any attention to the content of the posts.

Analysing message content can determine non-explicit features of the text, particularly if we combine it with the previous reasoning mechanism. In order to illustrate this proposal, we are going to consider the *topic* of the post as well as the *sentiment* towards that topic.

1. $topics(m)$ stands for all the topics discussed in the post m, including the relevant aspect D.

2. $sentiment_D(m)$ stands for the sentiment that the message m presents with respect to the aspect D that we are studying.

There are a number of algorithms that can be applied in order to extract topics (e.g. [76, 77, 78]) and sentiment (e.g. [79, 80, 81]) from messages. Obviously, it is possible to use human oracles. In all cases, the chosen method is independent from the theoretical development, as long as they satisfy the following requirements:

1. For any given message m, if n is a copy of m, then they have the same sentiment score.

$$\forall m, n \in \mathbb{M}, n \textit{ copy } m \Rightarrow \textit{sentiment}(m) = \textit{sentiment}(n) \tag{2.21}$$

2. For any given message m, let n be a copy of m. Then, they need to refer to the same topics.

$$\forall m, n \in \mathbb{M}, n \textit{ copy } m \Rightarrow \textit{topics}(m) = \textit{topics}(n) \tag{2.22}$$

3. For any given message m, if n is a response to m, then they need to share at least one topic.

$$\forall m, n \in \mathbb{M}, n \textit{ reply } m \Rightarrow \textit{topics}(m) \cap \textit{topics}(n) \neq \emptyset \tag{2.23}$$

Therefore, the set of potential interests of any given user $u \in \mathbb{U}$ would be defined by:

$$\begin{aligned} Topics_u = \{t \in \mathbb{T} : [& \\ & \exists m \in \mathbb{M} : u \ \textit{author} \ m \wedge t \in \textit{topics}(m)) \\ & \vee \exists m, n \in \mathbb{M} : (u \ \textit{author} \ m \wedge m \ \textit{copy} \ n \wedge t \in \textit{topics}(n)) \\ & \vee \exists m, n \in \mathbb{M} : (u \ \textit{author} \ m \wedge m \ \textit{reply} \ n \wedge t \in \textit{topics}(m) \cup \textit{topics}(n))]\} \end{aligned} \tag{2.24}$$

Equation 2.24 is a crisp set that includes all the topics that:

1. the user has explicitly mentioned in their messages
2. the user has mentioned by republishing someone's publication
3. the user has referred to by replying to someone's publication.

It is worth clarifying that it is possible that $\textit{topics}(m) \cup \textit{topics}(n)$ includes noisy topics to the set of potential interest. Therefore, the set $Topics_u$ acts as an upper bound, and we will need to deal with these noisy elements in next steps.

Interest is an inherently imprecise concept, since user may present more interest towards certain topics than others. Thus, it is a straightforward conclusion that this set needs to be modelled as a fuzzy one. Let us consider the following function that yields the number of times that the user u has written about the topic t, directly or through a response to another user's post.

$$\begin{aligned} \textit{interest}(u, t) = |\{m \in \mathbb{M} : u \ \textit{author} \ m \wedge \big[& t \in \textit{topics}(m) \\ & \vee \exists n \in \mathbb{M} : (m \ \textit{reply} \ n \wedge t \in \textit{topics}(n))\big]\}| \end{aligned} \tag{2.25}$$

Notice that *copies* are already included in authored messages. Consecutively, we can consider a normalised interest function as a membership function for our fuzzy interest set.

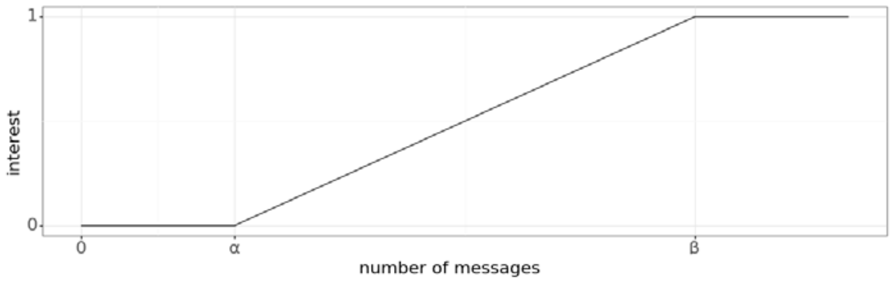


Fig. 2.1.: Behaviour of the membership function (α, β)-$interest$.

$$(\alpha, \beta)\text{-}interest(u,t) = \begin{cases} 0, & \text{if} \quad \textit{interest}(u,t) <= \alpha \\ \frac{x-\alpha}{\beta-\alpha}, & \text{if} \quad \alpha < \textit{interest}(u,t) < \beta \\ 1, & \text{if} \quad \beta <= \textit{interest}(u,t) \end{cases} \tag{2.26}$$

where α and β are context-dependant parameters (figure 2.1). Consequently, it is possible to define the following meta-relation:

$$(\alpha, \beta)\text{-}\textit{interest is-related-to is-similar-wrt-interest} \tag{2.27}$$

which can be used to define an inference mechanism in a similarity semantic network.

In the same fashion that *interest*, modelling user opinion towards a topic requires a mechanism to measure sentiment an a normalisation function. Given a set of linguistic labels $L = \{PP, P, Z, N, NN\}$ (very positive, positive, neutral, negative, very negative) to measure message sentiment, it is possible to determine the opinion of a user u towards a topic t such as:

$$\begin{aligned} sentiment(u,t,l) = |\{m \in \mathbb{M} \wedge u \textit{ author } m \wedge \\ \wedge\, t \in topics(m) \wedge sentiment(m) \textit{ is } l\}| \end{aligned} \tag{2.28}$$

whose analogue membership function would be:

Tab. 2.1.: Twitter mechanics and their implications.

	Common Interest	Common Opinion
Mention	Yes	Not necessarily
Reply	Yes	Not necessarily
Favourite	Yes	Yes
Retweet	Yes	Yes
Quote	Yes	Not necessarily

$$(\alpha,\beta)\text{-}sentiment(u,t,l) = \begin{cases} 0, & \text{if} \quad sentiment(u,t,l) <= \alpha \\ \frac{x-\alpha}{\beta-\alpha}, & \text{if} \quad \alpha < sentiment(u,t,l) < \beta \\ 1, & \text{if} \quad \beta <= sentiment(u,t,l) \end{cases} \tag{2.29}$$

The $|L|$-dimensional vector that results from computing (α,β)-*sentiment* for every user and topic would encode their positions (profile) with respect to the topic in question. As a consequence, it is possible to establish the meta-relation

$$(\alpha,\beta)\text{-}\textit{sentiment is-related-to is-similar-wrt-opinion} \tag{2.30}$$

which is useful to ensemble a reasoning mechanism in a similarity semantic network related to user opinion.

2.4 Deep Relations: a measure of common interest and opinion

We established Similarity Semantic Networks as tool to perform reasoning by similarity to deal with unknown users in SNS. In this section, we define the particulars on our proposal to implement the semantic relations *is-similar-wrt-interest* and *is-similar-wrt-opinion*.

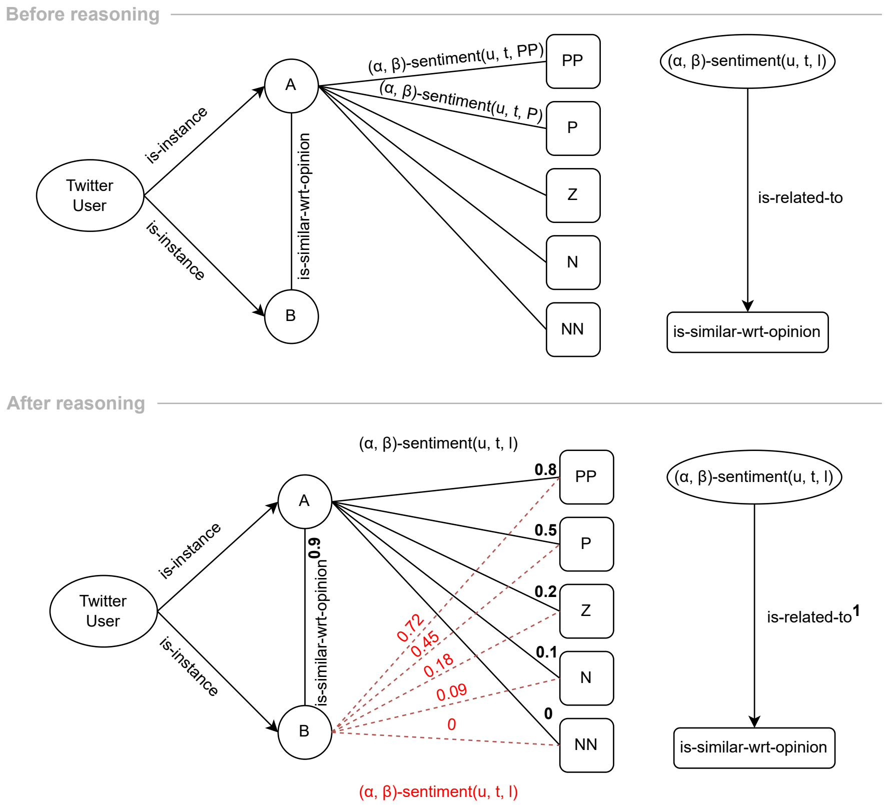


Fig. 2.2.: Example of Similarity Semantic Network for opinion towards a topic, before and after reasoning. We removed some graduations and redundancies for simplicity.

Table 2.1 sums up the implications of SNS mechanics. *Deep relations* are built upon basic ones (see equations 2.15 *et seq.*) and they are intended to model those aspects and/or interactions that are hidden or unobvious. We propose the use of two *deep relations*: *common interest* and *common opinions*.

Definition 2.4.1 (Deep Relation: common interest)**.** *Given two users u and v, we define $commoninterest(u, v) \equiv u$ common-interest v as the accumulated minimum grade in which both users share interest towards specific topics:*

$$\textit{commoninterest}(u, v) = \sum_{t \in C_T} \min \{(\alpha, \beta)\textit{-interest}(u, t), (\alpha, \beta)\textit{-interest}(v, t)\} \tag{2.31}$$

Definition 2.4.2 (Deep Relation: common opinion)**.** *Given two users u and v, we define $commonopinion(u, v) \equiv u$ common-opinion v as the accumulated minimum grade in which both users share opinion towards specific topics:*

$$\textit{commonopinion}(u, v) = \sum_{t \in C_T} \sum_{l \in L} \min \{(\alpha, \beta)\textit{-sentiment}(u, t, l), (\alpha, \beta)\textit{-sentiment}(v, t, l)\} \tag{2.32}$$

where $C_T = Topics_u \cap Topics_v$.

Although any combining functions can be used rather than $\min$, the minima accurately represents the degree in which both users share their interests (or opinions), as it should not be possible to have a common degree that is larger than any of the individual degrees.

It is possible to model the semantic relations *is-similar-wrt-interest* and *is-similar-wrt-opinion* using the *deep relations* presented in equations 2.31 and 2.32. Notice that computing the exact values of *common interest* or *common opinion* is costly, and it requires trained models for topic and sentiment extraction. Fortunately, we can use the semantic implications of SNS interactions to approximate the magnitude of the *deep relations*.

Definition 2.4.3 (Co-copies or co-retweet)**.** *Given two users u and v, we define $cocopies(u, v) \equiv u$ cocopies v as the the number of times that both users have retweeted the same message:*

$$\begin{multline} cocopies(u, v) = \left|\{x : x \in \mathbb{M} \wedge \exists m, n \in \mathbb{M} : \big[u \textit{ author } m \wedge v \textit{ author } n \wedge \right. \\ \left. \wedge\, m \textit{ copy } x \wedge n \textit{ copy } x\big]\}\right| \end{multline} \tag{2.33}$$

Definition 2.4.4 (Co-replies). *Given two users u and v, we define $coreplies(u, v) \equiv u$ coreplies v as the number of times that both users have replied to the same message:*

$$coreplies(u, v) = |\{x : x \in \mathbb{M} \land \exists m, n \in \mathbb{M} : \big[u \textit{ author } m \land v \textit{ author } n \land \\ \land m \textit{ reply } x \land n \textit{ reply } x\big]\}| \quad (2.34)$$

Definition 2.4.5 (Co-favourites). *Given two users u and v, we define u cofavourites v as the number of times that both users have marked as favourite the same message:*

$$cofavourites(u, v) = |\{x : x \in \mathbb{M} \land u \textit{ favourite } x \land v \textit{ favourite } x\}| \quad (2.35)$$

Normalised versions of equations 2.36 and 2.37 can be used as implementations of *is-similar-wrt-interest* and *is-similar-wrt-opinion*.

$$H_{\textit{commoninterest}}(u, v) = \gamma coreplies(u, v) + cocopies(u, v) + cofavourites(u, v) \quad (2.36)$$

and, analogously,

$$H_{\textit{commonopinion}}(u, v) = cocopies(u, v) + cofavourites(u, v). \quad (2.37)$$

Co-copies and *co-favourites* imply common interest therefore they can be used as is. However, *co-replies* may be related to other topics since the content of two different replies may present non-common topics (see equation 2.23), therefore it is necessary to add a *damping hyperparameter* γ whose purpose is to reduce the influence of the *co-replies*.

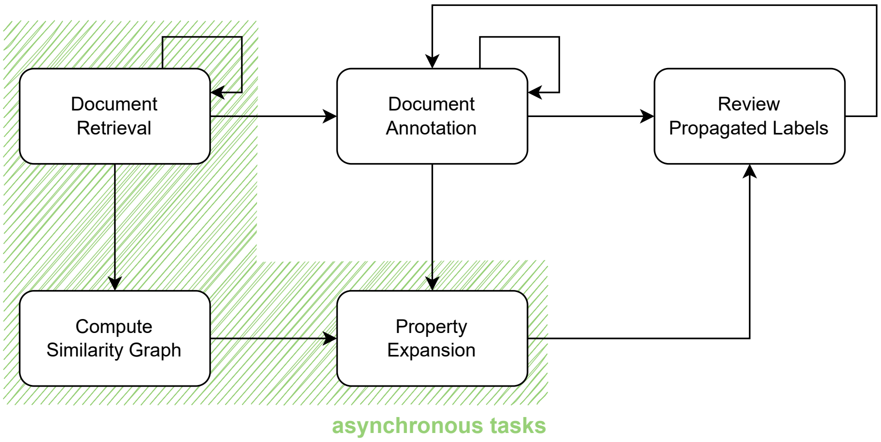


Fig. 2.3.: Proposed methodology

2.5 Proposed Methodology

In this section, we present our methodology to iteratively build a supervised dataset taking advantage of Social Networks mechanics. This methodology is a direct conclusion and a *proof-of-concept* implementation of the Similarity Semantic Network presented before. Keep in mind that there are several other possible implementations.

The target is to build quality dataset while reducing the effort required. In order to do so, we propose the use of a system that would allow us to infer properties of unknown users from other previously annotated documents. The basic workflow would be:

1. Rank tweets by utility.
2. Ask an oracle to annotate the top n tweets.
3. Expand properties to other user profiles using a *deep relation*.
4. Rank automatically-generated user annotations by utility.
5. Ask an oracle to validate the top m auto-annotations.
6. Repeat from step 2 until necessary.

In the following sections, we carefully explain each step in the proposed methodology, as well as the algorithm for equation 2.39.

2.5.1 Determining Properties to Study

The first step consist in determining the user properties that are going to be studied in the dataset. It is necessary to consider whether these properties can be expanded and, in such case, which is the most suitable *deep relation* to use. For example, it is possible to expand the interest towards music using relations co-retweet and co-replies (i.e., any user that replies or retweets a document related to music is likely interested in it); or the opinion that *"COVID19 vaccines are not safe"* using the co-copies relation (i.e. users retweeting a document against vaccines are likely to have a negative opinion about them). In other words, the first step consists in defining the similarity semantic network. Domain expertise is crucial in it.

As a general rule, to deal with opinion mining, we propose the use of co-retweet. In the event that the *interest* in a topic is the subject of the study, it is possible to use co-replies, by itself or combined with co-copies. However, it is possible to define other *deep relations* with different meaning and thus different use cases.

2.5.2 Data Collection

We propose two different mechanisms to collect data, that are:

Uninformed retrieval It only uses *a priori* information to collect data. It is based in static queries designed to match relevant data. It requires establishing an accurate query to reduce the amount of noise (e.g. documents not related to the relevant topic). In the case of synchronous events (such as political debates or natural disasters), related hashtags are specially useful since they filter most of the unrelated topics that have similar wording. Data must be raw or, at least, it must contain the message-message and user-message relations. See section 2.6.1 for details of our implementation.

Tracked retrieval There is another approach based on relevant user tracking. It makes use of a trained machine learning model to determine which documents are prone to be relevant. Upon detection, it queries the API to obtain *retweets* and, in particular, the authors of these. Timelines of found *retweeters* are downloaded to compile an informed dataset. For an in-depth description, please check 2.6.2.

2.5.3 Compute Deep Relation Graph

It is a common practice to use similarity-based graph in tasks such as *community discovery* to extract information from complex networks [82, 83]. In this case, we propose to use a similarity graph to spread known labels to other users.

Similarity-based approaches for community discovery tend to use *retweet* or *mentions* graph. However, these are directed graphs, so the information can only flow in one direction. It is possible to transform basic interactions into non-trivial similarity links by using *deep relations* ([75], see section 2.4).

Interaction mechanics in Social Networking Sites (SNS) have underlying meanings. In the case of Twitter, they may be *replying* (users who reply are interested in the same topic than the original tweet, but they may not have the same opinion), *liking* (users who like explicitly manifest their satisfaction with the content), *mentioning* (which can be virtually used with any purpose), and *retweeting* (that means republishing the tweet in your own timeline. It implies that users are interested in the topic and also that they agree with the content, enough to share it with their friends).

Given a set of users $\mathbb{U}$ and messages $\mathbb{M}$, *co-retweet* or *co-copies* relation is defined as in equation 2.33 (see section 2.4). It calculates the *retweets* that any two users have in common. As we stated above, the *retweet* mechanic implies that the user is interested in the topic and also agrees with the opinion of the author. Therefore, if users A and B have *retweeted* something from C, the *retweet graph* would have two directed links, from A to C and from B to C; with our proposal, both links would be undirected and a new edge would arise between A and B. Note that original tweets are considered *copies (retweets)* of themselves. Consequently, each node in the *co-retweet* graph would stand for a Twitter user, and edges connecting them represent common opinions.

If the dataset is static, similarity neighbourhoods will be computed only once. In the case that it is being created in real time, it will be necessary to include data structures to quickly update graph edges. Algorithm 1 describes the procedure to create a co-retweet graph from a set of tweets. Let $X \subseteq \mathbb{M}$ be a set of tweets, a function *author* : $\mathbb{M} \rightarrow \mathbb{U}$ that yields the author of a given tweet and a function *parent* : $\mathbb{M} \rightarrow \mathbb{M}$ that returns the original tweet:

Algorithm 1 Create co-copies graph

```
Require: list of tweets $X$
Ensure: co-copies graph $G$
Initialize G
parents ← $\{parent(x) : x \in X \land x$ is retweet$\}$
for all $rt \in$ parents do
    neigh ← $\{author(t) : x \in X \land x$ is retweet $\land\ parent(x) = rt\}$
    Append neigh to G.nodes
    Append $\{(author(x), n), \forall n \in$ neigh $: author(x) \neq n\}$ to G.edges
end for
```

The resulting graph G will be conformed of all authors in X connected by undirected weighted edges that stand for the degree in which connected users share opinions pertaining to the topic in question.

2.5.4 Data Annotation

In this step, oracles are asked to set specific values to properties for each document. However, it is not necessary to evaluate every collected document. We rank them using a function that yields the importance of the tweet, in terms of information it may offer. This function will vary with the context but, as a general rule, we recommend the use of the number of retweets when expanding through the co-retweet relation.

Tweets will be presented to oracles (in our case, human annotators) in a descending ranking order. Users with large neighbourhoods will be labelled first, therefore the amount of knowledge that can be inferred is maximal with respect to the number of annotated tweets.

2.5.5 Expansion of Properties

Given a similarity graph (in our case, *co-retweet* graph), it is possible to infer property values for new users in the neighbourhood of known users through a process of weighted expansion. We present our diffusion-based mechanism in this section.

Given a function $Q(m) \in [-1, 1]$ that measures the property Q in the message m, we define the property $Q_{\textit{direct}}$ of a user u as in equation 2.38.

$$Q_{\textit{direct}}(u) = \frac{\sum_{m \in \text{authored}(u)} Q(m)}{|\text{authored}(u)|} \tag{2.38}$$

where *authored*(u) are the documents authored by u. In our case, $Q(m)$ will be a human oracle, but it can be replaced with a trained model or any other mechanism, such as dictionaries. Equation 2.39 defines our proposed expansion mechanism.

$$Q^{(i)}(u) = Q^{(i-1)}(u) + \alpha(N, p_0, p_1)\varphi\left(u, Q_N(u)\right) Q_N(u) \tag{2.39}$$

where $N = |\text{Neigh}(u)|$ and

$$Q_N(u) = \sum_{x \in \text{Neigh}(u)} \frac{Q^{(i-1)}(x)}{N} \tag{2.40}$$

As the expansion is iterative, it is necessary to define $Q^{(0)}(u) = Q_{\textit{direct}}(u)$. α and φ are weight functions that take into account the particulars of the neighbourhood to modify the original property accordingly to relevance (equation 2.41) and confidence (equation 2.42).

$$\alpha(N, p_0, p_1) = \begin{cases} 0 & \text{if} \quad N \leq n_{p_0} \\ \frac{N - n_{p_0}}{n_{p_1} - n_{p_0}} & \text{if} \quad n_{p_0} < N < n_{p_1} \\ 1 & \text{if} \quad n_{p_1} \leq N \end{cases} \tag{2.41}$$

Where n_{p_0} and n_{p_1} stands for percentiles regarding the distribution of neighbours per user: those that are smaller than n_{p_0} are not considered (weight 0, because the neighbourhood is too small to be relevant) meanwhile the ones that have more neighbours than n_{p_1} are considered in full (weight 1, meaning that the neighbourhood is fully relevant). The weight in between follows a linear function as can be seen in figure 2.4.

As for the confidence function, it models the trust in the symbol (negative, neutral or positive) by computing the ratio between neighbours with the same symbol and the total number of neighbours, as described in equation 2.42.

$$\varphi(u, Q_N) = \begin{cases} |\{n \in Neigh(u) : Q^{(i-1)}(n) < 0\}| \div N & \text{if} \quad Q_N < 0 \\ |\{n \in Neigh(u) : Q^{(i-1)}(n) = 0\}| \div N & \text{if} \quad Q_N = 0 \\ |\{n \in Neigh(u) : Q^{(i-1)}(n) > 0\}| \div N & \text{if} \quad Q_N > 0 \end{cases} \tag{2.42}$$

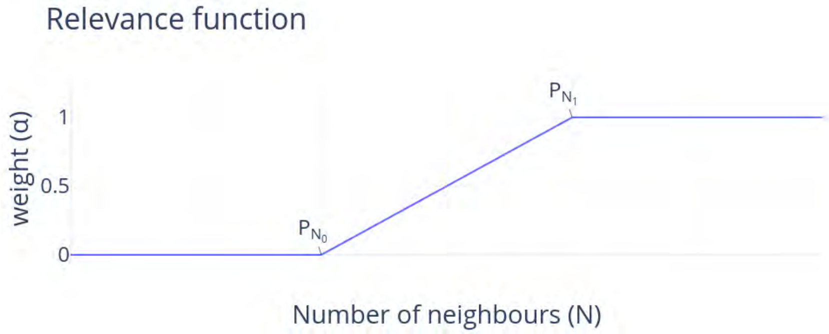


Fig. 2.4.: Neighbourhood relevance function. The function returns a value in the interval $[0, 1]$ that stands for the relevance of the neighbourhood. Values close to 1 represent neighbourhoods with a sufficient number of neighbours to be expanded.

All in all, our iterative expansion mechanism starts with directly computed property values for those users whose properties are known (Q_{direct}) and, for each iteration, it adds the bias of the neighbourhood taking into account their relevance and their coherence.

In order to infer knowledge to new users in the network, we use the approach described in equation 2.39. The process is iterative and it requires several hyperparameters, that are:

1. Steps (i), that refers to the number of iterations in each cycle of expansion. Lower values will lead to more conservative expansions (users in the direct neighbourhood) while higher ones will result in adventurous outcomes.

2. P_0 and P_1, that are percentile cuts used to weight the importance of the neighbourhood (equation 2.42). Predictions of user properties with a low number of neighbours are less accurate than those in which the user has a higher number of them. Hence, the sum of the properties of the neighbourhood is multiplied by a factor that weights its relevance in the context of the network.

These parameters are context-dependant and should be optimised to obtain the maximum performance in each situation. However, as a rule of thumb, we recommend values presented in section 2.8.

Computational Complexity

Property Expansion through a *co-retweet* graph inherently leans to parallelism. The task of building the graph can be decomposed, which is an advantage considering that modern multi-thread architectures may offer significant improvements over the computational time. Moreover, label propagation mechanisms relies on the results from previous iterations, therefore it is possible to split the task into different subgraphs so the implementation would also be parallel, as long as the threads keep synced between generations.

The time complexity of the algorithm depends on (1) the number of steps s of the propagation, (2) the number of labels that are being independently propagated ($p = |L|$), (3) the number of users or nodes $n_U = |\mathbb{U}|$ in the graph, (4) the number of posts or messages $n_M = |\mathbb{M}|$, (5) the number of messages that are retweets n_{rt}, (6) the number of manually annotated documents n_a, and (7) the number of neighbours for the i-th user $R_i = |Neigh(i)|$.

$$\Omega_{\text{alg1}} = \sum_{i=1}^{n_M} \left(\frac{n_{\text{rt}}}{n_M} \sum_{j=i}^{n_M} 1 \right) = \frac{n_{\text{rt}}}{n_M} \sum_{i=1}^{n_M} (n_m - i + 1) = \frac{1}{2} n_{\text{rt}} n_M - 1 \qquad (2.43)$$

$$\Omega_{\text{alg2}} = \sum_{i=1}^{n_a}\sum_{j=1}^{p} 1 + \sum_{i=1}^{s}\sum_{j=1}^{n_U}\sum_{k=1}^{p}\sum_{l=1}^{R_j} 1 = pn_a + \sum_{i=1}^{s}\sum_{j=1}^{n_U} pR_j \simeq pn_a + spn_U n_R \tag{2.44}$$

In the worst (and unlikely) case scenario, all users are connected with each other (clique) therefore $\forall i \in \mathbb{U}, R_i = n_U$, all messages are retweets hence $n_{\text{rt}} = n_M$, and all posts have been manually annotated thus $n_a = n_M$; in the best (and unlikely) case scenario, all users are completely isolated from the rest, therefore $\forall i \in \mathbb{U}, R_i = 0$, there are no retweets at all thus $n_{\text{rt}} = 0$ and there are no manual annotations hence $n_a = 0$; assuming $R \sim \mathcal{N}(n_R, \sigma_R^2)$, in the average case, $R_i = n_R$ and both n_a and n_{rt} are constants.

Graph builder (alg. 1)		**Label propagation (alg. 2)**	
$O(n_M^2)$		$O(pn_M + spn_U^2)$	$\sim O(n_U^2)$
$o(n_M)$		$o(pn_a + n_U)$	$\sim o(n_U)$
$\Theta(n_M + n_{rt}^2)$	$\sim \Theta(n_{rt}^2)$	$\Theta(pn_a + spn_U n_R)$	$\sim \Theta(n_U n_R)$

Despite that we have considered the entire neighbourhood, it can be limited to the most similar k neighbours, therefore the average cost of the expansion would be linear. In any case, it is worth mentioning that the main purpose is to reduce the human effort required to obtain a quality dataset, hence the only limitation would be that the expansion is done before oracles finish their tasks.

2.5.6 Validation of Automatic Labels

Automatic generated labels ought to be validated by an oracle to ensure that the inference is working properly. In essence, there are different type of situations we may encounter when revising propagated labels:

1. Users whose tweets are all retweets. As their properties would depend on the properties of the tweets, there would not be any additional information, therefore the labels can be confirmed automatically.

2. Users whose inferred properties present any contradiction. In the event that two properties p and q are exclusive (that is, that one can only occur if the other does not), if they happen to appear together, the expanded properties can be rejected automatically. This is a great filter since there may be many cases in which automatic rules can be applied.

3. Users that have published their own original content and that do not present any contradiction. These should be consulted to an oracle. All their collected tweets would be presented jointly to the oracle, and they will decide on the the correctness of the predictions. Workers will be asked several questions (see figure 2.5):

    a) Is there any evidence that at least one property is incorrect?

    b) Is there any evidence that at least one property is correct?

    c) Is there any contradiction?

Tweets involved in rejected predictions will increase their ranking to be prioritised in the annotation process, so the mistake can be sorted out as soon as possible.

2.5.7 Loop

This procedure may run iteratively, looping through the steps described in sections 2.5.4, 2.5.5 and 2.5.6; or through steps 2.5.2, 2.5.3, 2.5.4, 2.5.5 and 2.5.6 in the event of a real-time analysis.

It is important to balance the workload between each process, so (1) the transfer of knowledge gets reinforced by manual labelling, to (2) rapidly increase the number of known users through automatic labelling and to (3) improve the accuracy of the automatic labelling validating results for potentially conflicting users.

The balancing could be attained by distributing the workforce (e.g., five oracles labelling documents and ten more reviewing automatic user annotations, with expansions every day) or by timeframes (e.g., one day labelling documents and two days reviewing automatic annotations, with expansions every three days). This would vary depending on the actual task and human resources.

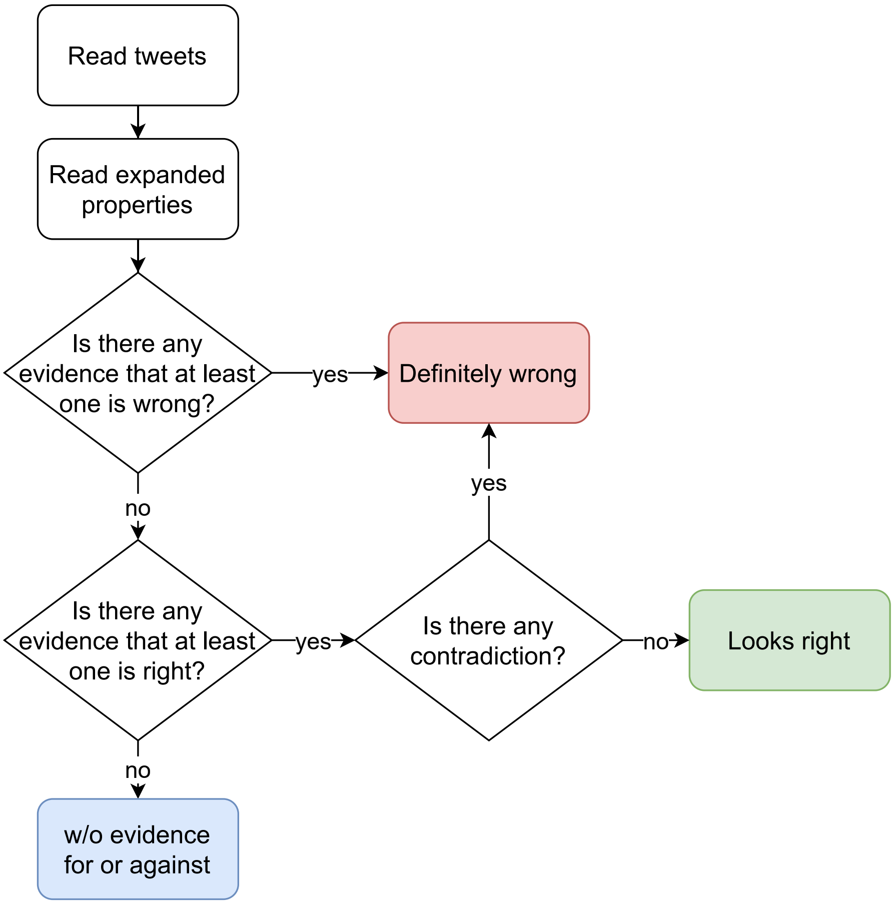

Read tweets
Read expanded properties
Is there any evidence that at least one is wrong?
yes
Definitely wrong
no
yes
Is there any evidence that at least one is right?
yes
Is there any contradiction?
no
Looks right
no
w/o evidence for or against

Fig. 2.5.: Decision tree that leads to the acceptance or rejection of each automatic annotation based on the presented evidence.

Algorithm 2 Property expansion (an implementation of equation 2.39)

```
Require: graph $G$, list of properties to expand props, steps, percentile $p_0$ and $p_1$, weight function
    alpha
Ensure: expanded properties are created
    ecount $\leftarrow$ set of edge count per node
    $np0 \leftarrow$ percentile $n_{p_0}$ of ecount
    $np1 \leftarrow$ percentile $n_{p_1}$ of ecount
    manns $\leftarrow$ load manual annotations of tweets
    $q_d \leftarrow$ init dictionary of user$\rightarrow$props$\rightarrow$value
    for all ann $\in$ manns do
        tcount $\leftarrow$ count number of tweets of the annotation's tweet author
        for all p $\in$ props do
            $q_d$[ann.tweet.author][p] $+= \frac{\text{ann.properties[p]}}{\text{tcount}}$
        end for
    end for
    $q_{ext} \leftarrow q_d$
    users $\leftarrow \{u : u \in$ keys of $q_d\}$
    for $i \leftarrow 0$ to steps do
        init $q_{ext}$_next to save results for next iteration
        for all $u \in$ users do
            neigh $\leftarrow$ get $G$ edges for user $u$
            for all $p \in$ props do
                for all $n \in$ neigh do
                    if $p \in$ keys of $q_{ext}[n]$ then
                        sum $+= q_{ext}[n][p]$
                        poscount $+= 1$ if $q_{ext}[n][p] > 0$
                        negcount $+= 1$ if $q_{ext}[n][p] < 0$
                    end if
                    if sum $> 0$ then
                        confidence $\leftarrow \frac{\text{poscount}}{|\text{neigh}|}$
                    else
                        if sum $< 0$ then
                            confidence $\leftarrow \frac{\text{negcount}}{|\text{neigh}|}$
                        else
                            confidence $\leftarrow \frac{|\text{neigh}|-\text{poscount}-\text{negcount}}{|\text{neigh}|}$
                        end if
                    end if
                    $q_{ext}$_next$[u][p] \leftarrow q_{ext}[u][p] * alpha(|\text{neigh}|, n_{p_0}, n_{p_1}) * confidence * sum$
                end for
            end for
        end for
        $q_{ext} \leftarrow q_{ext}$_next
    end for
```

2.6 The Nutracker Platform

The Nutcracker platform is a *proof-of-concept* implementation of the methodology proposed in section 2.5. It consist on several interconnected microservices that cover all the basis of the proposed methodology, from *document retrieval* to *validation of the expansion*, including *co-retweet graph maintainance*, *document ranking*, *annotation* and *property expansion*.

There are modules that act as workers and that can be scaled horizontally, if required. We expose an API that enables the interaction with the system, currently from a web application. This website is also responsible for the annotation and validation processes, as it constitutes the interface between the human oracles and the system.

To implement the platform, we used `python 3.9.6` ecosystem with `Flask 2.0.1`, `Flask-JWT-Extended 4.0.2`, `pyahocorasick 1.4.1`, `marshmallow 3.13.0` and its Flask bindings, `SQLAlchemy 1.3.23`, and `MariaDB` relational databases, for the backend; `redis 3.5.3` with `rq 1.7.0`, `tweepy 4.6.0`, `pandas 1.3.2`, `networkx 2.6.2`, `numpy 1.20.3`, `scipy 1.7.1`, `sklearn 1.0.2`, `plotly 4.14.3`, for the asynchronous computational nodes; and `HTML5`, `jQuery 3.4.1`, and `Bootstrap v4` for the frontend.

Platform's functional requirements are:

1. System must have restricted access credentials.
2. System must show a document that needs to be annotated.
3. System must show a series of related documents.
4. System must show a list of suggestions/insight regarding the current document.
5. System must show a list of labels and their correspondent values.
6. System must allow to arbitrarily add new labels (*folksonomy*).
7. System must offer a comment box for the current document.
8. System must allow reviewing previous annotations.
9. System must allow reviewing automatically-made annotations.
10. System must be able to execute long-running asynchronous tasks.
11. System must not reveal sensible information.

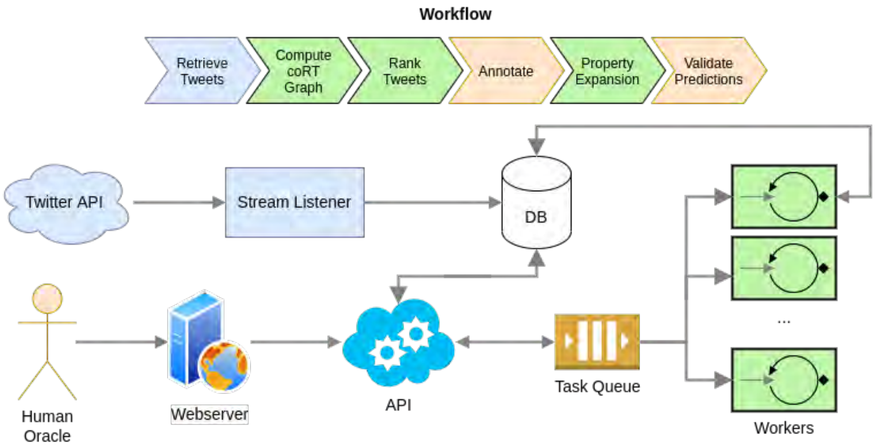


Fig. 2.6.: Diagram of the labelling and expansion platform. Workflow colours indicate the node that is responsible for each part of the process.

Analogously, non-functional requirements are:

- System may offer coloured categories for the labels.
- System may offer keyboard navigation for productivity reasons.
- System should have redundant buttons for convenience.
- System should allow to hide suggestions/insight that may bias the annotator.
- System should allow text search within the documents.
- System should offer statistics regarding the annotation process.
- System should allow video annotation of related events.

Figure 2.6 shows the general architecture of the platform. In the following sections, we revise each one of the nodes that are responsible for correct operation of the Nutcracker tool as well as their implementation.

2.6.1 Uninformed Retrieval

When working with external services like Social Media, the standard practise is to use their public APIs to interact (programatically) with the core elements of the service. Normally, each API would offer different access levels with their respective limitations, being the free level the most limited one[7].

There are two ways to retrieve data using Twitter API. The first one consist in static queries that allow retrieval of up to 500k tweets per month (other limitations apply[8]); the second one consist in streaming pipes that allow for continued retrieval of tweets in real time, up to 1% of the tweets that are being published at that very moment.

Both methods rely on *queries* that are limited to 512 characters and that can be altered a limited number of times. This restriction limits the adaptability of our tool to the continuously-changing environment of a Social Network, and it is mostly used to retrieve well-characterised tweets that have already been published; or for the synchronous download of real-time event-related content. One example of query could be "#PresidentialDebates", that was the hashtag used in the 2020 US Presidential Elections; it is also possible to target users, such as "@sanidadgob"; or to run a query by keywords, such as "*incendio granada*"; or even advanced queries such as "`andalucia has:links has:media`", which would result in tweets mentioning *Andalucía* and including links and media.

However, there are two points to consider when retrieving Twitter data. On the one hand, Twitter API may present several sampling biases. On the other hand, query limitations prevent manual targeting of users and topics since they can not be altered indefinitely. Therefore, it would be necessary to filter those users that do not present any evidence of relevance and to direct the search towards clusters of interests (see section 2.6.2 🏷).

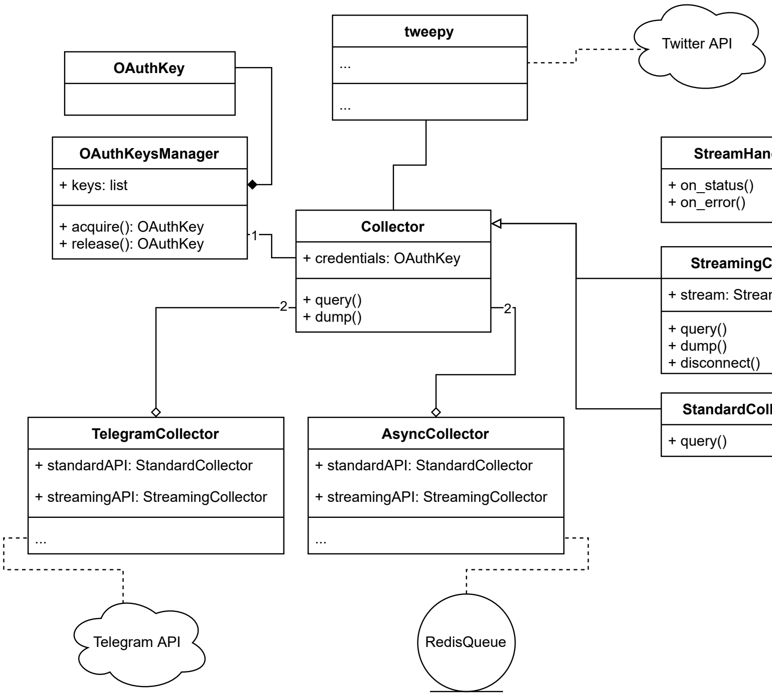


Fig. 2.7.: Partial UML diagram of Tweet Retrieval Microservice

Implementation

The uniformed retrieval node is a microservice that uses *tweepy* to query Twitter API. It is a multi-threaded consumer that receives queries from the user (using *redis* or a *Telegram Bot*), builds a request and listens for the API response (please refer to figure 2.7, page 57).

Retrieved tweets are raw dumped to a file (in order to save all the fields of the Status object, in case they are necessary in the future) and then serialised by a *marshmallow model* to save all relevant fields in a database, that is connected to the rest of the components of the Nutcracker platform. The component that translate from the Status object to the database model can be easily

[7] When discussing API limitations and unless specified otherwise, we will refer to Free Level Access quotas

[8] `https://developer.twitter.com/en/docs/twitter-api/getting-started/about-twitter-api`

override to adapt the tool to other data sources. Static queries finish when no more tweets match the query or when the user-specified limit is reached; streaming ones only finish when the proper stop command is sent.

2.6.2 Tracked Retrieval

One of the limitations of the *Uninformed Retrieval module* (see section 2.6.1) is that results are influenced by Twitter recommendation system for the owner of the API key. In order to retrieve information from specific parts of the network and to target content without taking into account the API key that is being used, we developed the *Tracker* module.

This node starts by performing an initial uninformed query. Then, it retrieves all the possible tweets (with respect to the quota limit and the matching results) and runs a trained Machine Learning (ML) model to decide whether the content is relevant or not. This is a preliminary decision whose only purpose is to act as filter of relevance. For those tweets that are classified as relevant, it generates tasks that are queued with the purpose of retrieving users that have retweet them (i.e., potentially relevant users). Once this information is obtained, the module maintains in real time a *co-retweet* graph, which is used to perform property expansion and to keep updating the rank of relevant users to re-query them.

In the same fashion than *Uninformed Retrieval*, it raw dumps all the tweets and store the relevant information with the necessary abstractions and preprocessing into the database.

Implementation

Since this module performs computationally-heavy tasks, it was developed using a *ventilator-worker-sink scheme* (please refer to figure 2.8 and 2.9, page 60).

- Ventilator, whose main purpose is to generate an initial set of tasks and serve as an interaction point with humans. It can be specialised in:

    - Twitter Account watchdog, which opens a stream that listen the timeline of any given Twitter user and queue retrieval tasks for all reported accounts (useful to watch for bot activity in accounts like @CtrlSec).

    - User interaction agent, which enables a way for Nutcracker users to schedule one-time retrieval tasks or to report and track specific Twitter users.

- Workers, that are hot-spawnable and can scale horizontally. Their main responsibilities are:

    - Perform retrieval-related task using Twitter API (see section 2.5.2, page 44).

    - Preprocess tweets and classify them using a trained ML model that it used to check whether the document is relevant or not.

    - Pass the relevant information to the sink and maintain a dump of all retrieved content.

- Sink, that has several purposes:

    - Maintain the similarity graph, updating nodes and connections between them when new information is received (see section 2.5.3, page 45).

    - Generate re-queries, for those nodes that have not been updated in a certain amount of time.

    - Propagate features, in order to delimit a cluster of potentially relevant users without requiring actual information on them (see section 2.5.5, page 47).

    - Generate lookup queries, to obtain information of those unknown users that are potentially relevant.

There can be multiple *ventilators*, which act as *producers*. They are in charge of creating tasks that can be solved in parallel using *workers*, and they communicate with the latter using a network queue. There can also be multiple *workers*, which purpose is to *consume* tasks by querying Twitter API. Retrieved information is raw dumped then processed to be sent to the *sink* through another network queue. *Sink* would use processed information to keep the graph updated and to generate more retrieval tasks.

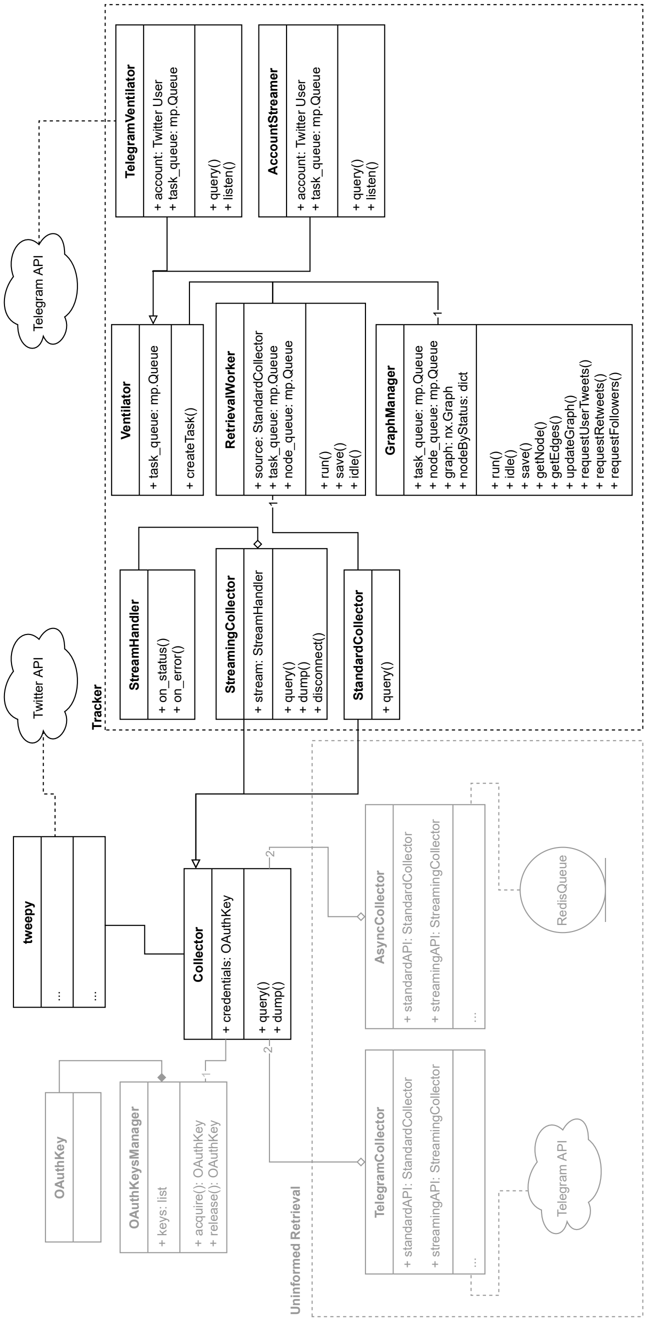


Fig. 2.8.: UML diagram of Tweet Retrieval Microservice. Highlighted elements belong to the tracker module.

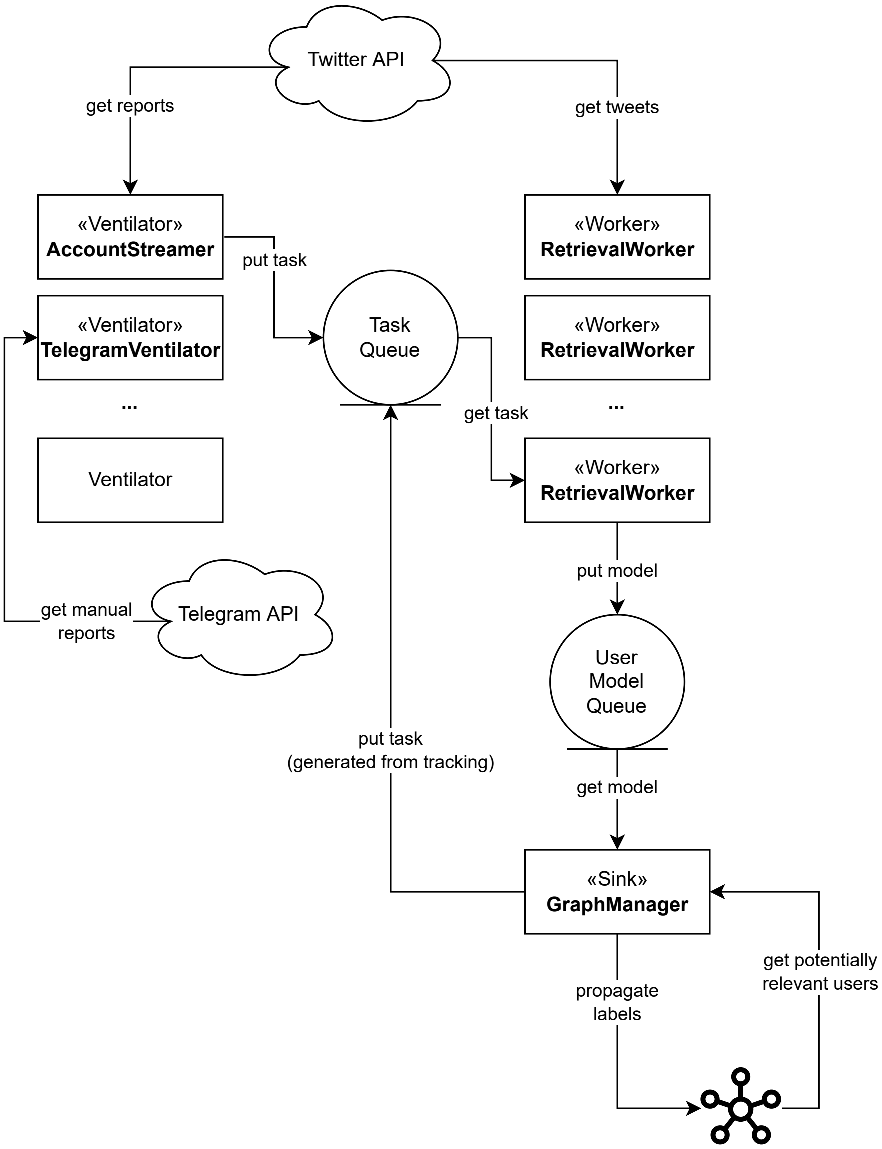
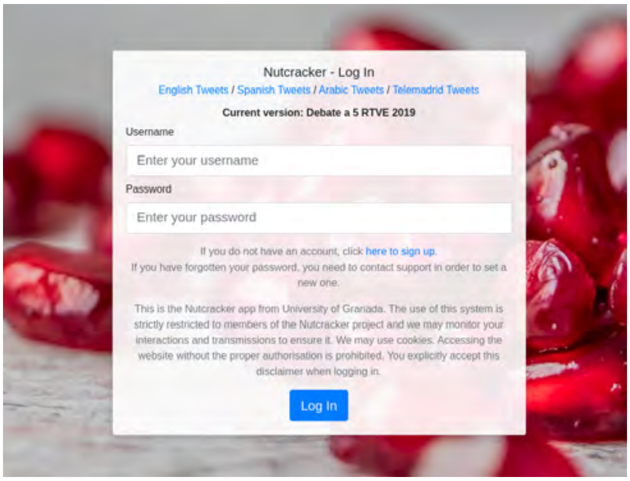


Fig. 2.9.: Basic working scheme of the Tracker component

2.6.3 Web tool

Nutcracker exposes an API (see section 2.6.4, page 67) that enables interaction between different services. The interface that enables human interaction is a web service that implements the annotation procedure.

The site is hosted at `http://nutcracker.ugr.es`. The landing page (please refer to figure 2.10, page 62) is a simple login form with several options that let the user sign up or navigate through the different available versions (datasets).



Nutcracker - Log In

English Tweets / Spanish Tweets / Arabic Tweets / Telemadrid Tweets

Current version: Debate a 5 RTVE 2019

If you do not have an account, click here to sign up.

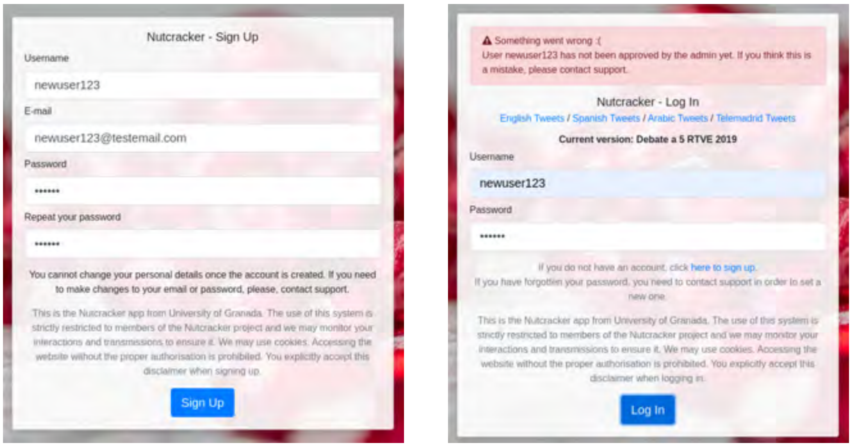
If you have forgotten your password, you need to contact support in order to set a new one.

This is the Nutcracker app from University of Granada. The use of this system is strictly restricted to members of the Nutcracker project and we may monitor your interactions and transmissions to ensure it. We may use cookies. Accessing the website without the proper authorisation is prohibited. You explicitly accept this disclaimer when logging in.

Fig. 2.10.: Log-in form

If the users choose to sign up, they will be asked for username, email and password. The administrator needs to approve the account before they can access the site (see figure 2.11).

Figure 2.12 shows the main page, that is used to annotate tweets. The site asks Nutcracker API for a top-ranked unlabelled tweet, and it shows it highlighted in order for the user to annotate it. Below the main tweet, the user can find other retrieved tweets for the same author that may offer context in order to better evaluate the document in question (however, a disclaimer is included telling the user to disregard any evidence that cannot be found in the original tweet (see figure 2.13)).



Nutcracker - Sign Up

Username

newuser123

E-mail

newuser123@testemail.com

Password

••••••

Repeat your password

••••••

You cannot change your personal details once the account is created. If you need to make changes to your email or password, please, contact support.

This is the Nutcracker app from University of Granada. The use of this system is strictly restricted to members of the Nutcracker project and we may monitor your interactions and transmissions to ensure it. We may use cookies. Accessing the website without the proper authorisation is prohibited. You explicitly accept this disclaimer when signing up.

⚠ Something went wrong :(
User newuser123 has not been approved by the admin yet. If you think this is a mistake, please contact support.

Nutcracker - Log In

English Tweets / Spanish Tweets / Arabic Tweets / Telemadrid Tweets

Current version: Debate a 5 RTVE 2019

Username

newuser123

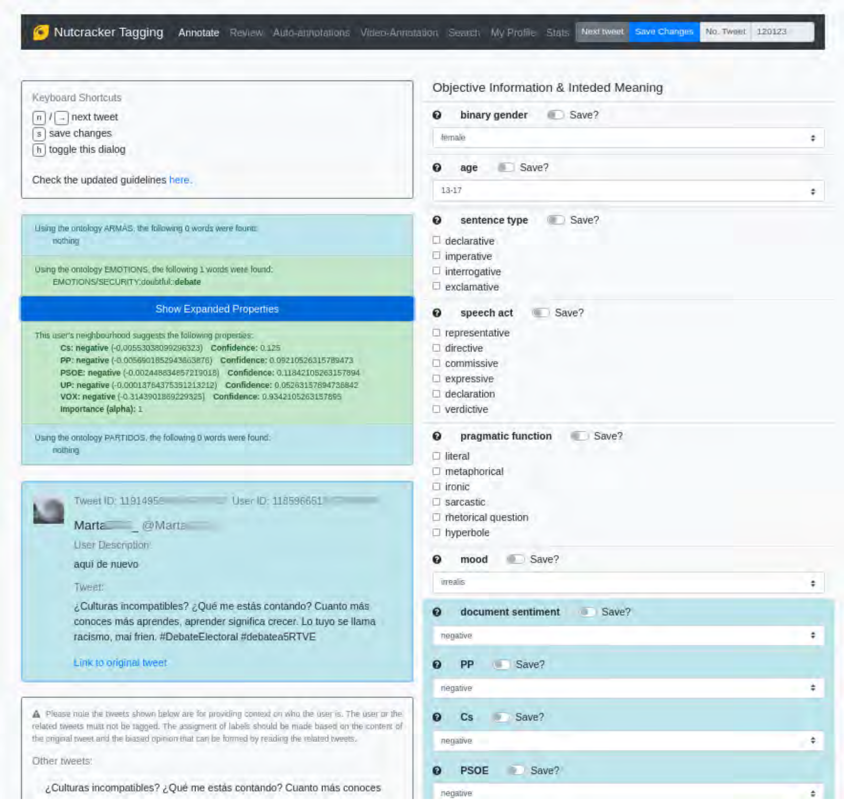
Password

••••••

If you do not have an account, click here to sign up.
If you have forgotten your password, you need to contact support in order to set a new one.

This is the Nutcracker app from University of Granada. The use of this system is strictly restricted to members of the Nutcracker project and we may monitor your interactions and transmissions to ensure it. We may use cookies. Accessing the website without the proper authorisation is prohibited. You explicitly accept this disclaimer when logging in.

Fig. 2.11.: Sign up site and failed log in.



Keyboard Shortcuts

n / → next tweet
s save changes
h toggle this dialog

Check the updated guidelines here.

Using the ontology ARMAS, the following 0 words were found:
nothing

Using the ontology EMOTIONS, the following 1 words were found:
EMOTIONS/SECURITY:doubtful::**debate**

This user's neighbourhood suggests the following properties:
Cs: negative (-0.00553038099296323) **Confidence:** 0.125
PP: negative (-0.0056901852943663876) **Confidence:** 0.09210526315789473
PSOE: negative (-0.002448834857219018) **Confidence:** 0.11842105263157894
UP: negative (-0.00013764375351213212) **Confidence:** 0.05263157894736842
VOX: negative (-0.3143901869229325) **Confidence:** 0.9342105263157895
Importance (alpha): 1

Using the ontology PARTIDOS, the following 0 words were found:
nothing

Tweet ID: 11914959[redacted] User ID: 118596651[redacted]

Marta[redacted]_ @Marta[redacted]

User Description:

aquí de nuevo

Tweet:

¿Culturas incompatibles? ¿Qué me estás contando? Cuanto más conoces más aprendes, aprender significa crecer. Lo tuyo se llama racismo, mai frien. #DebateElectoral #debatea5RTVE

Link to original tweet

⚠ Please note the tweets shown below are for providing context on who the user is. The user or the related tweets must not be tagged. The assignment of labels should be made based on the content of the original tweet and the biased opinion that can be formed by reading the related tweets.

Other tweets:

¿Culturas incompatibles? ¿Qué me estás contando? Cuanto más conoces

Objective Information & Inteded Meaning

binary gender Save?
female

age Save?
13-17

sentence type Save?
☐ declarative
☐ imperative
☐ interrogative
☐ exclamative

speech act Save?
☐ representative
☐ directive
☐ commissive
☐ expressive
☐ declaration
☐ verdictive

pragmatic function Save?
☐ literal
☐ metaphorical
☐ ironic
☐ sarcastic
☐ rhetorical question
☐ hyperbole

mood Save?
irrealis

document sentiment Save?
negative

PP Save?
negative

Cs Save?
negative

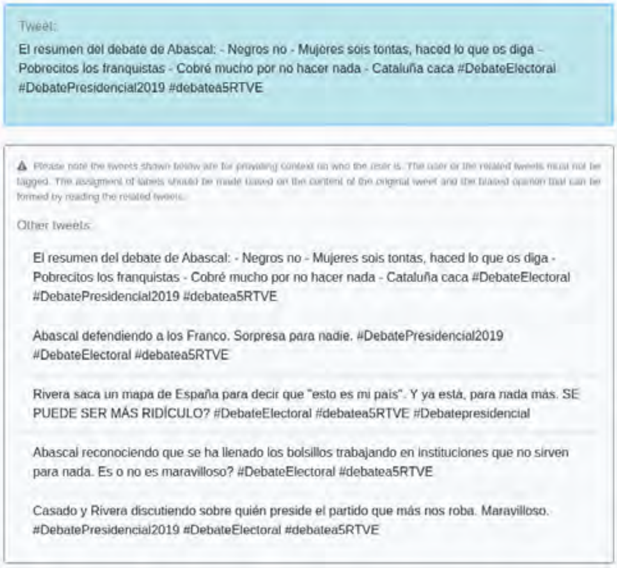
PSOE Save?
negative

Fig. 2.12.: Nutcracker's tweet annotation interface

The desktop version of the site is a 2-column layout in which the first column gives information to the user, and the second one contains the input fields to annotate the tweets. Labels shown in this column may be:

- Single choice questions.
- Multiple answer questions.
- Double labels, subject to the content (explicit vs. implicit).

Labels, label types and values are stored in the database and they may be changed on a per-instance basis in order to adapt the labels to the specific datasets. Double labels are intended to model the differences between what it is explicitly said (denoted) and what is implicitly said (connoted). Although models are normally trained with *connotations*, this distinction enables new analysis avenues.



Tweet:

El resumen del debate de Abascal: - Negros no - Mujeres sois tontas, haced lo que os diga - Pobrecitos los franquistas - Cobré mucho por no hacer nada - Cataluña caca #DebateElectoral #DebatePresidencial2019 #debatea5RTVE

⚠ Please note the tweets shown below are for providing context on who the user is. The user or the related tweets must not be tagged. The assignment of labels should be made based on the content of the original tweet and the biased opinion that can be formed by reading the related tweets.

Other tweets:

El resumen del debate de Abascal: - Negros no - Mujeres sois tontas, haced lo que os diga - Pobrecitos los franquistas - Cobré mucho por no hacer nada - Cataluña caca #DebateElectoral #DebatePresidencial2019 #debatea5RTVE

Abascal defendiendo a los Franco. Sorpresa para nadie. #DebatePresidencial2019 #DebateElectoral #debatea5RTVE

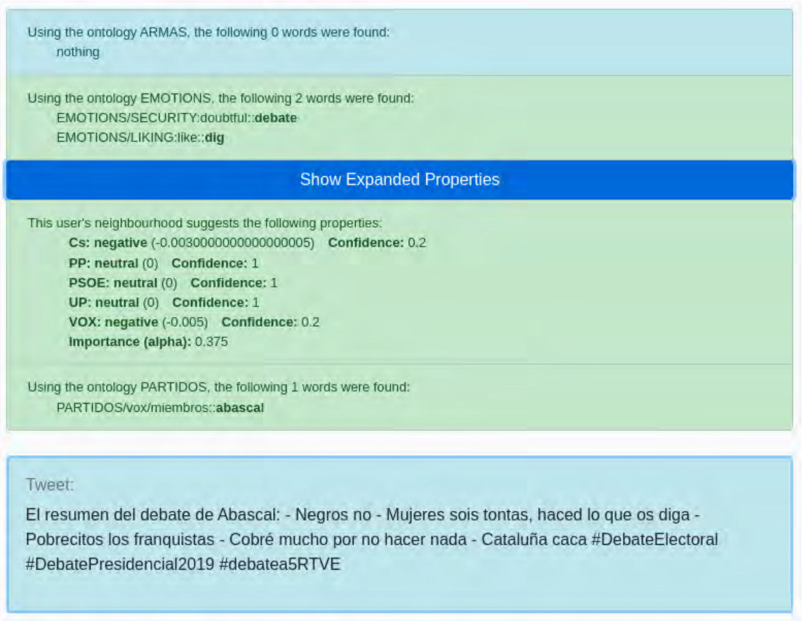
Rivera saca un mapa de España para decir que "esto es mi país". Y ya está, para nada más. SE PUEDE SER MÁS RIDÍCULO? #DebateElectoral #debatea5RTVE #Debatepresidencial

Abascal reconociendo que se ha llenado los bolsillos trabajando en instituciones que no sirven para nada. Es o no es maravilloso? #DebateElectoral #debatea5RTVE

Casado y Rivera discutiendo sobre quién preside el partido que más nos roba. Maravilloso. #DebatePresidencial2019 #DebateElectoral #debatea5RTVE

Fig. 2.13.: Other tweets of the same author are available to offer context.

Above the main tweet, users can find the output of the different *assistants*. Assistants perform an specific task intended to help the user in the annotation process (but they do not alter or change labels). These can be ML models, ontologies or even the result of the expansion process (please refer to figure 2.14 🖼, page 65).



Using the ontology ARMAS, the following 0 words were found:
nothing

Using the ontology EMOTIONS, the following 2 words were found:
EMOTIONS/SECURITY:doubtful::**debate**
EMOTIONS/LIKING:like::**dig**

This user's neighbourhood suggests the following properties:
Cs: negative (-0.0030000000000000005) **Confidence:** 0.2
PP: neutral (0) **Confidence:** 1
PSOE: neutral (0) **Confidence:** 1
UP: neutral (0) **Confidence:** 1
VOX: negative (-0.005) **Confidence:** 0.2
Importance (alpha): 0.375

Using the ontology PARTIDOS, the following 1 words were found:
PARTIDOS/vox/miembros::**abascal**

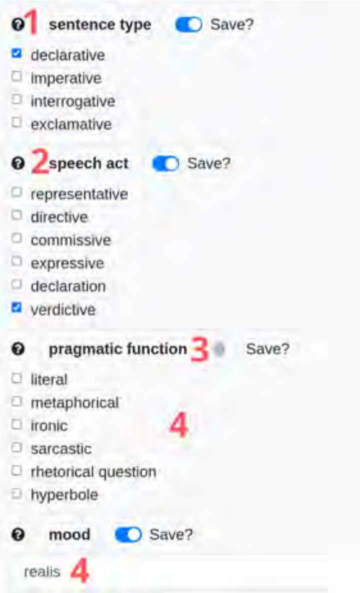
Tweet:

El resumen del debate de Abascal: - Negros no - Mujeres sois tontas, haced lo que os diga - Pobrecitos los franquistas - Cobré mucho por no hacer nada - Cataluña caca #DebateElectoral #DebatePresidencial2019 #debatea5RTVE

Fig. 2.14.: Assistants provide insight to the user.

Once the analysis of the document is finished, the user is required to fill out the labels presented in the second column. Figure 2.15 🖼 shows an example of those labels. The help icon can be used to obtain a detailed description of the label. The *save switch* enables a mechanism to ensure that the user is not unintentionally saving unwanted information.

After the expansion (see section 2.5.5 🏷, page 47) is triggered, another part of the webtool becomes available. The result of the expansion is a batch of automatic user annotations that needs to be reviewed. The *auto-annotation* tab (see figure 2.16 🖼) of the tool offers a different interface in which the user is asked to answer several questions designed to assert if the label is correct or not (please refer to figure 2.5 🖼, page 52).



sentence type — Save?

☑ declarative
□ imperative
□ interrogative
□ exclamative

speech act — Save?

□ representative
□ directive
□ commissive
□ expressive
□ declaration
☑ verdictive

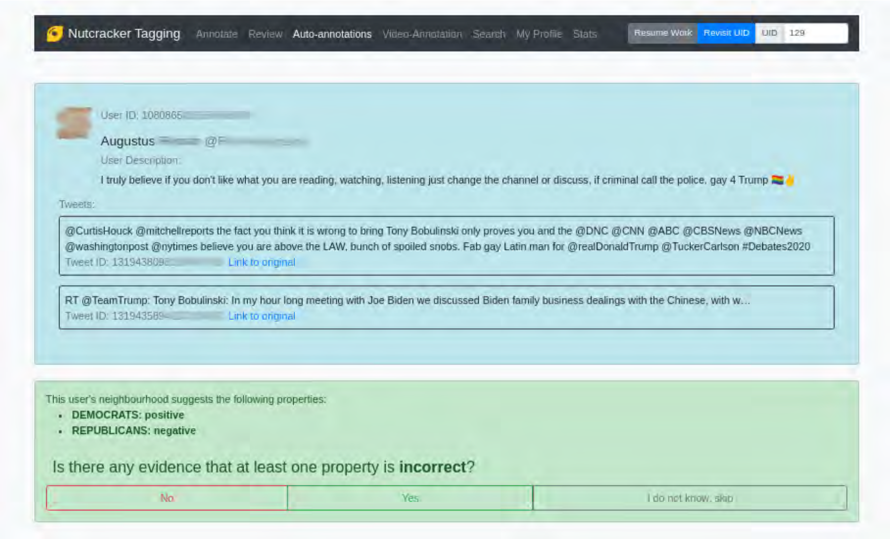
pragmatic function — Save?

□ literal
□ metaphorical
□ ironic
□ sarcastic
□ rhetorical question
□ hyperbole

mood — Save?

realis

Fig. 2.15.: A subset of labels. The first element is the help icon, that can be clicked to obtain a description of the label. The second element is the name of the label. The third one is the *save* switch (when unmarked, the label is saved as blank despite the chosen value). Elements marked as *four* are possible values for these labels.



UID 129

User ID: 1080865[redacted]

Augustus [redacted] @[redacted]

User Description:

I truly believe if you don't like what you are reading, watching, listening just change the channel or discuss, if criminal call the police. gay 4 Trump 🏳️‍🌈✌

Tweets:

@CurtisHouck @mitchellreports the fact you think it is wrong to bring Tony Bobulinski only proves you and the @DNC @CNN @ABC @CBSNews @NBCNews @washingtonpost @nytimes believe you are above the LAW. bunch of spoiled snobs. Fab gay Latin man for @realDonaldTrump @TuckerCarlson #Debates2020
Tweet ID: 13194380[redacted] Link to original

RT @TeamTrump: Tony Bobulinski: In my hour long meeting with Joe Biden we discussed Biden family business dealings with the Chinese, with w…
Tweet ID: 1319435[redacted] Link to original

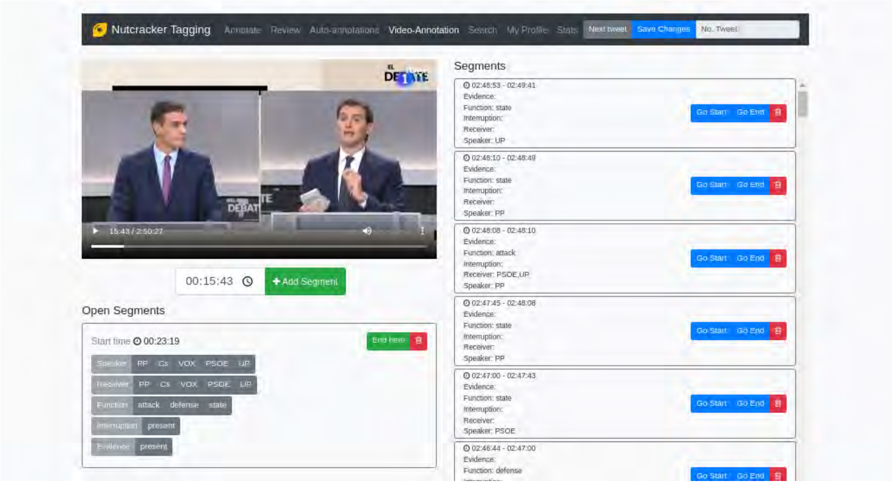
This user's neighbourhood suggests the following properties:

- **DEMOCRATS: positive**
- **REPUBLICANS: negative**

Is there any evidence that at least one property is **incorrect**?

Fig. 2.16.: Nutcracker's automatic annotation review interface

The questions have binary answers, however there is a third button that allows the user to skip the current annotation, that will be assigned to other user. This mechanic was necessary to deal with content written in languages not spoken by the reviewer (but that could be understood by other team members).



Start time 00:23:19

02:48:53 - 02:49:41
Evidence:
Function: state
Interruption:
Receiver:
Speaker: UP

02:48:10 - 02:48:49
Evidence:
Function: state
Interruption:
Receiver:
Speaker: PP

02:48:08 - 02:48:10
Evidence:
Function: attack
Interruption:
Receiver: PSOE,UP
Speaker: PP

02:47:45 - 02:48:08
Evidence:
Function: state
Interruption:
Receiver:
Speaker: PP

02:47:00 - 02:47:43
Evidence:
Function: state
Interruption:
Receiver:
Speaker: PSOE

02:46:44 - 02:47:00
Evidence:
Function: defense

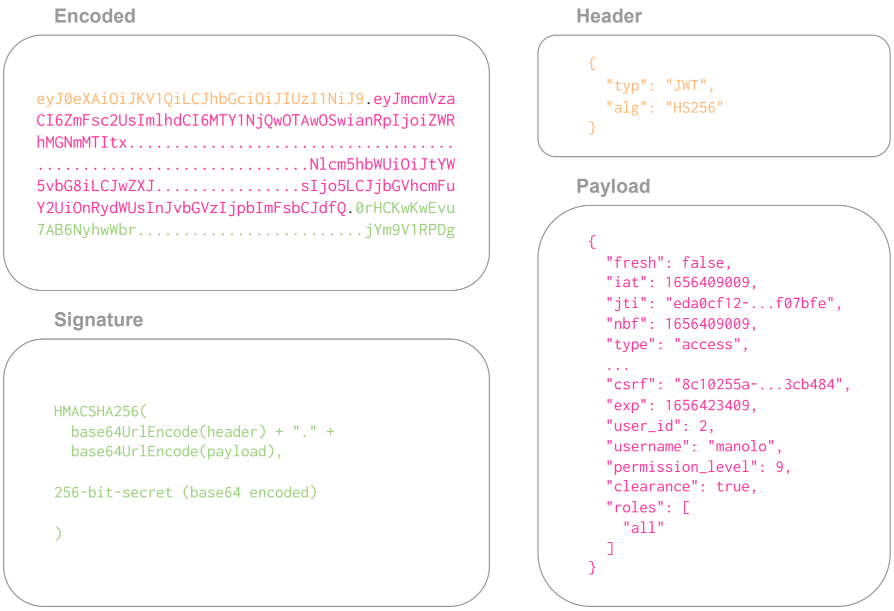
Fig. 2.17.: Nutcracker's video annotation interface

The last important part of the tool is the video annotation tab. It embeds a video that will be tagged using segments. Each segment is defined with two timestamps (begin and end) and several labels that can be defined on a per-instance basis. Segments can overlap to better represent situations like speaker interruptions.

2.6.4 Nutcracker API

The main interaction point between the different modules of Nutcracker is its API. It enables interaction with the stored instances in the database as well as with the workers, therefore it is possible to:

1. Insert new instances of tweets, users, annotations...
2. Delete existing instances.
3. Perform updates over old instances.
4. Database query.
5. Review automatic annotations (propagation).
6. Full-text search.
7. Schedule tasks.



Encoded

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJmcmVza
CI6ZmFsc2UsImlhdCI6MTY1NjQwOTAwOSwianRpIjoiZWR
hMGNmMTItx....................................
..............................Nlcm5hbWUiOiJtYW
5vbG8iLCJwZXJ................sIjo5LCJjbGVhcmFu
Y2UiOnRydWUsInJvbGVzIjpbImFsbCJdfQ.0rHCKwKwEvu
7AB6NyhwWbr.........................jYm9V1RPDg
```

Signature

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),

256-bit-secret (base64 encoded)

)
```

Header

```
{
  "typ": "JWT",
  "alg": "HS256"
}
```

Payload

```
{
  "fresh": false,
  "iat": 1656409009,
  "jti": "eda0cf12-...f07bfe",
  "nbf": 1656409009,
  "type": "access",
  ...
  "csrf": "8c10255a-...3cb484",
  "exp": 1656423409,
  "user_id": 2,
  "username": "manolo",
  "permission_level": 9,
  "clearance": true,
  "roles": [
    "all"
  ]
}
```

Fig. 2.18.: An example of JWT access token

8. Query task status.

API implementation follows the REST constrains (RESTful) with JSON payloads. Endpoints are protected using JSON Web Tokens (JWT) that are generated once the user has logged in (please refer to figure 2.18 🖼 to see an example, page 68). Each endpoint requires a specific permission level to avoid unnecessary alterations to the database. Normal users can retrieve and annotate any tweet, review user annotations, search, and get and add video annotations. Elevated privileges are required to add or delete tweets, annotations, user annotations; and to schedule tasks, check tasks status, and other administrative operations.

2.6.5 Asynchronous Workers

The purpose of this service is multiple. It consist on one or more *workers* designed to perform long-running asynchronous tasks that are queued using ◯ Nutcracker API. We describe below the list of currently supported tasks:

Tab. 2.2.: API endpoints.

GET	`/tweet`	Get relevant tweet
POST	`/tweet`	Create a tweet
GET	`/tweet/<int:tid>`	Get tweet by id
GET	`/tweet/annotations`	Get last annotation for all tweets
GET	`/tweet/<int:tid>/annotation`	Get last annotation for specific tweet
POST	`/tweet/<int:tid>/annotation`	Create new annotation for specific tweet
GET	`/tweet/<int:tid>/suggestion`	Get ontology matches and propagated labels
GET	`/tweet/findByKeywords`	Search tweets using keywords
POST	`/user`	Creates new author
GET	`/user/<int:uid>`	Get author by id
GET	`/user/<int:uid>/tweets`	Get tweets of specific author
GET	`/user/<int:uid>/annotation`	Get last annotation of specific author
POST	`/user/<int:uid>/annotation`	Create annotation for specific author
GET	`/user/annotation`	Get last unreviewed annotation
PUT	`/user/annotation/<int:uaid>`	Review specific user annotation
GET	`/user/annotation/findByStatusAndDecision`	Get user annotations by status and decision
GET	`/labels`	Get labels for document annotation
GET	`/video/labels`	Get labels for video annotation
GET	`/video/<string:name>/annotations`	Retrieve all annotations for specific video
POST	`/video/<string:name>/annotation`	Create new video annotation for specific video
DELETE	`/video/<string:name>/annotation/<int:vaid>`	Delete specific video annotation
GET	`/stats`	Get annotation statistics
GET	`/tasks`	Get completed tasks scheduled by current user
GET	`/tasks/findByStatus`	Get tasks that matches the status
GET	`/task/runByName`	Schedule task by name
POST	`/auth/register`	Register a new user
POST	`/auth/login`	Log in
POST	`/auth/logout/access`	Revoke access token
POST	`/auth/logout/refresh`	Revoke refresh token
POST	`/auth/token/refresh`	Refresh access token
GET	`/auth/token/valid`	Check if tokens are valid

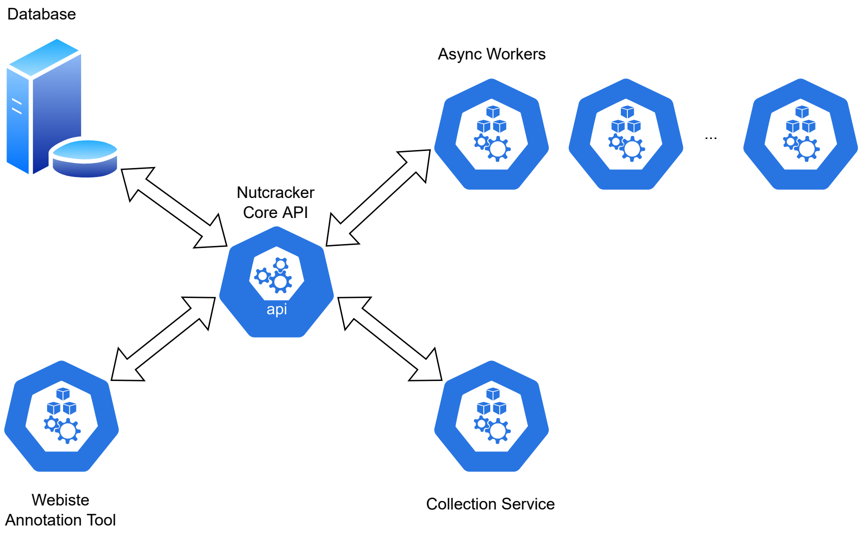


Fig. 2.19.: The API is the core element of the Nutcracker platform, as it allows interaction between services.

- Upload tweets from file, which receives a JSON dump, preprocess it, and stores it in the database.

- Rank tweets, that assigns a numerical value to every tweet on the database that can be used to prioritise the tweets that are shown to the annotators first.

- Annotate emotions, that uses a dictionary of relevant words as well as several regular expressions to procedurally annotate emotions in stored documents.

- Create similarity graph, that stores a `nx.Graph` that is a similarity representation of the stored complex network.

- Expand properties, that takes the manual annotations to compute a direct property of every known user and it performs label propagation to infer property values from unknown users.

Tasks can be easily implemented to extend the capabilities of such workers. Although workers are hot-spawnable and horizontally scalable, they need to be *containerised* and *orchestrated*. This is supported by our *proof-of-concept (PoC)* implementation, however we never performed such kind of production deployment.

2.7 Dataset Production

In order to test the proposed methodology as well as the PoC platform, we produced 4 datasets related to politics. These are:

2019 Spanish National Elections Debate (Spanish) It contains more than 120k tweets related to the 2019 Spanish General Elections, retrieved using the official hashtag #Debatea5RTVE. Tweets were labelled by members of The Nutcracker Project.

2021 Madrid Regional Elections Debate (Madrid) It contains more than 200k tweets in Spanish related to the 2021 Madrid Regional Elections, retrieved using the official hashtag #DebateTelemadrid. Annotations were made with the help of a class of Political Science students under the supervision of two experts of The Nutcracker Project.

2020 USA Presidential Debates (USA) It contains more than 11k tweets related to the 2020 USA Presidential Debates, retrieved using the official hashtag #Debates2020. Annotations were made with the help of italian Linguistic Students under the supervision of three experts of The Nutcracker Project.

Arabic (Arabic) It contains roughly 1.4M tweets in Arabic. This dataset was built incrementally by using tracked retrieval (see section 2.6.2, page 58) and manually-reported users and tweets. Tweets were labelled by members of The Nutcracker Project.

Table 2.3 contains detailed description of the aforementioned datasets. There are significant differences between them. *Madrid* dataset has the largest number of edges between nodes; *USA* is the smallest one; and *Arabic* is the least connected one. We advise the reader to keep these particulars in mind, since they would notably influence the results.

Tab. 2.3.: Description of size and graph dimensions for each produced dataset.

Dataset	Spanish	Madrid	USA	Arabic
No. tweets	120117	226348	11231	1441281
No. users	51817	44182	9114	95133
mean coRT/node	199	1037	508	2.21
std coRT/node	396	1272	590	2.42
No. labels	5	6	2	2
Balanced	No	No	No	No

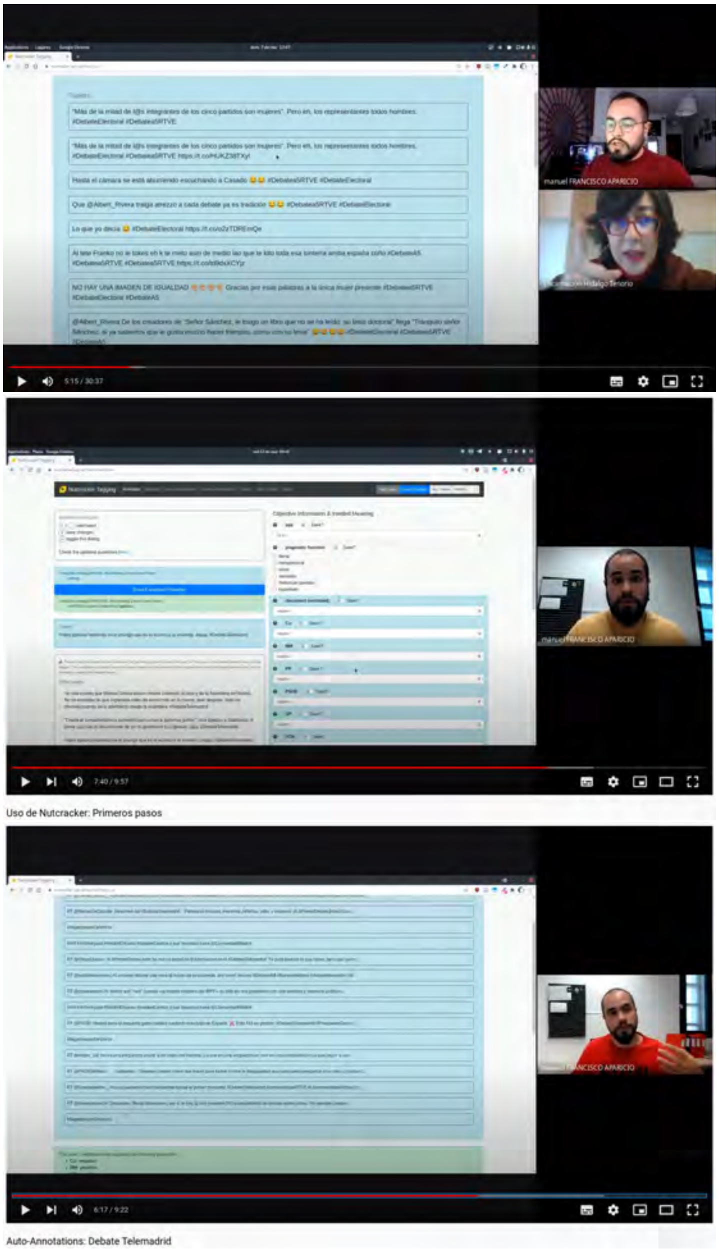
To produce these datasets, we followed our proposed methodology (see section 2.5, page 43). Oracles were human expert annotators with a previous training in the platform usage. In early development stages, it as a guidelines document (see appendix A); after our proposed methodology was established, we used video-tutorials (please refer to figure 2.20, page 73).

2.7.1 Label set

Datasets were produced with different purposes therefore they have different label sets. We sum them up below, and we provide an in-depth description of these labels in appendix B.

2019 Spanish National Elections Debate (Spanish) We used *binary gender*, *age*, *sentence type*, *speech act*, *pragmatic function* and *mood* as general labels; *document sentiment*, *PP*, *Cs*, *PSOE*, *UP* and *VOX* (that were the candidate's political parties) as expandable properties; and a full taxonomy of emotions [28].

2020 USA Presidential Debates (USA) We used *binary gender*, *age*, *sentence type*, *speech act*, *pragmatic function* and *mood* as general labels; *document sentiment*, *REPUBLICANS* and *DEMOCRATS*, (that are the two main political parties in USA) as expandable properties; and a full taxonomy of emotions [28].



"Más de la mitad de l@s integrantes de los cinco partidos son mujeres". Pero eh, los representantes todos hombres. #DebateElectoral #Debatea5RTVE

"Más de la mitad de l@s integrantes de los cinco partidos son mujeres". Pero eh, los representantes todos hombres. #DebateElectoral #Debatea5RTVE https://t.co/HUKZ38TXyl

Hasta el cámara se está aburriendo escuchando a Casado 😂😂 #Debatea5RTVE #DebateElectoral

Que @Albert_Rivera traiga atrezzo a cada debate ya es tradición 😂😂 #Debatea5RTVE #DebateElectoral

Lo que yo decía 😂 #DebateElectoral https://t.co/o2zTDREmQe

Al tete Franko no le toques eh k te meto aun de medio lao que te lolo toda esa tontería arriba españa coño #DebateA5 #Debatea5RTVE #Debatea5RTVE https://t.co/td9dxXCYjr

NO HAY UNA IMAGEN DE IGUALDAD 👎👎👎👎 Gracias por esas palabras a la única mujer presente #Debatea5RTVE #DebateElectoral #DebateA5

@Albert_Rivera De los creadores de "Señor Sánchez, le traigo un libro que no se ha leído: su tesis doctoral" llega "Tranquilo señor Sánchez, si ya sabemos que le gusta mucho hacer tiempo, como con su tesis" 😂😂😂😂 #DebateElectoral #Debatea5RTVE #DebateA5

manuel FRANCISCO APARICIO
5:15 / 30:37
Objective Information & Implied Meaning
manuel FRANCISCO APARICIO
7:40 / 9:57
Uso de Nutcracker: Primeros pasos
FRANCISCO APARICIO
6:17 / 9:22
Auto-Annotations: Debate Telemadrid

Fig. 2.20.: Video lessons made to demonstrate platform usage to human oracles.

2021 Madrid Regional Elections Debate (Madrid) We used *age*, *pragmatic function* and *implicit connotations* as general labels; *document sentiment*, *Cs*, *MM*, *PP*, *PSOE*, *UP* and *VOX* (that were the candidate's political parties) as expandable properties.

Arabic (Arabic) We used *document sentiment*, *is reply*, *text function*, *kind of user*, *language* and *linked to terror or radicalism* as general labels; and *relevance*, as expandable property.

2.7.2 Inter-rater reliability

One crucial step when producing datasets consists in measuring the subjectivity of the annotators (also known as *raters* or *coders*). There are several metrics that can be used to characterise annotation reliability [84]. We chose *Krippendorff's Alpha* [85, 86] due to its versatile behaviour and native capabilities to model multi-rater and multi-label environments. We used the implementation available in `nltk.metrics.agreement`[9].

Table 2.4 provides a description regarding (1) the number of annotators involved, how many of them were native speakers and how many were experts on the field (w.r.t. the dataset context); (2) the number of annotated documents, mean and standard deviation per annotator; (3) size of the sample used to compute the reliability, as well as the ratio against the number of annotators; and (4) reliability scores. It shows that the reliability strength varies from substantial to almost perfect (0.75 or more).

Although we know which users are relevant, note that most documents in the Arabic dataset are not fully annotated[10], therefore we cannot include reliability measures for this dataset. There is another limitation regarding Spanish dataset. The size of the sample used to measure the reliability is marginally smaller (lowest ratio between sample size and number of annotators). Despite that we advise the reader to take *Krippendorff's Alpha* value for this dataset with caution, we trust that this metric is reliable for the purpose of this study, since propagated data seems to prove that annotations are accurate (see section 2.8).

[9] `https://www.nltk.org/api/nltk.metrics.agreement.html`

[10] Annotations are tentative (tracked collection)

Tab. 2.4.: Annotation statistics and reliability

	Spanish	Madrid	USA	Arabic[10]
No. Annotators	6	43	11	3
↪Native Speakers	2	43	0	1
↪Experts	2	2	3	3
Annotated Tweets	2340	4384	3338	6624[10]
↪mean per ann.	214.54	115.76	316.36	-
↪std per ann.	203.53	40.07	234.88	-
Sample size	40	1092	261	-
Ratio sample/annotators	6.67	25.4	23.73	-
Krippendorff's Alpha	.80401	.75459	.80026	-
↪Strength [87]	Almost perfect	Substantial	Almost perfect	-

Tab. 2.5.: Hyperparameters used in the conducted experiments.

steps (i)	1
p_0	0.1
p_1	0.25

2.8 Experimental Work

In order to check the performance of our proposal against current weak-supervision techniques, we applied several classic machine learning algorithms (Support Vector Machine (SVM), Random Forest (RF), AdaBoost (AB) and Multinomial Naive Bayes (MNB), all of them available in *scikit-learn* package [88]) as well as active learning (ActiveLearner available in *modAL* [89]) and deep learning (using state-of-the-art BERT-based language model [90]).

Experiments were designed to test the following aspects:

- Classification accuracy of weak-supervision methods trained with different sizes of training sets, which measures how good these mechanisms are when predicting and also how many labelled documents they need.
- Expansion accuracy or classification performance of our proposal, which measures how good our proposal is when predicting user property values.
- Expansion reach or ability to spread, which measures how far our methodology is able to predict user property values and how many labelled documents we need.

For standard algorithms, experiments were conducted using default hyperparameters. Our proposal's choice of hyperparameters is shown in table 2.5 and they were based on expert evaluation.

2.9 Results and Discussion

Results shows that the number of automatic annotations grows quickly with the first annotations due to the ranking strategy, and it stabilises after the annotation process reach non-influential users. For example, for the Spanish National Elections dataset, the top 25% has more than 191 connected users, hence the grow. The ratio between number of accepted annotations versus total annotations has a mean value of 0.89, with an almost negligible standard deviation of 0.0045. It keeps steady regardless of the number of manual annotations. This points that the chosen user representation (*co-retweet* graph) behaves coherently.

We applied weak-labelling techniques using aforementioned classifiers. Section 2.6 shows performance metrics for experiments run with 100 labelled instances. Our proposal manages to score the best $f1$-score results for all three datasets, specially for the larger ones, *Madrid Elections* and *Spanish Elections* datasets. In the worst case scenario, property expansion manages to improve $f1$-score by 0.275 w.r.t. the best performing classifier.

Figure 2.21 shows that our proposal significantly improves the results of other weak supervision techniques when the number of annotations is low. However, as the training sample grows, so does the performance of the rest of methods. In some cases, they even surpass the expansion mechanism. Such is the case of the USA dataset, which is the smallest one (see table 2.3).

After the expansion, all the automatic annotations were evaluated. Oracles were asked several questions regarding the evidence (see section 2.5.6). Their answers determined whether the expansion was rejected or accepted (see figure 2.5). Table 2.7 presents the results of this evaluation for the Spanish dataset. Most of the automatic labels could be confirmed programmatically, as their subjects only *retweeted* content, and therefore the decision on the sentiment of those users could be derived from the sentiment of the original tweets.

When annotating tweets, we assume that the cost of evaluating a tweet is uniform [91], since documents may have 240 characters as much. The number of manual annotations is called *effort*. Table 2.9 shows the effort required to reach at least 0.75 in $f1$-score. Note that we stopped several experiments

Tab. 2.6.: Micro performance metric results for AL (Active Learning), AB (AdaBoost), BERT, MNB (Multinomial Naive Bayes), RF (Random Forest) and SVM (Support Vector Machine) when trained with 100 labelled instances. Our proposal achieves best overall results for all three datasets.

dataset	meth.	accuracy	precision	recall	$f1$-score
Spanish	AB	0.462	0.469	0.379	0.410
	AL	0.382	**0.893**	0.029	0.055
	BERT	0.422	0.423	0.556	0.480
	MNB	0.522	0.403	0.579	0.469
	RF	0.529	0.597	0.566	0.488
	SVM	0.543	0.567	0.620	0.484
	Ours	**0.728**	0.859	**0.680**	**0.759**
Madrid	AB	0.194	0.562	0.482	0.423
	AL	0.221	0.562	0.506	0.417
	BERT	0.206	0.481	0.479	0.480
	MNB	0.214	0.468	0.500	0.417
	RF	0.256	0.632	0.533	0.466
	SVM	0.254	0.605	0.531	0.459
	Ours	**0.437**	**0.747**	**0.945**	**0.835**
USA	AB	0.890	0.898	0.967	0.925
	AL	0.890	0.883	0.966	0.923
	BERT	0.889	0.892	0.967	0.928
	MB	0.885	0.874	0.967	0.918
	RF	0.905	0.912	**0.969**	0.934
	SVM	0.897	0.890	0.965	0.926
	Ours	**0.932**	**0.989**	0.934	**0.961**

before they reach that threshold. Our proposal is the method that requires less human annotations therefore ideal to reduce the weak supervision labelling costs.

2.10 Conclusions

Nowadays, supervised models are the *de facto* standard of industrial solutions and the most popular approach when applying ML. The process of dataset supervision is costly and time consuming, and it may take years depending on the difficulty of the labelling task and the amount of data required.

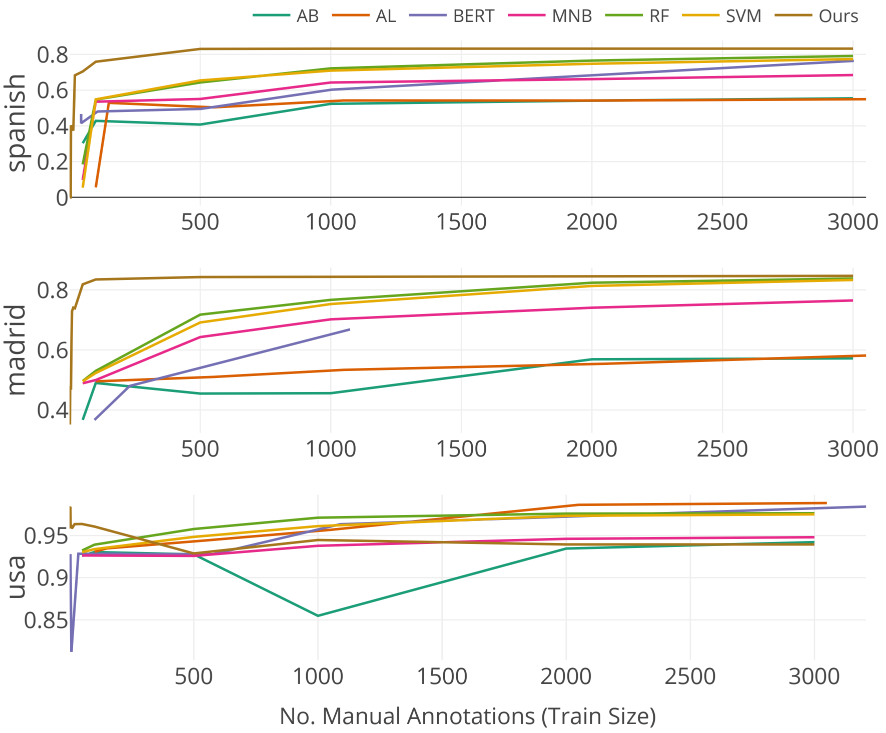


Fig. 2.21.: Evolution of $f1$-score against the number of manual annotations. Our proposal achieves a higher performance with less effort (manual annotations) w.r.t. the rest of the weak supervision methods.

Current solutions include manual approaches (that can be internal or outsourced), that are expensive; automatic ones (like data programming or synthetic labelling), that have restricted use cases; and weak-supervision techniques, that are considered semi-automatic but produce databases with less than optimum quality.

Most common weak supervision techniques are based on models that are trained upon textual features. However, SNS offer meta-data that may be used to improve dataset quality while reducing the effort required to supervise the dataset.

In this chapter, we proposed a human-in-the-loop methodology to reduce SNS data supervision costs. To that end, we introduced the concept of *Similarity Semantic Network*, that are a mechanism that enables similarity reasoning in semantic networks. Then, we presented a particular implementation of a

Tab. 2.7.: Results of the evaluation process for the automatic annotations generated by the proposed expansion mechanism.

Dataset	Spanish	Madrid	USA
Confirmed	35614	30263	5184
Indiscernible	316	25	38
Rejected	641	704	118

Tab. 2.8.: Accepted annotations for the total of automatic annotations generated when varying the number of iterations with 100 manual annotations. The reach and accuracy of the methodology changes with the number of iterations, until it stabilises.

dataset	iter.	accepted	total	ratio
Spanish	1	31025	35200	88.14%
	2	31894	36882	86.48%
	3	31978	37086	86.22%
	4	31981	37099	86.20%
Madrid	1	27771	34183	81.24%
	2	27848	34309	81.17%
	3	27848	34311	81.16%
	4	27848	34322	81.14%
USA	1	4990	5175	96.43%
	2	5113	5653	90.45%
	3	5121	5670	90.32%
	4	5121	5670	90.32%

Similarity Semantic Network that uses *deep relations* to propagate labels from known users to unknown ones. Since both *deep relations* and the *inference rules* of the Similarity Semantic Network are interpretable, our proposal can be used to deal with environments were accountability and transparency are required.

We implemented a *proof-of-concept* platform, that we called Nutcracker, to test our methodology in real-case scenarios. It presents multiple capabilities, such as data retrieval, data annotation, semantic network building and maintenance, and label expansion. We approximated the *common-opinion deep relation* using the *co-retweet* function. Ultimately, we also produced, with the help of several human experts, four weak-supervised datasets related to politics: 2019 Spanish National Elections Debate, 2021 Madrid Regional Elections Debate (both in Spanish), 2020 USA Presidential Debates (in English) and Arabic (in Arabic).

Tab. 2.9.: Effort required to reach at least 0.75 of $f1$-score or until stopping criteria is reached (more than 3000 annotations). Results shows that our proposal is, by far, the method that requires the less human effort. *Notice that USA dataset results are over the threshold from the beginning, therefore effort is orientative.

dataset	method	effort	$f1$-score
Spanish	AB	3000	0.5542
	AL	3050	0.5493
	MNB	3000	0.6845
	RF	2000	0.7654
	SVM	3000	0.7738
	Ours	100	0.7590
Madrid	AB	3000	0.5719
	AL	3050	0.5813
	MNB	3000	0.7647
	RF	1000	0.7668
	SVM	1000	0.7529
	Ours	50	0.8185
USA*	AB	50	0.9260
	AL	50	0.9308
	MNB	50	0.9264
	SVM	50	0.9303
	RF	50	0.9701
	Ours	1	0.9786

Our results show that the number of correct automatic annotations grows quickly with respect to manual annotations. The ranking strategy promotes those tweets that will yield more information. In the case of the Spanish dataset, we obtained more than 8000 automatic annotations with only 10 manual ones, which translates to an eight-hundredth of the baseline effort. Assuming that the cost of labelling a tweet is uniform [91], our method is able to obtain the same $f1$-score than other methods reducing the effort in, at least, one order of magnitude.

The ratio between accepted annotations versus the total number of them has a mean value of 0.89 with less than a 0.5% deviation, which implies that our proposal is also superior to other weak-supervision techniques in terms of accuracy. In the worst case scenario, our expansion mechanism still overcomes the best performing classifier by 0.275 points in $f1$-score.

Tab. 2.10.: Micro performance metrics for our proposal against the number of annotations.

	accuracy			precision			recall			*f*1-score		
#anns	mad	spa	usa	mad	spa	usa	mad	spa	usa	mad	spa	usa
5	.054	.513	.928	.524	.896	.980	.423	.260	.939	.468	.403	.959
7	.172	.509	.933	.831	.896	.986	.502	.253	.936	.626	.394	.961
9	.308	.460	.929	.852	.690	.988	.637	.269	.930	.729	.387	.958
14	.313	.453	.932	.848	.710	.988	.658	.253	.934	.741	.373	.961
20	.255	.656	.937	.697	.805	.989	.785	.593	.939	.739	.683	.964
50	.409	.678	.937	.727	.847	.989	.936	.603	.939	.818	.704	.964
100	.437	.728	.932	.747	.859	.989	.945	.680	.934	.835	.759	.961
500	.460	.796	.904	.758	.866	.957	.948	.798	.902	.843	.831	.929
1000	.463	.798	.931	.759	.866	.959	.951	.801	.931	.844	.832	.945
2000	.470	.799	.923	.761	.866	.958	.954	.801	.922	.847	.832	.939
3000	.472	.799	.923	.762	.866	.958	.953	.801	.922	.847	.832	.939

Although results of our experiments were good, there are certain limitations of our *proof-of-concept* that need to be taken into account. We did not deal with the problem of topic extraction, since we assumed that all the documents pertaining to an event hashtag will present the same topics. This assumption introduces noise and may alter results or diminish the quality of the obtained dataset. Despite that we only performed one cycle of annotations, it is straightforward to notice that the number of automatic annotations reach a plateau due to the ranking strategy. The same is expected to happen with iterative supervision, since properties cannot be expanded infinitely. After the plateau is reached, other weak supervision mechanism may present better performance.

Similarity Semantic Networks and *deep relations* can be applied to analyse other SNS, shaping the abstract relations with the particular mechanics of the particular social network. Consequently, future work should include developing concrete reasoning mechanisms in other social networks as well as testing its performance in other topics rather than politics.

3 Improving Comprehensibility of Interpretable Classification Models

Objectives

1. Study current methods for interpretable Machine Learning (ML) pipelines to establish how comprehensible they are.
2. Propose new mechanisms in any of the steps of the classification pipeline to ensure comprehensibility of interpretable models.

3.1 Introduction

Machine Learning (ML) models are not perfect. One of the main principles of ML is *generalisation*, in which the target is to minimise the error (or loss) when working with never-seen instances or samples. But, with the exception of very specific tasks or datasets, there is always an error. It is not the only limitation that raises concern, as there are many ethical questions that should be taken into account when training ML models. In 1988, a hospital in United Kingdom was found guilty of racial and sexual discrimination for using a computer program to take initial decisions for job applicants [92]. The program was trained using data from the admission process, and the former imitated the bias of the latter.

Racism and sexual discrimination are quite common problems for machine learning algorithms (big companies like Google [93] or Microsoft [94] faced similar issues), however there are other ethical concerns that need to be addressed separately. Organisations like FAT/ML [40] are trying to make these issues visible and to make researchers and practitioners aware of the problems they may face. Responsibility, explainability, accuracy, auditability and fairness are the principles that should be respected when producing models that may take part in a decision making processes or other applications that may have a social impact.

Models working with Social Networking Sites (SNS) data are subject to the same principles. Currently, censor algorithms are a controversial topic regarding this matter, since preemptive closing of accounts limits free speech and should have explicit reasoning [95, 96]. Recommendation systems also present similar issues. It has been proved that filtering the content showed to a user may impact the way they they think, their interests or even affect third parties [97, 98, 99, 100].

We are going to focus in those cases in which the decision needs to be fair, therefore models are required to be interpretable. A model that is fully interpretable can be described analytically to understand its strengths and weaknesses, to discover patterns in data that may have a negative result, or even to facilitate experts evaluate and correct the bias.

The typical way to produce interpretable models requires using predictors or classifiers that are, by definition, interpretable. Linear and logistic regressions, decision trees and k-nearest neighbours are good examples of these algorithms [101, 102]. Unfortunately, a model that is interpretable may not be comprehended in practice, since complex tasks require complex models whose analysis tends to be unmanageable.

There are several ways in which we can evaluate how good a model is [103]. Normally, in terms of accuracy, efficiency and interpretability. Despite that, in recent years, machine learning research has focused mostly in accuracy metrics (such as *precision*, *recall* or *ROC AUC*), forgetting about interpretability. State-of-the-art approaches (such as ELMo or BERT [104, 105]) are based on *deep* techniques. Deep Artificial Neural Networks (DNN) are very popular and, currently, the cutting-edge technology to deal with many complicated tasks. Yet, they cannot be interpreted (black-box models).

Trending solutions try to explain black-box models by using surrogated models as proxy evaluators. The technique consists in training a second (and interpretable) model to predict the output of the first model. Then, evidence on the behaviour of the main model is extracted from the analysis of the posthoc model, which is usually problematic [41]. And even if we were to accept explainable ML as a valid solution, models would still be too complex to understand.

In order to achieve truly comprehensible models, we need to reduce the complexity of interpretable models. Our hypothesis is that it is possible to reduce model complexity by using less but more relevant features. We think that it can be achieved encoding each document using features that are inherent predictors of a class. This is, partially, what linguists do, and, although in a different domain, Moreo et al. [106] successfully tried a similar approach.

Our proposal relies on what we have called *distinguishing expressions*, that are sequences of words that are relevant for a class but not for the others. First, we greedily search candidates to generate expressions; then, we select those that are useful in terms of statistical relevance and exclusivity; we encode documents using selected expressions; and we feed the samples to any interpretable pipeline.

The rest of this chapter is organised as follows. In Section 3.2, we review related work in the field. Section 3.3 conducts an initial screening on the comprehensibility of out-of-the-box interpretable classifiers. We present our proposal in Section 3.4. Results are discussed in Section 3.5. Finally, conclusions are presented in Section 3.6.

3.2 Related Work

Text mining research in Social Media has become very relevant in the last few years, as they are ground for quite a few computer science applications. From analysing human behaviour to stock prediction, there are a lot of ongoing projects and researches based on topic detection [107, 108, 109, 110], measuring user's influence [111, 112, 113, 114], sentiment analysis [115, 116, 117, 118, 119], opinion mining [120, 121, 122, 123], and text summarisation [124, 125], among others.

Machine learning models have been developed to classify and predict properties of SNS users. Several methods have been used traditionally for keyword (and feature) extraction from text. Once extracted, most popular encodings are based on words and/or n-grams extraction. *Bag of words (BoW)* [126] and *TF-IDF* [127] are the most popular document representations.

However, these techniques may not be good enough under *microblogging* circumstances, since they result in highly-dimensional and very sparse feature matrices [128]. Applications like authorship identification (duplicate accounts, inter-network identification...) have demonstrated that traditional methods are not feasible in short-message contexts like Twitter, as they tend to assume a minimum text extension under which models would not be suitable enough. Alternative techniques like *style markers* have been proved better [129]. Other document-pivot methods present similar problems, and recent approaches based on co-occurrence, *TF-IDF* and/or pattern recognition techniques (such as FP-Growth [130]) are being used for this purpose [131, 132].

In any case, when dealing with natural language models, the feature space is usually extremely wide. Normally, a set of documents has a few thousand unique words that are used as features, despite many of them can be considered noisy or not relevant at all. It is required to select and/or recombine them, not only to decrease the complexity of the problem but also to improve classification performance [133].

There are several classes of feature selection (FS) methods, but they are generally grouped into three categories [134]:

- Filtering methods. Given a set of features, they apply an evaluation function and select the k best, where k is an hyperparameter. They score each feature taking into consideration different aspects of them, like document frequency (DF) or TF-IDF.

- Wrapper methods. Given a set of features, they select different subsets of them to train a classifier and check out how good its performance is. Since selected features are tested directly within the classifier, they normally achieve better performance than filters.

- Hybrid methods. They combine both techniques: first, they perform a filter to reduce the number of features; then, optimal subset is computed by feeding a classifier.

There is another group of feature selection techniques called *embedded methods* [135]. These methods are inherent to the classification stage, since they take part in the training process (e.g. decision trees), and they are usually not considered independently.

There are loads of filtering methods available and backed by the scientific community. We have selected a few of them based on their current relevance, goodness metrics and community acceptance. They are presented below:

CHI2 (chi-square) χ^2 is one of the most popular filtering methods for feature selection problems. It is possible to use the statistical test in order to check the independence of two events ($p(AB) = p(A)p(B)$), in this case, a feature and a class.

$$\chi^2_{(t,c)} = \frac{D \times \left[p(t|c)p(\bar{t}|\bar{c}) - p(\bar{t}|c)p(t|\bar{c})\right]^2}{p(t)p(\bar{t}) - p(c)p(\bar{c})} \tag{3.1}$$

χ^2 is defined for text classification through equation 3.1, where: D stands for the total number of documents, t is a feature and c is a class [136, 137, 138].

Information Gain (IG) IG is based on entropy (information theory). It measures the gain of a feature with respect to a given class (decrease in entropy between considering the feature or not) [139]. IG is defined as stated in equation 3.2, where t is a feature and c a class [140, 141, 142, 143].

$$IG_{(t,c)} = p(t|c)\log\frac{p(t|c)}{p(t)p(c)} + p(\bar{t}|c)\log\frac{p(\bar{t}|c)}{p(\bar{t})p(c)} \tag{3.2}$$

Mutual Information (MI) MI measures how much information two variables share, in this case, a feature t and a class c. It is defined in equation 3.3 [142, 144]. It can be proved equal to IG for binary problems [137].

$$MI_{(t,c)} = \log\frac{p(t|c)}{p(t)p(c)} \tag{3.3}$$

with t and c being a feature and a class, respectively.

Odds Ratio (OR) OR measures the probability of a term t and a class c co-occurring normalised by the probability of t occurring in other classes [138, 144].

$$OR_{(t,c)} = \log \frac{p(t|c)(1 - p(t|\overline{c}))}{(1 - p(t|c))p(t|\overline{c})} \tag{3.4}$$

where t is a feature and c is a class.

Expected Cross Entropy (ECE) ECE computes the distance between the class c distribution and the class distribution co-occurring with the feature t. [145] defined ECE as in equation 3.5.

$$ECE_{(t,c)} = p(t) \left(p(c|t) \log \frac{p(c|t)}{p(c)} + p(\overline{c}|t) \log \frac{p(\overline{c}|t)}{p(\overline{c})} \right) \tag{3.5}$$

ANOVA F-value It is used to check if there is a significant difference between the variance of two variables [146]. In one-way ANOVA, the F-statistic is defined as in equation 3.6:

$$F\text{-statistic} = \frac{\text{variance between groups}}{\text{variance within groups}} \tag{3.6}$$

Galavotti-Sebastiani-Simi coefficient (GSS) [147] proposed a simplified version of χ^2 given by equation 3.7 [143].

$$GSS_{(t,c)} = p(t,c)p(\bar{t},\overline{c}) - p(t,\overline{c})p(\bar{t},c) \tag{3.7}$$

where t is a feature and c is a class.

All these filters present similar drawbacks. Mainly, features are selected regardless of the classifier, so the selected feature subset may not be ideal for every classification stage. However, they are quicker computing the aforementioned subset than other categories of FS mechanisms.

On the other hand, wrappers assess the relevance of each subset of features within the context of a given classifier. Goodness metrics for the classification model determine how good each feature subset is. Hence, the FS method will choose the subset of features that yields the best performance. There are tons of generic wrappers for feature selection, such as [148, 149, 150, 151]. They usually achieve better performance than filtering methods.

However, the complexity of evaluating each subset is quite high (exponential). To overcome this disadvantage, they are usually combined with metaheuristics such as genetic algorithms, ant colony optimisation, particle swarm optimisation and/or iterated local search [152]. Despite the complexity of performing a search, they still are suboptimal approaches and classifier-dependant.

Wrapper approaches can also be combined with filters in order to narrow the space search to a promising area (hybrid methods). Hence, they are suitable for early convergence (which can lead to local extrema) and prone to overfitting (at least with small datasets) [153].

Since 2012, *state-of-the-art* is abandoning these *handcrafted* methods in favour of auto-encoded features [154]. Deep learning techniques automate the pipeline building process by learning features and subsequent classification rules at the same time. Nowadays, their popularity is outstanding. [154] lists several categories of deep models, including the following:

- Recurrent Neural Networks (RNNs) and their famous variant, Long-Short Term Memory (LSTM). They are designed to extract dependencies and patterns over time within sequence of words.
- Convolutional Neural Networks (CNNs) try to capture invariant structures over documents, such as expressions or figures of speech.
- Attention, which focuses on correlation between words by weighting each word with respect to others in the document.
- Transformers, built upon the *Attention* concept to overcome the sequential limitation of RNNs.

These techniques yielded language models such as ELMo [104], BERT [105]), GPT-2 [155] and GPT-3 [156]. In its most general form, the mechanism relies on representing words (and even context) as multi-dimensional vectors (*embeddings*) in a manner in which those related to similar words (in terms of meaning and/or related context) are closer in the space. This allows that certain operations can be made over them with acceptable accuracy, such as adding restrictions or substracting partial meanings (arguably, the most popular example is the one were $\overrightarrow{\text{king}} - \overrightarrow{\text{man}} + \overrightarrow{\text{woman}} = \overrightarrow{\text{queen}}$).

After computing such *embeddings*, they reduce dimensionality, by *encoding* hidden patterns within the weights of intermediate layers, until (1) they can be handled by an arbitrary classifier or until (2) dimensionality is reduced to the classes themselves.

We cannot consider these approaches since they behave as black-box models and cannot be interpreted. In fact, most of the techniques we have reviewed regarding feature selection were focused in reducing dimensionality and improving classification performance. However, little attention has been paid to *how interpretable* are the selected features.

It is possible to obtain a set of features that are easier to understand by humans without significantly affecting classification performance [41]. In order to ensure model integrity, they should be transparent [40]. Furthermore, for some text classification problems, there are features that should not be selected at all, at least in *microblogging* context. For examples, those related to typing errors or contractions, when these are due to the limit of characters.

There is no general consensus on how to measure interpretability. This concept, along with similar ones like *comprehensibility*, *understandability* and so on, are usually synonyms [157]. However, there is kind of a distinction between those related to the ability to read the model (*interpretability*) and those related to the human capacity to understand it (*comprehensibility*) [158]. [102] defined *interpretability* as "*the ability to explain or to present in understandable terms to a human*". They elaborated a taxonomy on how to evaluate interpretability:

1. Application level, consisting in an expert evaluation of the model itself.

2. Human metrics, where any human can perform such evaluation without the need of being a domain expert. It is normally performed by comparison with other models/explanations.

3. Proxy tasks, when we make the assumption that the user understands the model and we only compare parameters within it (e.g. depth of a decision tree).

It is particularly interesting to perform evaluations at application level. Nevertheless, this is a very expensive task that most studies do not contemplate unless it is strictly necessary.

Our proposal focuses in a functional evaluation (proxy task). We based it in [106], who tried a similar approach in a *frequent answered questions (FAQ)* retrieval method and continued with [159, 160, 161]. However, this model computes the whole set of *minimal differentiator expressions*, which is computationally eager and it is only viable in closed domains like the one proposed by the authors. It is not a valid solution for an open, always-growing environment like Social Media.

3.3 Preliminary Analysis

SNS offer a lot of analysis possibilities, but it comes at a price. The peculiarities of the posts in these platforms (such as misspelled words, *hashtags* and the lack of context) complicate the results for classic techniques, and it becomes a necessity to tweak or replace them [128]. In this section, we will measure accuracy metrics against number of features for several document representation and feature selection mechanisms.

There are several types of features that can be extracted from text. Arguably, within the context-dependant categories, the most important and widely used ones are words, set of words (cooccurrent terms) and n-grams. These are used to represent documents, normally in a one-hot encoding fashion. Other approaches that can produce features could be topic modelling methods (such as LDA and its byproducts) or embedding-related representations (such as doc2vec, fastText, ELMo, BERT or GPT-2) [104, 105, 162, 163, 164, 165, 166]. LDA also fails to deliver good document representations since there is no context in *microblogging* posts, and workarounds consists normally in tweet aggregation [167]. Selecting tweets to concatenate them is not a trivial task and can bias the final result. As for embedding methods, they rely on weights for neural network hidden layers, becoming impossible for any human to understand it.

Consequently, we are going to analyse the effectiveness of BoW and TF-IDF encodings with word features. We are also considering n-grams and term cooccurrence. However, these techniques deliver a tremendous number of characteristics that need to be filtered and/or weighted, not only for efficiency sake but also for interpretability purposes.

Feature selection mechanisms can be categorised in *filtering methods*, *fusion methods*, *mapping methods*, *clustering methods* and *deep learning based methods* [135]. Once again, deep learning methods are not useful for our purposes. From the remaining ones, we have chosen *mutual information (MI)*, *information gain (IG)*, *chi-squared* (χ^2), *GSS coefficient*, *odds ratio* and *ANOVA f-values* due to their popularity in machine learning applications [135, 144].

3.3.1 Data and Settings

We limited the classifiers to interpretable ones. We only compared results between themselves to show if, in fact, feature sets enhance accuracy and interpretability. Taking Molnar [101] into account, we chose k-nearest neighbours, decision trees and, additionally, random forests. We did not perform hyperparameter tuning on them.

We conducted cross-validated experiments in four collections of tweets. We followed a one-vs-rest approach to binarise the classes (multi-label evaluation will be considered in future work).

- US Airlines Sentiment. A collection of approximately 15k tweets from *Crowdflower's Data for Everyone library*[1] tagged for sentiment towards US airlines. It contains three classes (positive, neutral and negative), although we have binarised it in positive or not.

- Twitter User Gender. A dataset of around 25k tweets labelled as *male*, *female*, *brand* or *unknown* from *Crowdflower's Data for Everyone library*[1]. We have removed *brand* and *unknown* instances in order to have binary classes (12k instances remaining).

- Sentiment140 dataset [168]. 1.6M instances tagged as *positive*, *negative* or *neutral* sentiment. We change the classes into positive or not.

- TASS Sentiment Analysis dataset [169]. A collection of 7k Spanish tweets gathered by *SEPLN*, classified in positive, neutral and negative sentiment. We binarised the classes in the same fashion (positive or not).

[1] `https://www.figure-eight.com/data-for-everyone/`

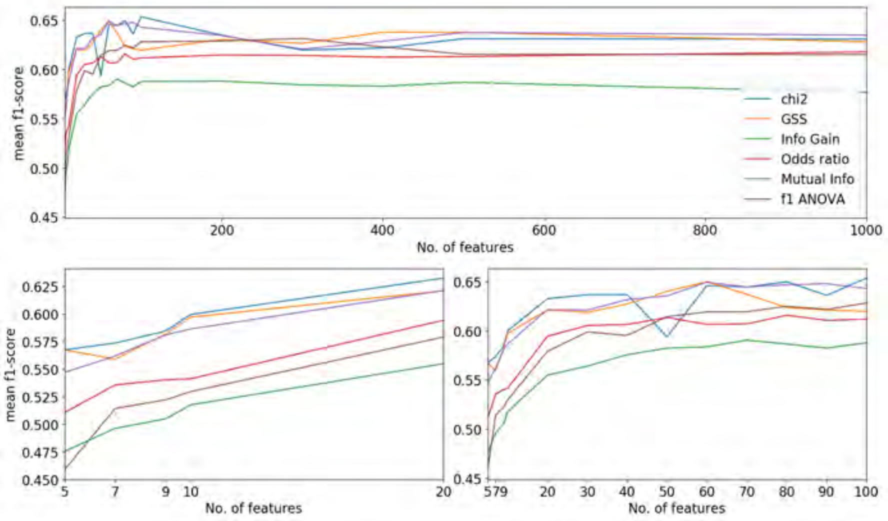


Fig. 3.1.: Comparison between feature selection methods. They show the typical behaviour where the score increases with the number of features up to a point where there is no sense to keep increasing the space search anymore.

We used the *bag of words* and *TF-IDF* implementations available in the *scikit-learn package* [88] with the default parameters set. We also used the same implementations for bigrams and term cooccurrence, and we implemented a class that transform cooccurrence matrix into set of words to be used as features.

For the listed utilities and classifiers, we used the implementations available in the *scikit-learn 0.20.3 package* with the default parameters set, unless specified otherwise.

3.3.2 Feature Selection Comparison

There are a lot of features that can be extracted from text, regardless that they are words, cooccurrent terms or n-grams. It is almost mandatory to apply feature selection mechanisms to any model whatever their purpose in order to reduce the training complexity. In particular for this study, feature selection is necessary for our models to be interpretable. We have chosen six popular selection methods and we compared them to illustrate their behaviour.

method	std
MI	0.105677
χ^2	0.195409
GSS	0.216597

Tab. 3.1.: Standard deviation for top 3 performing feature selection methods. Mutual Information shows less scatter than the other two techniques.

We ran cross-validation tests in all datasets with three different classifiers (kNN, DT and RF) and we calculated the mean $f1$-score for each feature selection method.

Figure 3.1 shows the relation between $f1$-score and the number of features. We calculated the mean $f1$-score among datasets and classifiers, as stated in section 3.3.1.

Most methods describe nondecreasing monotony until they reach 100 features. From here, they either stabilise or shift to a nonincreasing tendency. Top 3 methods are χ^2, *GSS coefficient* and *mutual information (MI)*. Despite the apparent win of χ^2 and *GSS*, especially below 20 features, the standard deviation shows that MI test are less dispersed than the other two, as can be seen in table 3.1. This peculiarity made us opt for *mutual information*, since there is not significative difference in $f1$-score whatsoever and there is still room for improvement.

There is a neat question regarding the optimal number of features that is not answered yet. How many features can a human handle? How large can be the number of features without loosing interpretability?

In 1956, Miller [170] established precedent in what would be considered *working memory*. The article shows that 7 ± 2 is the number of chunks that a person can remember for a short time. The author referred to that number as a unitary measure but not in terms of minimum possible unit. That means that any human can handle between 5 and 9 concepts, situations, facts, melodies, etc., rather than words, movements or sounds (minimum expression). Further researches proved that this number could be even smaller [171].

At first glance, it seems reasonable to pick 9 as the “optimal number of features” for interpretability purposes, as it is the upper bound that human short-term memory seems to have. This will make models not only interpretable in theory but also in practice. However, this will only constitute a proof of concept and further researches are required to empirically support this claim.

3.4 Distinguishing Expressions

This section describes the proposed algorithm, as well as the feature ranking method we use to prioritise the evaluation and selection of the features.

Words are a tool used to abstract reality. As such, we are the ones that give sense to them, even creating new meanings. However, they require context: it is impossible to communicate effectively without sentences. Given a word, depending on the words that precede or follow it, the meaning can vary from one end to the other, and not only the order influences it but also adjectives, prepositions and even *stop words* (that are usually removed in preprocessing steps). Some expressions are prone to be interpreted in several ways but they maintain the bias to some extent.

Consequently, we rely on this to find the set of expressions that defines a class. We first order the potential candidates using a ranking method that we have called *CF-ICF*. From these candidates, we build expressions that will be used as features. Finally, we select those that meet some relevance and distinguishability criteria (recall and precision), in order to reduce the amount of features needed.

Let $X = d_1, d_2, ..., d_n$ be a set of documents where each document $d \in X$ belongs to a class C, where $C \subseteq X$. We consider S as the set of *stop words*.

Definition 3.4.1 (Expression)**.** *Given a document $d \in X$ as a sequence of words $d = (t_1, t_2, ..., t_n)$, e is said to be an expression of the document, noted as e expr x, iff:*

1. *It is not composed strictly by stop words.*
2. *All words of the expression can be found in the document and the order is preserved (e is a regular expression that matches the document).*

$$e \textit{ expr } d \longleftrightarrow \begin{cases} \exists t_i \in e : t_i \notin S \\ \wedge \\ /t_1 * t_2 * ... * t_n/ \textit{ matches } d \end{cases} \tag{3.8}$$

Definition 3.4.2 (Distinguishing expression)**.** *Given r (minimum relevance or recall) and p (minimum precision), an expression e is said to be (r, p)-distinguishing for the class C, noted as e dexpr$_{r,p}$ C iff:*

1. *Recall of e for the class C is over a given threshold r.*
2. *Precision of e for the class C is at least p.*

$$e \textit{ dexpr}_{r,p}\ C \longleftrightarrow \begin{cases} \frac{|e \textit{ expr } d_i : d_i \in C|}{|C|} > r \\ \wedge \\ \frac{|e \textit{ expr } d_i : d_i \in C|}{|e \textit{ expr } d'_i : d'_i \in X|} > p \end{cases} \tag{3.9}$$

The algorithm (3) we propose for feature extraction will compute a set of distinguishing expressions D taking into consideration that each expression needs to meet some frequency and distinguishability criteria with respect to a class. In other words, each expression e should be skewed towards a class, such that the frequency in which that expression appears in the class exceeds a certain threshold (r) meanwhile the ratio between the matches in the class and the total number of matches is above a given boundary (p).

Once we have the set of distinguishing features, we can transform each document to a binary vector (v_i) where each component v_i stands for the appearance of the i-th distinguishing expression in the document.

3.4.1 CF-ICF

We build distinguishing expressions from the words present in the class instances. Usually, there are several expressions that can accomplish our criteria, but a few of them are more useful than the rest. As we want to maximise statistical relevance and distinguishability, we proposed a ranking method based on *TF-IDF* that takes into consideration the class whose expression we are looking for.

Algorithm 3 DE feature extraction algorithm

```
Require: Training set of documents $X$, vector of labels $y$ for each document,
    minimum required precision for the expression $p$
Ensure: Set of distinguishing expressions $D$
$D \leftarrow \emptyset$
for all $d \in X$ do
    $t \leftarrow$ tokenize $d$
    remove stopwords of $t$
    stem words from $t$
    sort elements in $t$ by cficf
    add $t$ to $X^p$
end for

set $r$

for all $t \in X^p$ do
    $o \leftarrow$ QUEUE
    put all elements of $t$ in $o$
    repeat
        $e \leftarrow$ pop $o$
        $tp \leftarrow$ count $e$ matches in $\{X_i^p : y_i = True\}$
        $fp \leftarrow$ count $e$ matches in $\{X_i^p : y_i = False\}$
        $R \leftarrow \frac{tp}{|\{y_i : y_i = True\}|}$
        $P \leftarrow \frac{tp}{tp+fp}$
        if $R \geq r$ then
            if $P \geq p$ and $e \not\subseteq S$ then
                accept $e$
                $D \leftarrow D \cup e$
            else if $|e| < \alpha$ then
                n $\leftarrow e \times t$
                put all elements of $n$ in $o$
            end if
        end if
    until $e$ is accepted
end for
return $D$
```

Let X be a set of documents $\{d_0, d_1, ..., d_n\}$ where $n = |X|$. L is a binary property that can be present (or not) in each document, such that for a given document $d_i \in X$, $L(d_i) \in \{0, 1\}$. We will now define the sets $X_L^+ \subseteq X$ and $X_L^- \subseteq X$ such that:

$$X_L^- = \{d \in X : L(d) = 0\} \tag{3.10}$$

$$X_L^+ = \{d \in X : L(d) = 1\} \tag{3.11}$$

where n_L^+ and n_L^- stand for $|X_L^+|$ and $|X_L^-|$, respectively.

The function $f(t, d)$ yields the number of times that a word t appears in the document d. From here, we can define classic *TF-IDF* as follows:

$$tf(t, d, X) = \frac{f(t, d)}{\max\{f(t, d'), \forall d' \in X\}} \tag{3.12}$$

$$df(t, X) = |\{d \in X : t \text{ in } d\}| \tag{3.13}$$

$$idf(t, X) = \log \frac{n}{df(t, X)} \tag{3.14}$$

$$tfidf(t, d, X) = tf(t, d, X)idf(t, X) \tag{3.15}$$

Now, let $cf(t, L)$ be a function that returns the number of times that the word t appears in the documents of X_L^+ and let $d_L = \bigcup_{d \in X_L^+} d$ (meaning that d_L is the result of concatenating all the documents in X_L^+). We can define cf as a function of tf using the concatenated d_L as follows:

$$cf(t, L) = \sum_{d \in X_L^+} f(t, d) = tf(t, d_L) \tag{3.16}$$

In the same way, let $n_L^-(t) = |\{d \in X_L^- : t \text{ in } d\}|$ and $n_L^+(t) = |\{d \in X_L^+ : t \text{ in } d\}|$. We can also express *IDF* as follows:

$$idf(t, X) = \log \frac{n}{n_L^-(t) + n_L^+(t)} \tag{3.17}$$

Now, idf can be modified to define icf, as shown below:

$$icf(t, L, X) = \log \frac{n}{n_L^-(t) + 1} \tag{3.18}$$

$$cficf(t, L, X) = cf(t, L)icf(t, L, X) \tag{3.19}$$

The proposed ranking method can be seen in equation 3.19. It is straightforward to notice that

$$idf(t, X) = icf(t, L, x) \Leftrightarrow n_L^+ = 1 \tag{3.20}$$

$$cficf(t, L, X) = \sum_{d \in X'} tfidf(t, d, X'), \textit{ where } X' = \{d_L\} \cup X_L^- \tag{3.21}$$

We can now study the relation between both methods:

1. If t is only in one document of X_L^+ (or none), then:

$$cficf(t, L, X) = \sum_{d \in X_L^+} tfidf(t, d, X) \tag{3.22}$$

2. If t is in more than one document of X_L^+, then:

$$cficf(t, L, X) > \sum_{d \in X_L^+} tfidf(t, d, X) \tag{3.23}$$

Note that if t is in several documents of X_L^+, then *CF-ICF* only depends on the number of times that this word appears in those documents. If t only appears in X_L^+, *ICF* will be the maximum possible ($\log n$), which implies that the maximum possible value for *CF-ICF* will also be $\log n$, since *cf* is normalised between $[0, 1]$ and it will only be achieved if exists a word t that is the most frequent in the class L and it is not present in the documents of X_L^-.

Figures 3.2 and 3.3 describe the behaviour of *CF-ICF* when keeping cf and icf fixed, respectively. It shows that is more important for the expression to be distinguishing than relevant (icf decreases faster than cf grows).

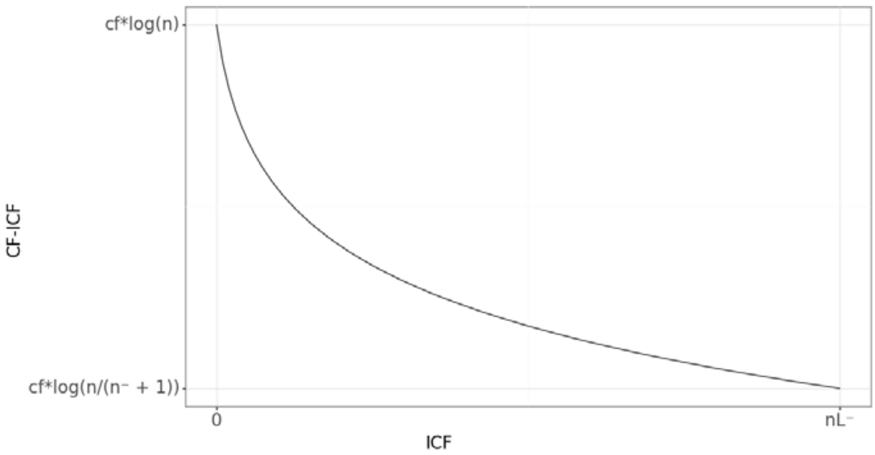


Fig. 3.2.: Behaviour of the *CF-ICF* ranking method when keeping cf fixed. It can be seen that the function decrease faster than a linear one.

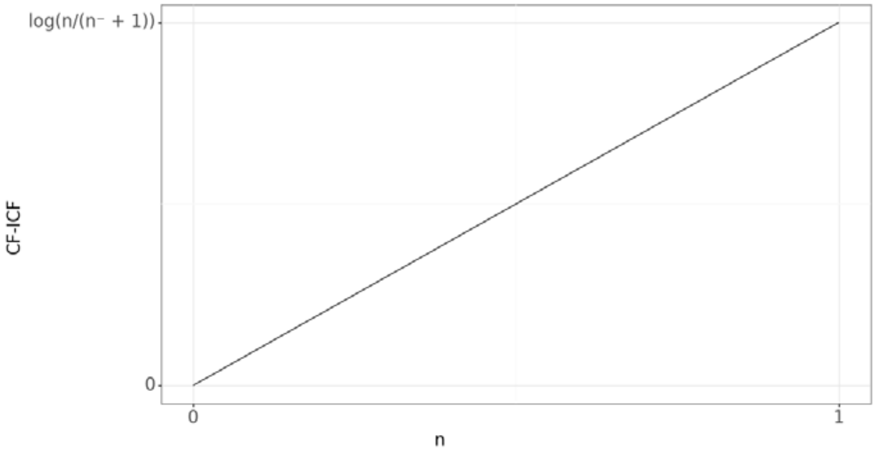


Fig. 3.3.: Behaviour of the *CF-ICF* ranking method when keeping icf still, showing that function grows linearly.

3.5 Results and Discussion

We evaluated our model in terms of accuracy and interpretability of the models produced. In this section, we review the results of goodness metrics and compare the complexity of the models between our proposal and other approaches.

3.5.1 Accuracy Metrics

We present in this section the experiments that we conducted to compare the performance in terms of accuracy of our algorithm against *bag of words*, *TF-IDF*, *bigrams* and *term cooccurrence* plus mutual information.

Test were run over the four dataset and conditions described in section 3.3.1 following a k-cross fold validation approach, where k was adjusted in order for each partition to be around 1000 tweets. Each model was trained with one partition and validated against the rest. This decision was taken considering that this is a viable size to build and tag a dataset without the need of too much resources. Any model that can work under this circumstances will have a fair generalisation capacity.

Figure 3.4 shows mean $f1$-scores proving that our proposal was one of the best performing approaches, along with bigrams. It is possible to notice that, in the Spanish dataset, our methods still showed consistency meanwhile models that used bigrams got erratic behaviour. DE kept low deviation throughout the four datasets, unlike other methods.

As we previously established, we are interested in features underlying to a class. In order to define clusters, it is preferable that we have characteristics directly associated with them than having the features that univocally describe it (which is practically impossible). That means we prefer *precision* over *recall*. Figure 3.5 shows once again that the best two models are the ones based on bigram and DE features.

Finally, we ran these tests using three different classifiers: k-nearest neighbours (kNN), decision trees (DT) and random forests (RF). Figure 3.6 show mean $f1$-scores for cross validation tests using DE features. Results are very similar for DT and RF, being kNN the most irregular.

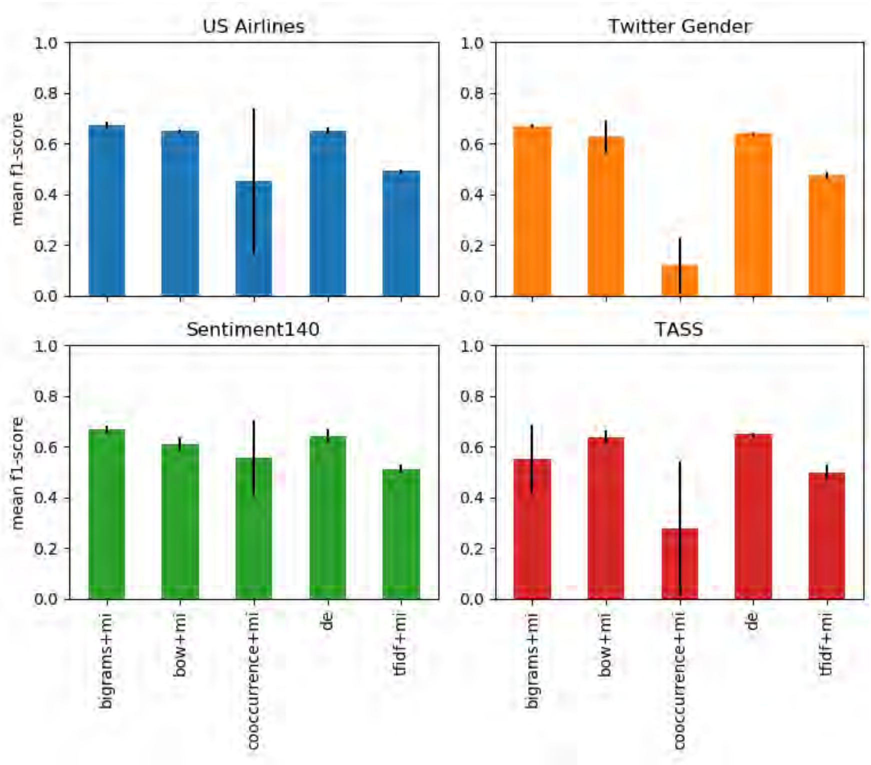


Fig. 3.4.: Mean $f1$-score and standard deviation for selected features over the four dataset, using k-nearest Neighbourhs, Decision Tree and Random Forest classifiers. We discarded *term cooccurrence* and *TF-IDF* for further tests due to their poor baseline performance.

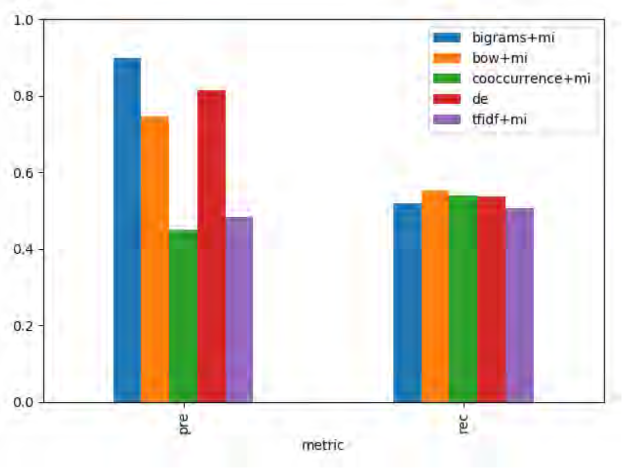


Fig. 3.5.: Mean precision (pre) and recall (rec) for each kind of document encoding, using kNN, DT and RF classifiers. Bigrams and our proposal show better performance than the rest, especially regarding precision.

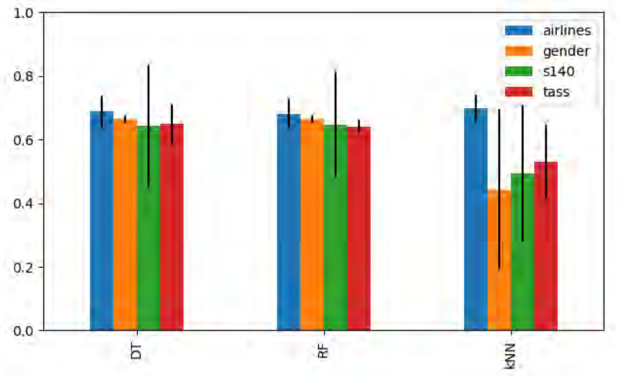


Fig. 3.6.: Mean $f1$-score and deviations for crossvalidation results using DE features and three different classifiers: decision tree (DT), random forest (RF) and k-nearest neighbours (kNN).

DE

$$\neg\textit{food} \;\wedge\; \neg\textit{favourite} \;\wedge\; \textit{happen} \;\wedge\; \textit{(insult)} \;\wedge\; \textit{music} \;\wedge\; \neg\textit{wonder} \quad\implies\quad \textit{male}$$

TF-IDF

$$\begin{aligned}&\textit{hoover} \leq 0.5 \wedge \textit{texture} \leq 0.5 \wedge \textit{vitamin} \leq 0.5 \wedge \textit{malta} \leq 1.5\\ &\qquad \wedge \textit{vitamin} \leq 2.5 \wedge \textit{malta} \leq 0.5 \wedge \textit{supply} \leq 0.5 \wedge \textit{texture} \leq 1.5\\ &\qquad \wedge \textit{vitamin} \leq 1.5 \wedge \textit{(brand name)} \leq 0.5 \wedge \textit{(political party)} \leq 0.5 \implies \textit{male}\end{aligned}$$

Tab. 3.2.: Examples of mean-length rules generated by a decision tree trained over the Twitter Gender dataset. The first one corresponds to distinguishing expressions meanwhile the second belongs to a model trained with *bag of words* and *mutual information*. Insults, brand names and political parties have been removed from the rules.

3.5.2 Interpretability Evaluation

We have already established that there is no general consensus in how to measure interpretability, especially between different models [101]. However, it is straightforward to compare models using the same classifier. In this section, we are going to evaluate the improvements on interpretability by analysing the particulars of trained models.

k-nearest Neighbours

There is not much explanation needed for this classifiers. Each instance is represented with vector as large as the number of features. Whenever we need to classify a new element, it will measure the distance to all the instances in the training dataset. After keeping the k lower distances, the class of the new element is determined by the mode of those k elements.

Since there is no parameters or weights to learn, we can state that the less features the model has, the more interpretable the model is. Table 3.3 shows conducted tests. DE biased features kept the highest score with the less number of features. Specifically, it achieved a mean of 0.5608 $f1$-score with 7 features.

kNN with No. features:	5	7	9	20	50
BoW	0.5277	0.5315	0.5143	**0.5481**	0.5318
Bigrams	0.3440	0.3429	0.3503	**0.4637**	0.2618
DE	0.5569	**0.5608**	0.5243	0.5595	0.5512

Tab. 3.3.: $f1$-score for kNN classifier using several number of features and methods. DE has the best ratio between score and number of features.

A model with 7 features would be easier to interpret than another which uses 20 of them (optimal number of features for BoW and bigrams with the default parameter set).

Decision Tree

Decision trees are widely used, even outside machine learning applications. They are easy to build and interpret, since following a path through the tree is straightforward. When training the model, the data is divided into disjoint subsets following certain criterion. Each path starts in the root of the tree and finish in a leaf.

The number of features is not the only thing that we need to take into account anymore. The difficulty of interpreting the model increase as the tree does, since each leaf is a new path. Table 3.4 shows that DE features result in trees one order of magnitude smaller.

We noticed that, due to biased features, there were subtrees where the path would not alter the result of the classification (whatever the path you take it would lead to the same result). Consequently, we designed a lossless algorithm to prune the trees: for each subtree, if all its leaves are of the same class, prune the subtree. This reduced the path length and avoided unnecessary steps in the rule. We show in table 3.4 the attributes of the trees before and after applying the prune.

Random Forest

Random forests add one more layer that complicates the interpretation. They fit a certain number of trees from random samples of the data. The final classification are the result of a voting process between all of them. Since trees are trained from random samples, it may avoid local extrema.

DT with:	No. of leaves	Mean path length	$f1$-score
BoW	366	12.00	0.627711
Bigrams	389	12.16	0.669784
DE	49	8.10	0.642069
Pruned DT with:	No. of leaves	Mean path length	$f1$-score
BoW	265	10.26	0.627711
Bigrams	284	10.37	0.669784
DE	36	7.44	0.642069

Tab. 3.4.: Description of the complexity of the decision trees resulting of training with 9 features. DE generate far more simple trees than the other two methods.

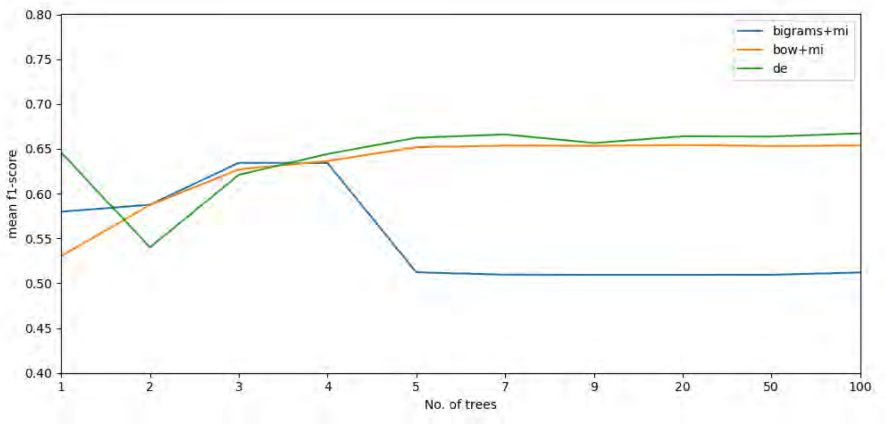


Fig. 3.7.: Behaviour of $f1$-score while changing the number of trees in the forest. The constant score above 5 trees points to limitations in the number of features.

After testing with different number of trees, results showed that the final score does not change above 5 trees, as can be seen in figure 3.7. This points to a limit in the information that 9 features can carry. In this regard, it does not make much sense to keep more than 5 trees.

As for the underlying trees, they described the same behaviour as if they were trained alone. This were the expected results since they are trained over random samples. Table 3.5 shows the specific attributes.

RF with:	Mean No. of leaves	Mean path length	$f1$-score
BoW	234.3	11.6	0.5736
Bigrams	227.4	11.27	0.5122
DE	39.8	7.67	0.5722

Pruned RF with:	Mean No. of leaves	Mean path length	$f1$-score
BoW	194.5	11.2	0.5736
Bigrams	202.8	11.03	0.5122
DE	33.6	7.45	0.5722

Tab. 3.5.: Description of the complexity of the decision trees underlying a random forest of 5 trees trained with 9 features. Generated trees have approximately the same attributes than when trained alone.

3.6 Conclusions

Nowadays, Machine Learning (ML) has many applications. Not all of them imply high-stake decisions, but there is a subset of problems in which the use of a model may have social consequences. Such is the case of health-related solutions, where ethics committees play an important role; insurance policies, where the risk factor of every applicant is evaluated; financial institutions, that take into account the financial profile of the individual before granting credits; or even pre-emptive closing of Social Networking Sites (SNS) accounts.

Such scenarios require that models are accurate, auditable and fair. In order to guarantee these qualities, algorithms are required to be interpretable. Current state-of-the-art solutions are black-box models focusing on accuracy and efficiency rather than interpretability. Research trend suggests that models can be explained using surrogate interpretable models that mimic the behaviour of the main model. However, it comes at a price. The interpretable model can be analysed, but it is the main model the one that is going to be used in production, therefore conclusions may be problematic or misleading [41].

Unfortunately, not all interpretable algorithms can be easily comprehended. The complexity of the model may result in an analysis, by far, too difficult for any human to handle. It is necessary to reduce model complexity up to a point whether any expert can study and understand it.

We believed that it was possible to do it using *distinguishing expressions*, that are class predictors in the form of sequence of words. Candidates are generated using a custom ranking strategy (CF-ICF), and features are selected to meet relevance and distinguishability criteria.

Our results show that our approach is able to produce models as accurate as the baselines while reducing model complexity. In terms of features, methods like kNN are able to slightly improve $f1$-score with less than half the features used with bag-of-words or TF-IDF document encodings. Decision tree models require a tenth of the rules when using our proposal, with a 30% reduction in the length of each rule. Random Forests show no improvement with more than five trees, which suggest that such amount is sufficient to model the search space when using nine features. Yet, they benefit from the same complexity reduction than decision trees.

Although results are promising for Twitter, more experiments are necessary to determine if Distinguishing Expressions are suitable to be used in non-microblogging contexts. Limitations like computational cost of candidate generation and selection may impact results due to a combinatory explosion.

4 A Quantification Approach to Bypass Aggregation Bias

Objectives

1. Understand types of bias that may be present in Social Networking Sites (SNS) data.
2. Study applicability of quantification models to SNS data.
3. Explore performance of standard quantifiers to establish the influence of different types of bias in the training and validation sets.
4. Propose new adjustments of the prevalence count that take advantage of spatial and/or temporal features.

4.1 Introduction

Due to Twitter user prompt response to events, the social network has been traditionally used to monitorise opinion, real-time events and even to offer rapid response to natural catastrophes (e.g., [172, 173, 174, 175, 176, 177]).

Several techniques are applied to build solutions to real-time monitoring, each one of them with different advantages. However, the major handicap of real-time analysis is that, as little time as it takes to process a document, there are thousands of them being published each second. Depending on the situation, it may be preferred that the system is quick and sensible (prediction of a terrorist attack) or that it does not cause any false alarm (spam detection). Therefore, the hunt of *a perfect processing pipeline* is replaced by *the quickest*, *the most sensible* or *the best-performing* model... but not all of them at once [178].

There are real-time systems focused on obtaining individual evaluation of documents (e.g., SPAM detection [179]). But there are many situations in which the target is to predict or analyse the trend and to estimate class volumes. The most popular approach towards that end consists in a machine learning pipeline with a data collection phase, several preprocessing steps and a more or less complex classifier [180, 181, 182, 183]. Recent research has focused on classification accuracy, improving each and everyone of the steps in the classification pipeline, using intricate techniques that are, most often, black-box models. Unfortunately, once documents are classified, they rely on a *classify and count* aggregation mechanism, which is the most simple approach.

In 2016, Gao and Sebastiani [46] suggested that, in such cases in which the target is to predict the class prevalence values, we are doing it wrong. The task of estimating population distributions across a number of classes (or class prevalence values) is called *quantification*. They proposed an approach to estimate sentiment counts using *quantifiers*, and proved that they perform better than state-of-the-art classification-oriented models. This change of perspective is not only useful for static datasets but also for real-time series, in which data is aggregated using time-based windows to analyse the evolution of trends over time [184, 185, 186, 187].

In this chapter, we explore the utility of quantifiers in Twitter using the PHEME dataset [188]. The aim is to perform an exploratory analysis of avaliable quantification methods and their suitability to deal with biased SNS data. To do that, we will study how the training sample affects the performance when there is a *prior probabililty shift* in validation; analogously, we will study if a prior probability shift in training dramatically affects the performance of estimating class prevalences in a steady test sample. Lastly, we will measure the impact of *covariate shift*, both in train and test samples. Ultimately, we will study if the use of similarity measures can improve quantification performance.

The rest of this chapter is organised as follows. Section 4.2 explores related work in the field. Section 4.3 introduce the quantification task and a general taxonomy of quantifiers, as well as error measures and evaluation protocols. In Section 4.4, we try to empirically answer the questions related to the quantifiers behaviour when faced with different kinds of bias, and we present our results in Section 4.5. Section 4.6 present our proposal of similarity-based quantifiers. Section 4.7 puts into practice what we have learn to deal with a sentiment estimation task in one of our datasets. Ultimately, we present our conclusions in Section 4.8.

4.2 Related Work

Twitter present different kind of biases that is necessary to take into account when conducting studies over their data. Some of them are related to sampling mechanisms; others are related to the natural bias of their population or to secondary effects of the recommendation algorithms. In the following sections, we present related work that tried to characterise these biases.

4.2.1 Sampling Bias

In Machine Learning (ML), there is a general assumption that a training sample is *independent and identically distributed*:

- **Identically distributed**. Data is supposed to be the result of the same generation mechanism, therefore there would not be any difference between the training sample and the real-life data.

- **Independent**. Instances are independent from one another, hence the generation of an instance i would not affect in any way the generation of another instance j.

In other words, given a training sample trn that is composed of N instances $\{(X_i, y_i)\}_{i=1}^{N}$, these are supposed to be drawn from a distribution $P(X, y)$ such that $\forall i \in [1, N], (X_i, y_i) \sim P(X, y)$. We call it the *i.i.d. (or iid, or IID) assumption* [189, 190].

Dataset not constrained to the i.i.d. principle may present drifts that would affect the model and, consequently, its performance. These can be characterised as one of the following three types [191, 192]:

Covariate shift Independent variables distribution skew. The distribution of the features in test differ from the distributions seen in train. This usually happens when the training sample is not representative enough.

$$P_{trn}(X) \neq P_{tst}(X) \wedge P_{trn}(Y|X) = P_{tst}(Y|X) \tag{4.1}$$

Prior probability shift Label distribution skew. The distribution of the labels in test differ from the distributions seen in train. This usually happens when the training set is artificially balanced however real-case scenarios do present imbalance.

$$P_{trn}(Y) \neq P_{tst}(Y) \wedge P_{trn}(X|Y) = P_{tst}(X|Y) \tag{4.2}$$

Concept shift Feature distribution remains the same however their association with the labels is not the same than the one seen in the training set. The reverse case is also called concept shift (see eq. 4.4).

$$P_{trn}(X) = P_{tst}(X) \wedge P_{trn}(Y|X) \neq P_{tst}(Y|X) \tag{4.3}$$
$$P_{trn}(Y) = P_{tst}(Y) \wedge P_{trn}(X|Y) \neq P_{tst}(X|Y) \tag{4.4}$$

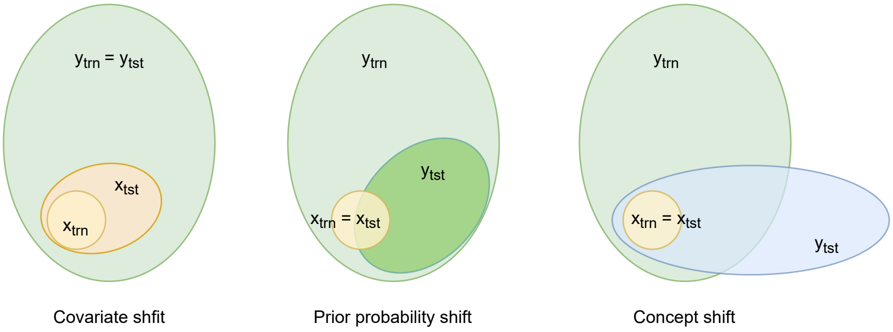


Fig. 4.1.: Venn diagrams representing covariate shift, prior probability shift, and concept shift.

4.2.2 Twitter API Bias

For many years now, Twitter has been a very popular platform among researchers. As opposed to other Social Networking Sites (SNS), their content is public by default. In 2006, they release their first public API, making it possible for developers to access their data easily. However, there were (and still are)[1] extensive limitations regarding (1) which documents can be retrieved and (2) the quantity of documents that can be retrieved (*quota* hereafter).

In 2012, they release the v1.1 version of their API. There were two main versions: Search (or Filter) API, which allowed querying the Twitter 7-day archive; Streaming API, that allowed for a continuous 1% document retrieval through a filtered stream. Both are subject to sampling mechanisms designed to *"serve consumer use cases"*[2]. In other words, there is an intentionally-introduced bias that may alter research results [193]. In order to bypass the bias, there were also several enterprise alternatives (that significantly improved the quotas or even gave access to the full stream and archive of Twitter data, such as *Decahose* or *Firehose*), but their elevated price did not make them suitable for (most) research purposes.

In November 2021[3], Twitter released their second version of the API. Along with major changes in its implementation, they introduce new access levels, including the *Academic Research Track*. This tier offers new capabilities and significant improvements in the API quotas. Moreover, it reduces the sampling bias from the Search API, and it becomes possible to remove it completely since approved researchers may access full-archive search endpoints. Despite that there are two kind of streams (filtered and sampled), they are unfortunately still limited to 1% of the tweets.

It is may not possible to expect IID-data from non-uniform sampling methods such as Twitter's. In fact, there is previous work that tried to characterise and measure the particular bias of the sampling mechanism.

[1] Twitter API limits vary overtime and can be consulted in their developer portal (`https://developer.twitter.com/en/docs/twitter-api/rate-limits#v2-limits`).

[2] Quoted from their site (`https://developer.twitter.com/en/products/twitter-api/academic-research/product-details`).

[3] Regardless of Twitter API v1.1 deprecation, it has been active throughout the most part of the development of this thesis. All our datasets were retrieved while the *Academic Research Track* did not exist, therefore they present all the sampling biases included in v1.1.

Morstatter et al. [194] studied the difference between the Firehose and sampled Twitter streams. They collected the result from the same query against both APIs and they proved that the sampled streams are representative enough of Twitter's activity. For example, top n hashtags (when n is relatively large) do not present significant differences; the probability distribution of topics (extracted using LDA) are similar; at least 50% of most-relevant users are detected. However, this only happens when there is a large coverage of the query by the Streaming API, and it is usually misleading for small samples. Later, Morstatter et al. [195] studied different samples from old Twitter API to determine that the Sample API is uniformly sampled from the firehose data but they confirm that there is a sample bias in the Streaming API.

However, the Sample API present other downsides. Morstatter et al. [196] analyse the set of documents returned by this API to determine that the sampling mechanism is time-based and therefore highly manipulable by bots, since it is possible to programatically control the publication time. Subsequently, Pfeffer et al. [197] proved that, albeit different, *Decahose* sampling is also time-based and may be easily influenced.

Joseph et al. [198] compared simultaneous samples of the Streaming API collected using the same keywords and determined that, on average, 96% of retrieved documents were present in all the samples. Those that were exclusive from one sample did not alter the popularity or the structure of the user network. In fact, they conclude that the 4% difference is the result of *technical artefacts* and not a systemic bias.

González-Bailón et al. [10] compared three different Twitter samples to determine the nature of the introduced bias. They conclude that the Streaming API introduces a non-uniform bias that is reduced with long queries that include non-popular hashtags (that are otherwise ignored due to the *importance* of popular ones). In the same manner, users that publish more tweets have a better chance to be included in the sample. This particularly affects the *mentions* graph, since users that are mentioned the most do not necessarily respond to these. On the contrary, users that *retweet* content are usually more active in the social network. Hence, the bias of the mention graph seems to be larger than the bias of the *retweet* graph.

Morstatter and Liu [193] determined that the Streaming API shows key differences in terms of frequent hashtags and discussed topics. They proposed a technique to detect if there is a bias in a sample without requiring access to the full data, using bootstrapped samples from the Sample API and comparing it to the results of the Streaming API. They also proposed a way to reduce collection bias.

4.2.3 Recommendation Systems Bias

At the same time, well-known issues are derived from the use of recommendation systems in SNS. Historically, platforms used to present all new content to its users, until they turned popular enough for the information flow to be unmanageable by any human. Arguably, most recent case was Instagram [199]. Recommendation systems were introduce to rank content regarding user's interest, in order to show first the most interesting updates. Although these were good-intended (and necessary) solutions, they proved to distort perception of reality [200].

There is also a measurable social impact from these algorithms [201], some of them derived from the reduced possibility that unpopular content is recommended and others related to the training target of these. Such seems to be the case of YouTube. Back in 2017, Guillaume Chaslot (former Google employee) accused the company of optimising for *watch-time* instead of relevance. In other words, Youtube allegedly recommends the videos that are more likely to keep the user consuming content from their platform [202]. This is ground for additional problems, such as that the recommendations boost misinformation and conspiracy theories [203]. This kind of effects that recommendation system have on content popularity have been thoroughly studied (e.g., [204, 205, 206]). According to Mark Zuckerberg (founder and CEO of Facebook), "*one of the biggest issues social networks face is that, when left unchecked, people will engage disproportionately with more sensationalist and provocative content*" [207]. This constitutes a major handicap for Content Filtering recommendation algorithms, since they are based on other's people interests.

Nonetheless, it is certain that most relevant recommendation systems are kept secret. There is no information about the target functions of these algorithms or about the training data that is used. Initiatives like Algotransparency[4] are trying to expose the effects of these algorithms, and claim that APIs should be

[4] `https://algotransparency.org`

made available to allow researchers to monitor their behaviour[5]. Twitter is not exempt of such problematic. Usher et al. [208] explain that Twitter may be amplifying gender biases in journalism, specially since male Washington-based journalists tend to interact more with their own gender. Badawy et al. [209] analysed the attempt of bot and troll accounts to manipulate public opinion regarding 2016 USA Presidential Elections, and Lee and Xu [210] confirmed that *attacks* (sensationalist content) got more retweets.

4.2.4 Population Bias

Twitter population is known to have strong biases towards male users and young people. According to datareportal [211] and financesonline [212], there are 229 million active users, with the top five countries being US, Japan, India, Brazil and Indonesia. Most of the users are older than 18 but younger than 34, and there is a clear gender bias: between 68.5% and 71.2% of the users are male, however females under 24 years old prefer Twitter over males in the same age group.

Previous work confirm that Twitter is a non-representative sample of the population. Mislove et al. [213] conducted a study regarding Twitter demographics in the US, using user self-reported data. They demonstrate that there is an overrepresentation of population in populous counties and that there exists a male bias in the social network.

Filho et al. [214] ran an experiment using data from Brazilian elections to check if it was possible to predict the outcome of the elections using Twitter data. They found that males are overrepresented, as well as the young population (up to 25 years old). After applying their methodology, they obtained good prediction outcomes in 4 out of 6 cities.

Kounadi et al. [215] tried to create a model to predict burglaries and robberies. They assert that geo-located tweet datasets are not able to replace population models by themselves, as they may be biased towards more active users, and suggest that they may be used jointly with other data such as census.

[5] From their manifesto, available at `https://www.algotransparency.org/our-manifesto.html`

Bermingham and Smeaton [216] used data regarding Irish elections to monitor political sentiment and predict the outcome of the elections. They concluded that their proposal accuracy was not enough compared to traditional polling methods, and that their study would require further analysis to determine the source of such inaccuracy.

Graells-Garrido et al. [217] performed an experiment to check whether the abortion debate in Chile was well represented in Twitter. They confirmed the bias in Twitter population, however they suggest that it can be used if they are paired with other demographic attributes.

All in all, it is well known that Twitter has a strong bias in its population that needs to be taken into account when conducting studies on its data.

4.2.5 Quantification in Twitter

There is little work related to the quantification task in Twitter. It is important to distinguish between quantification as in *determine the specific sentiment values* and quantification as in *estimating class prevalence values*. In this chapter, we focus on the latter.

To the best of our knowledge, Gao and Sebastiani [218] were the first to try a quantification approach to estimate sentiment classes, later extended in Gao and Sebastiani [46]. They detailed a wide range of techniques and evaluation metrics, as well as many reasons to encourage the scientific community to adopt better approaches than *classify and count* to estimate prevalence values. A few years later, they revised their evaluation protocol in Moreo and Sebastiani [219].

Vilares et al. [220] described their approach using neural networks and SVM with a custom loss function (*Kullback-Leibler Divergence*, or KLD [221]).

Quantification has not been widely adopted yet, although there are many reasons and applications [222].

4.3 Quantification

Quantification is the task of predicting prior probabilities of an unlabelled sample. Attending to the nature of the quantifiers, we can distinguish between [45]:

- Aggregative quantifiers, that use an underlying classifier.
- Non-aggregative quantifiers, that use other techniques that do not rely on classification models.

According to [45], and with respect to the mechanics of the quantifier, there are three groups in which we can classify current approaches for quantification. These are (1) *classify, count and correct*, (2) algorithm adaptations and (3) distribution matching.

Notation

We use standard notation for ML and the one proposed in González et al. [45]. $D = \{(x_i, y_i)\}_N^{i=1}$ is a dataset of N individuals, being x_i an encoding of an instance and $y_i \in \mathcal{L}$ its class, where $\mathcal{L}$ is the set of possible classes. Binary classes are usually denoted as $\{+1, -1\}$. D^+ and D^- stand for the documents that belong to the positive and negative classes, respectively. We use the caret ($\hat{}$) symbol to denote *estimation*.

4.3.1 Classify, Count and Correct

Classify, count and correct methods rely on a classification task of the instances in the unlabelled sample. The predictions are used to estimate the prevalence of each class in the sample; after that, a correction is applied to deal with the bias of the classifier.

Classify and Count (CC) It trains a classifier in order to produce an estimation of the classes for the unlabelled sample [223]. This estimation is later used to estimate class prevalences. No corrections are applied. This is the most straightforward method nevertheless it is not a good quantification method, since i.i.d. principle must prevail in order for it to work.

Probabilistic Classify and Count (PCC) It follows the same scheme than CC but using a probabilistic classifier [224]. The idea is to obtain classifier posteriors and estimate the prevalence for each class as the average of such posteriors.

Adjusted Classify and Count (ACC) It relies on a correction applied to the CC estimates. Such correction is computed using the confusion matrix, which requires validation sets that should be obtained from the training sample (e.g. using cross-validation). The idea is to apply a linear transformation to the class estimates using equation 4.5.

$$\hat{p} = \frac{\hat{p_0} - fpr}{tpr - fpr} \tag{4.5}$$

where $\hat{p_0}$ is the estimate of CC, tpr stands for *true positive rate* and fpr for *false positive rate*.

Probabilistic Adjusted Classify and Count (PACC) Which is a straightforward evolution of PCC with the same corrections applied in ACC [224]. Note that true positive and false positive rates are substituted for their probabilistic versions, *true positive probabilistic average* and *false positive probabilistic average*, respectively (see equation 4.6 *et seq.*).

$$TP^{pa} = \frac{\sum_{i \in D^+} P(y_i = +1|x_i)}{|D^+|} \tag{4.6}$$

$$FP^{pa} = \frac{\sum_{i \in D^-} P(y_i = +1|x_i)}{|D^-|} \tag{4.7}$$

with the adjusted estimated prevalence being

$$\hat{p} = \frac{\hat{p}_0 - FP^{pa}}{TP^{pa} - FP^{pa}} \tag{4.8}$$

As a general rule, CC and PCC will perform reasonably well when the class prevalence values of the unlabelled sample are similar to the ones seen in the training sample. In the event that there is a noticeable distribution shift, their adjusted versions (ACC and PACC) will yield better results, since they are aware of the bias of the underlying classifier.

Adjusted methods present one additional drawback when the training sample is highly imbalanced [45]. In this case, when there are few positives, the classifier will try to predict the majority class, hence reducing the fpr at the expense of the tpr. This result in excessive corrections, which can be mitigated using threshold selection techniques for classification [225].

4.3.2 Algorithm Adaptation

Algorithm adaptation consists on modifying existing classification methods to the task of quantification, for example by employing loss functions related to quantification instead of classification.

Quantification Trees Which are an adaptation of decision trees to deal directly with the quantification task [226]. They do so by choosing a measure that takes into account the false positives and false negatives to select the best splits for quantification.

Instance-based Quantification Barranquero et al. [227] proposed a quantifier method based on a similarity neighbourhood in the same fashion than the well-known k-nearest neighbours (kNN) algorithm.

Ensembles Pérez-Gállego et al. [228] proposed an ensemble for binary quantification in which they generate samples with different prevalences that they will later use to train different classifiers. An aggregation of the estimates yields the final output of the ensemble.

4.3.3 Distribution Matching

Distribution matching approaches try to match the distribution between the train sample and the unlabelled one, in order to adjust the classifier to match the class distribution drift.

Expectation-maximization Which is based on the idea that probabilistic classifier posteriors can be modified to obtain new priors that maximise the likelihood (that is the case of Expectation-Maximisation (*Saerens-Latinne-Decaestecker*) (EMQ) [229], a.k.a. *Expectation-Maximization Quantifier*). Thus, the classifier can learn on a biased sample and its outputs will later be corrected without requiring more training.

Iterative Methods Vucetic and Obradovic [230] proposed a bootstrapping strategy to initially train a model on the training sample and obtaining posterior probabilities. After that, new samples are generated from the train and the process is repeated until stopping criteria are met (convergence).

Mixture Models That adapt the distribution of the unlabelled sample by matching the outputs of a probabilistic classifier with the mixture of the distributions obtained from the training sample [231].

4.3.4 Evaluation Protocols

When evaluating classifiers, a dataset D is split into train and test subsets. Training instances are fed to the classifier for it to learn upon them and, after that, the classifier is asked to predict the labels of the test sample. Performance is measured comparing classifier predictions with the ground-truth labels.

In the same manner that classification instances are individuals, quantification instances are samples of individuals. In order to check quantification performance, the dataset D is divided into train and test subsets. Train instances are used to feed the quantifier so it can learn. However, in order to check quantification performance, a set of samples are drawn from the test subset. We present in this section the different techniques that are currently used to build these samples [219].

Artificial Prevalence Protocol (APP) Samples are generated using different prevalence values that vary artificially. First, a sample size s, a grid of values G and a number of repetitions $r \in \mathbb{Z}^+$ is determined (e.g., $G = \{.01, .02, \cdots, .99\}$ with $s = 100$ and $r = 1$). Then, given a set of individuals, we generate r samples of s individuals for each of the elements in G, until we have a set of $n = r|G|$ samples. The quantifier is asked to estimate the prevalence values for each of the n samples, and the estimations are compared to the known prevalence values of the samples to evaluate quantification performance. In the case of single-label quantification, the grid G is explored for every possible class in $\mathcal{L}$. Take into account that there may be classes in the test subset whose prevalence values do not allow to explore the full grid of prevalence points G for specific sample sizes without replacement. Although this protocol is exhaustive, it has been criticised because there may be testing samples whose prevalence values are not realistic, therefore they would not be found in real-life data [219].

Natural Prevalence Protocol (NPP) In NPP, samples are generated using i.i.d. sampling from the original distribution. Since this may be complicated (or even impossible), its most often implementation consist in drawing n samples of size s from the test subset, without artificially varying the prevalence of each class. As a result, the prevalence for each class in the generated samples will be very close to the ones seen during training (low drift). Hence, estimating the prevalence of each sample would be easy, and naïve *classify and count* approaches would perform reasonably well. NPP was the chosen protocol in Gao and Sebastiani [46, 218], which was later revised in Moreo and Sebastiani [219].

In conclusion, APP is, normally, the chosen protocol to evaluate quantification performance [223, 224, 232].

4.3.5 Evaluation Metrics

There are several standard error measures that are being used to evaluate the performance of quantifiers [46, 227].

Bias This measure is applied to binary quantification problems in which the symbol of the error is important (to check whether the model is under- or overestimating the prevalence value).

$$Bias(p, \hat{p}) = \hat{p} - p \tag{4.9}$$

Absolute Error (AE) The absolute error stands for the absolute difference between estimated class prevalence values and true prevalence.

$$AE(p, \hat{p}) = |\hat{p} - p| \tag{4.10}$$

Squared Error (SE) The square error is used to penalise large deviations in the predictions.

$$SE(p, \hat{p}) = (\hat{p} - p)^2 \tag{4.11}$$

Relative Absolute Error (RAE) The relative absolute error is used to penalise those small errors if the true class prevalence is small. Since it may be undefined in some cases, it is a common practice to add a smoothing constant.

$$RAE(p, \hat{p}) = \frac{|\hat{p} - p|}{p} \tag{4.12}$$

***Kullback-Leibler* Divergence (KLD)** It is also known as *normalised cross-entropy*. However KLD is the *de-facto* standard for binary quantification, it is not a true metric and it is less interpretable than the ones presented above [227].

$$KLD(p, \hat{p}) = \sum_{l \in \mathcal{L}} \log_e \frac{p_l}{\hat{p}_l} \tag{4.13}$$

There are also mean versions of AE, SE, and RAE, that are called Mean Absolute Error (MAE), Mean Squared Error (MSE) and Mean Relative Absolute Error (MRAE), respectively. MAE is more robust when facing outliers, however MSE

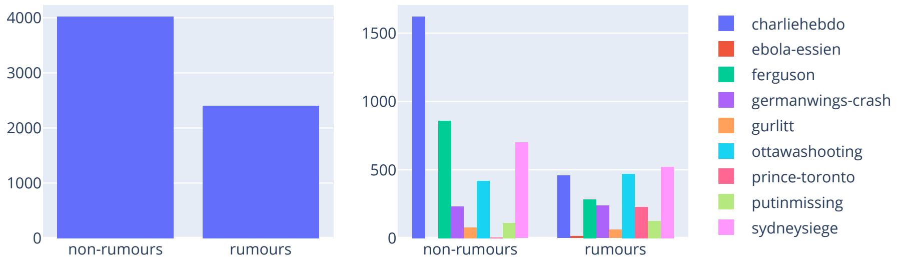


Fig. 4.2.: PHEME class histograms.

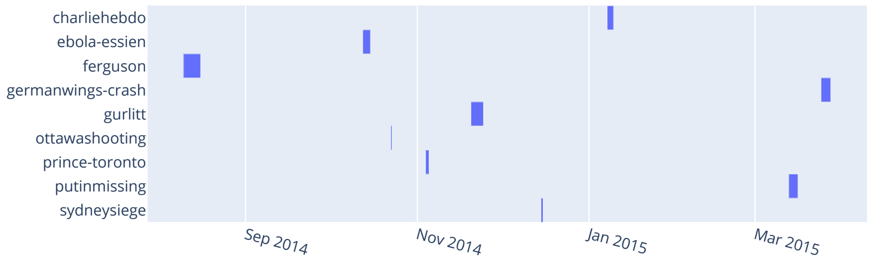


Fig. 4.3.: Tweet activity for each topic.

tends to penalise big differences. MRAE relativise the error, since it is not the same a 0.1 absolute error when the class prevalence is 0.95 than when it is 0.05.

4.4 Experimental Work

In order to check whether *adjusted* aggregation perform better than *classify and count* methods (which is the naïve approach, yet the most popular in the literature), we used the PHEME dataset [188]. It presents tweets classified as *rumour* or *non-rumour*. There are nine topics named *Charlie Hebdo*, *Ebola Essien*, *Ferguson*, *Gemarn Wings Crash*, *Gurlitt*, *Ottawa Shooting*, *Prince Toronto*, *Putin Missing* and *Sydney Siege*. Topics do not present overlapping timestamps (see figure 4.3), therefore it is important to keep this feature out of the classifier's visibility to prevent data leakage. Figure 4.2 presents histograms of class counts, both aggregated and on a per-topic basis.

Tab. 4.1.: Experimental parameter setup for evaluation protocols

	prior probability shift				covariate shift	
	train		test		train	test
parameter	APP	NPP	APP	NPP	APP	APP
prevalence points	101	-	101	-	9	9
repeats	5	505	5	505	1	1
sample size	3000	3000	100	100	2500	100

We used `python 3.10.5`, `sklearn 1.1.1`, `quapy 0.1.6`, `numpy 1.23.0` and several other non-critical utilities. We compared the performance of CC, PCC, ACC, PACC and EMQ, using Logistic Regression (LR) as base classifier. LR was chosen because it outputs fairly well-calibrated posterior probabilities [233]. We used both NPP and APP evaluation protocols (please refer to table 4.1 for parameters, page 125).

Tweets are 300-dimensional truncated SVD embeddings obtained from TF-IDF encodings. We performed a simple 80% split for training, with the remaining 20% left for test purposes. Tests instances are never used during the training phase, either by the classifier or by the quantifier. Whenever a validation split is required (i.e., when ACC and PACC compute the adjustment), it is obtained from the training set. Time sampling was performed using 30-minute windows. Since tweets of different classes are not being uniformly published (see figure 4.5), time sampling will result in a *prior probability shift* (please refer to table 4.4 to see normalised histogram of class prevalence values per window, page 126). This is the kind of problem where *adjusted classify and count* quantifiers excel. Since topics naturally evolve over time, time-based windows may also present *covariate shift*. We ran binary quantification tests (using just the *rumour/non-rumour* classes, however we performed artificial sampling to vary topics prevalence to input *covariate shifts*). In order to measure performance, we used AE, RAE, SE and KLD. Most figures present results in terms of AE, since it can be easily interpreted.

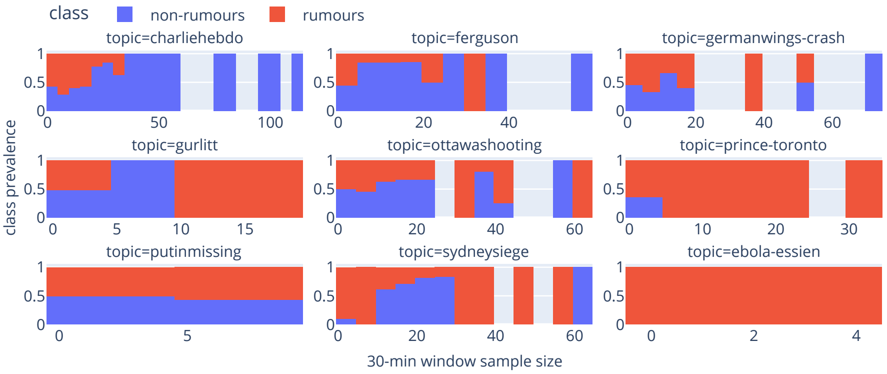


Fig. 4.4.: Class prevalence values when performing 30-minute window sampling.

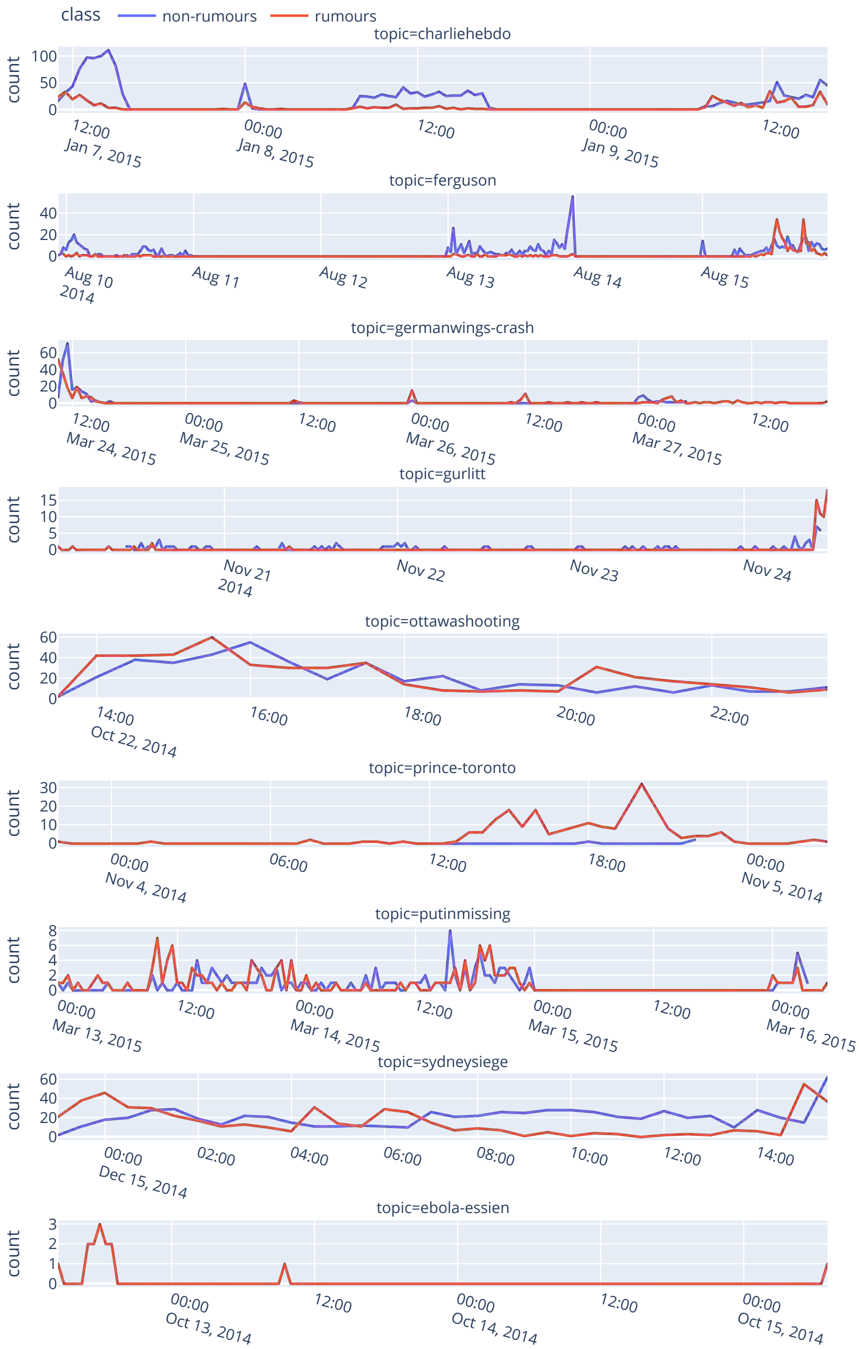


Fig. 4.5.: Topic tweet distribution sampled in 30-minute windows.

4.5 Results and Discussion

We conducted four main lines of experiments to see how quantification is affected when several shifts are inputted either to the training or test set. Throughout this section, we will try to answer the following questions:

1. **How does the training sample affect quantification performance?** Since documents obtained from the time-based sampling are already sampled by Twitter API, it is important to know how a biased training set would affect a real-case scenario.

2. **What is the typical behaviour of a quantifier when there is a *prior probability shift* in the test sample?** Once we have a trained quantifier, it is important to know how it will behave with samples whose class prevalence values differ from the ones seen in train.

3. **How does a change of topic (but not class) affect quantification performance?** This is a typical situation in SNS when a subject is discussed over time. Especially during events such as political debates in which several topics are discussed (e.g., economy, foreign policy, health system...). We artificially modify topic prevalence values both in train and test samples.

How does training drift affect quantification of a steady test sample?

We generated 3000-instance samples using a grid of 101 prevalence points with 5 repetitions and discarded the ones that had less than 2 instances in any of the two classes. We trained quantifiers in each sample and then we used it to estimate class prevalence values in a fixed test set. Figure 4.6 shows results in terms of absolute error.

CC presents a V-shaped error profile with the minimum error centred in 0.5, i.e., when the classes are totally balanced. PCC shows the same kind of profile, however the minimum error correspond to the point in which the training prevalence matches the test distribution. ACC, PACC and EMQ show U-shaped profiles, with different variability. Larger dispersion rates are present when the prevalence of any of the two classes is close to zero, which suggest that, even with the adjustments, the underlying classifier is not performing good enough to properly estimate prevalence values. ACC works best when the

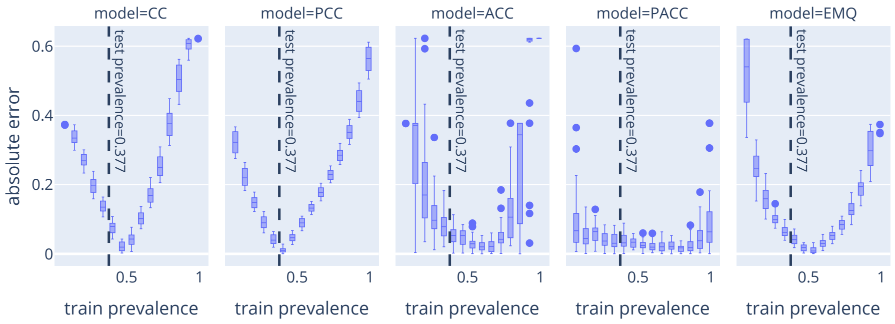


Fig. 4.6.: Absolute error when predicting test set class prevalence values when training with artificially-sampled training sets. PACC is the most stable method, as well as the best performing one.

training prevalence is almost balanced. In our tests, it seems like this range goes from $[p_{te}, 1 - p_{te}]$, where p_{te} is the true prevalence of the test sample. EMQ show superior predictions when both classes are balanced, despite that the error rates are generally lower than when using CC. PACC shows the best overall behaviour, with minimum error rates even with very imbalanced training samples. Moreover, it is the model with the lowest dispersion in the error distribution, and it manages to keep the absolute error below 0.1 in most cases.

How does test drift affect quantification?

This is the most typical experiment when dealing with quantification tasks. Our setup consist in a fixed training set and 100-instance samples for tests purposes. Test samples are artificially generated using a grid of 101 prevalence points with 5 repetitions. In this case, we do not discard any sample, since the quantifier should be able to accurately predict one-class samples.

Figure 4.7 shows absolute error distribution with respect to sample drift. Both CC and PCC are monotonically non-decreasing and, although PCC dispersion rates are lower, both of them are more or less constrained within the same error interval. Remaining quantifiers are non-monotonic. Absolute error for these quantifiers is constrained under 0.2 in most cases. After surpassing a 40% drift, they present a clear increase in the error but also in stability.

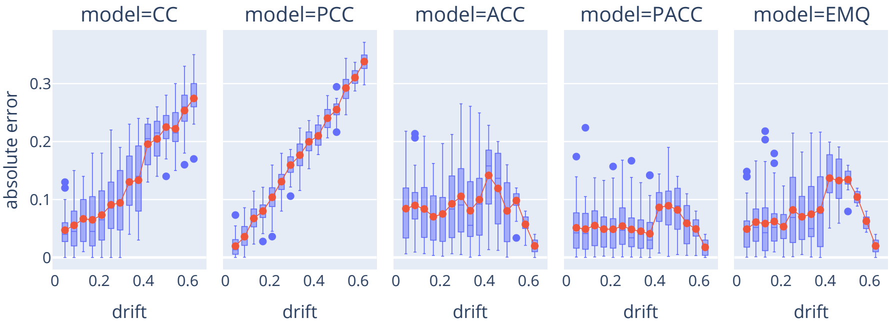


Fig. 4.7.: Absolute error when predicting artificially sampled test samples when training with the standard training set. Red line represents the mean value. Results show that PCC is the most stable method and PACC present the lowest overall error.

How does a change of topic affect quantification?

There is a typical situation in SNS that may affect classification and quantification performance, since it inputs covariate shift. When discussing a specific subject over time, documents may present different feature distribution or even distinct words and/or feature importance. This is particularly noticeable when retrieving content using event-related streams (e.g., political debates and natural disasters).

To test how this kind of shift will affect quantification performance, we generated artificial samples varying the topic prevalence (regardless of the class), using a grid of 9 prevalence points and 1 repeat. We drastically reduced the number of prevalence points since there are 9 topics, which means that maintaining the previous setup will yield $3 \cdot 10^{11}$ recombinations. We performed two kind of experiments:

1. 2500-instance samples used as training sets with varying topic prevalence values. Test set is static and the same for all the trained models. The purpose of this experiment is to check how a drift in training would affect production performance.

2. 100-instance samples used as test sets with varying topic prevalence values. A unique model is trained over the train set. The purpose of this experiment is to check how a change of topic (while maintaining the same target variable) scenario would affect quantification performance in a real-case scenario.

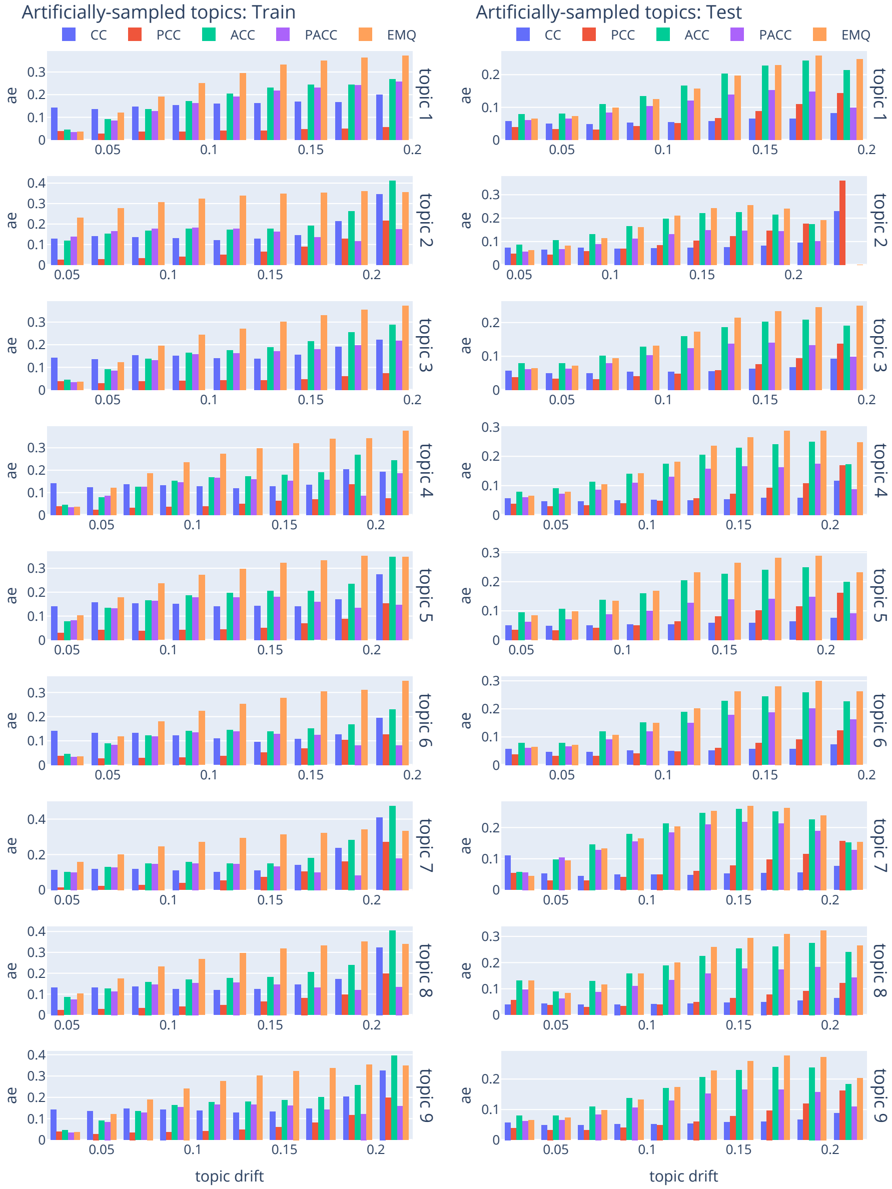


Fig. 4.8.: Absolute error when using artificially-generated samples with different topic prevalence values as train and test sets, respectively.

Figure 4.8 present results for both experiments in terms of absolute error. EMQ shows the worst performance in almost every case for both experiments, which is expectable. Since the Expectation-Maximisation mechanism is sensible to feature distribution, its performance would suffer when distributions are altered. PCC seems to have the best results when varying training topic prevalence, which suggests that the adjustment computed by ACC and PACC needs a representative training set to be able to correct the estimates without increasing the error rates.

When artificially varying the test samples, PACC is between the best performers in artificial tests. In some topics, PCC performance shows a natural detriment consistent with prior probability drift.

It is specially interesting to compare the cases of topic 6 and 7 between both experiments. ACC and particularly PACC show an improvement in terms of absolute error with higher drifts in train, while these show higher errors in test. Although further experiments are required, this results suggest that artificially balancing topics in train may result in better performance when estimating prevalence on variable test samples.

Figure 4.9 and 4.10 show absolute error in a nine-per-nine grid when varying topic prevalence values in training; as for the second experiment, Figure 4.11 and 4.12 show the same grid in testing. PCC shows the best overall performance when varying the training set while PACC perform the best with varying tests samples. All in all, our experiments confirm that *covarite drift* drastically affect quantification performance.

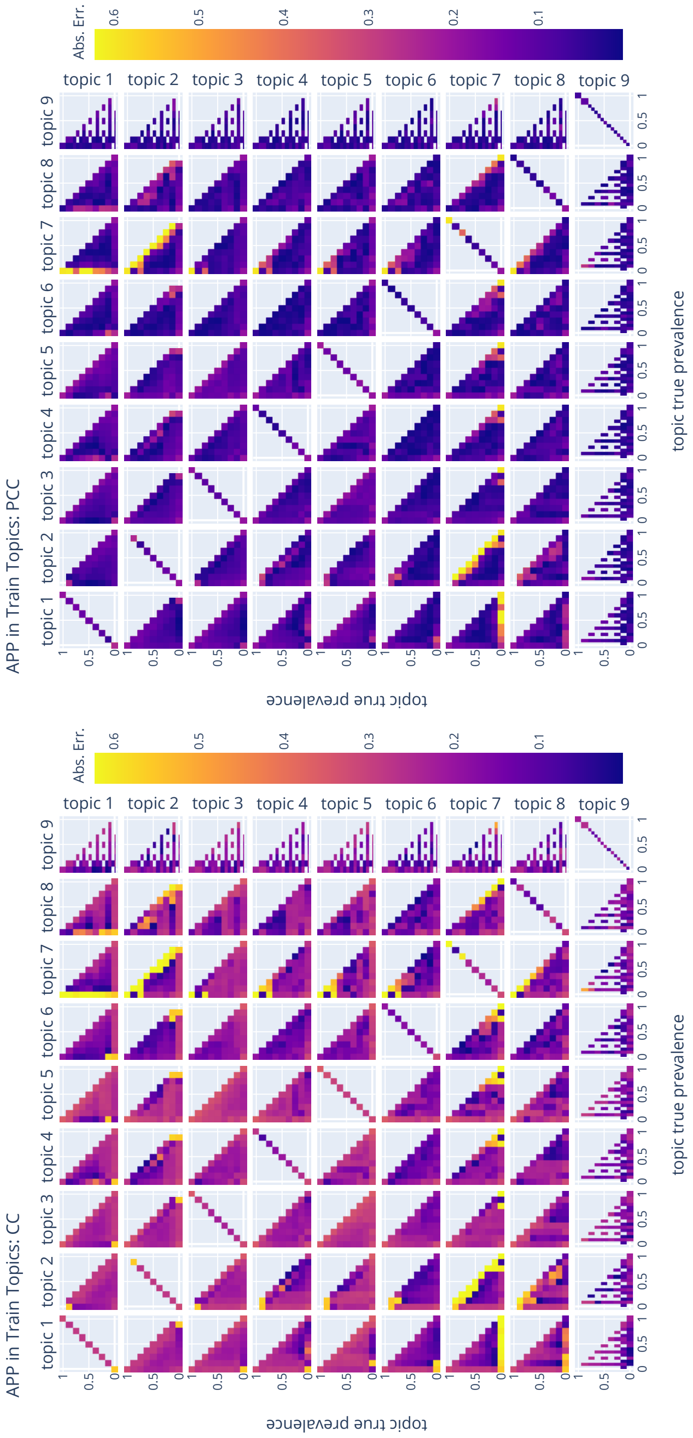


Fig. 4.9.: Absolute error when comparing topic vs. topic prevalence values artificially sampled and used as training sets (1/2).

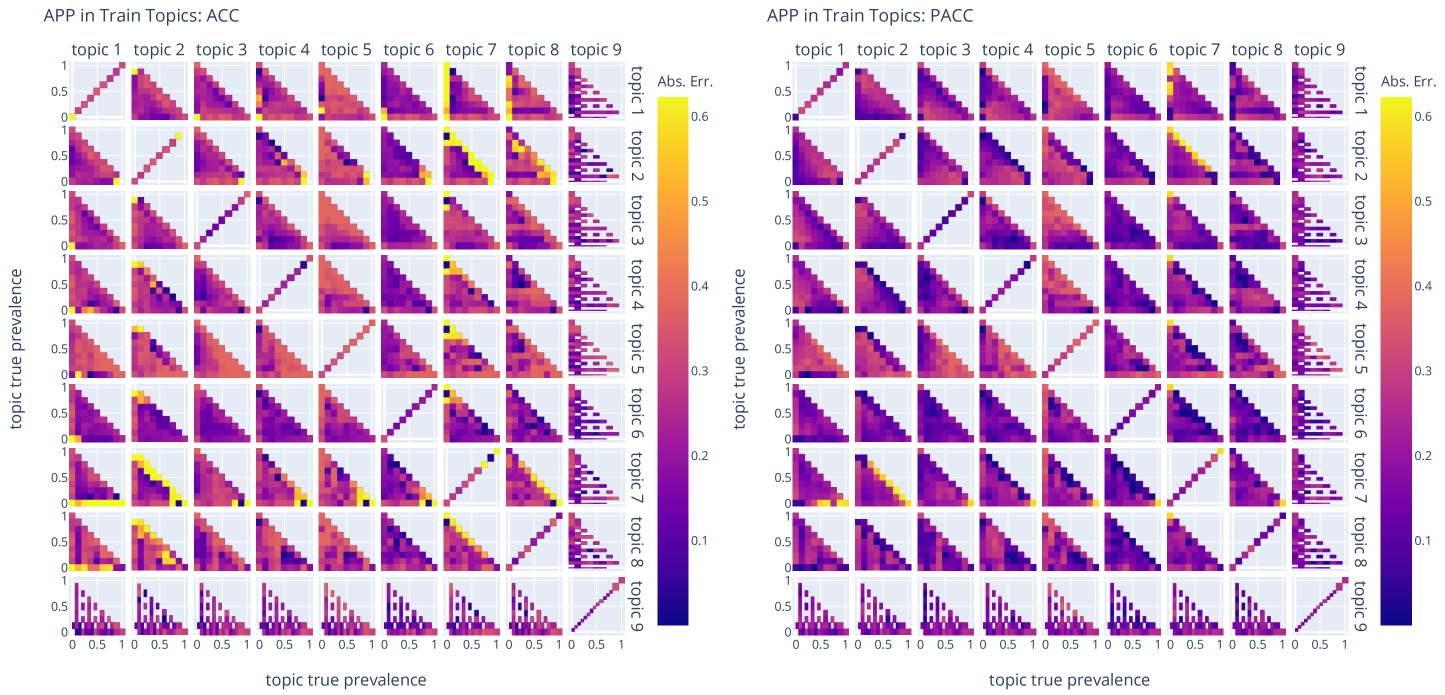


Fig. 4.10.: Absolute error when comparing topic vs. topic prevalence values artificially sampled and used as training sets (2/2).

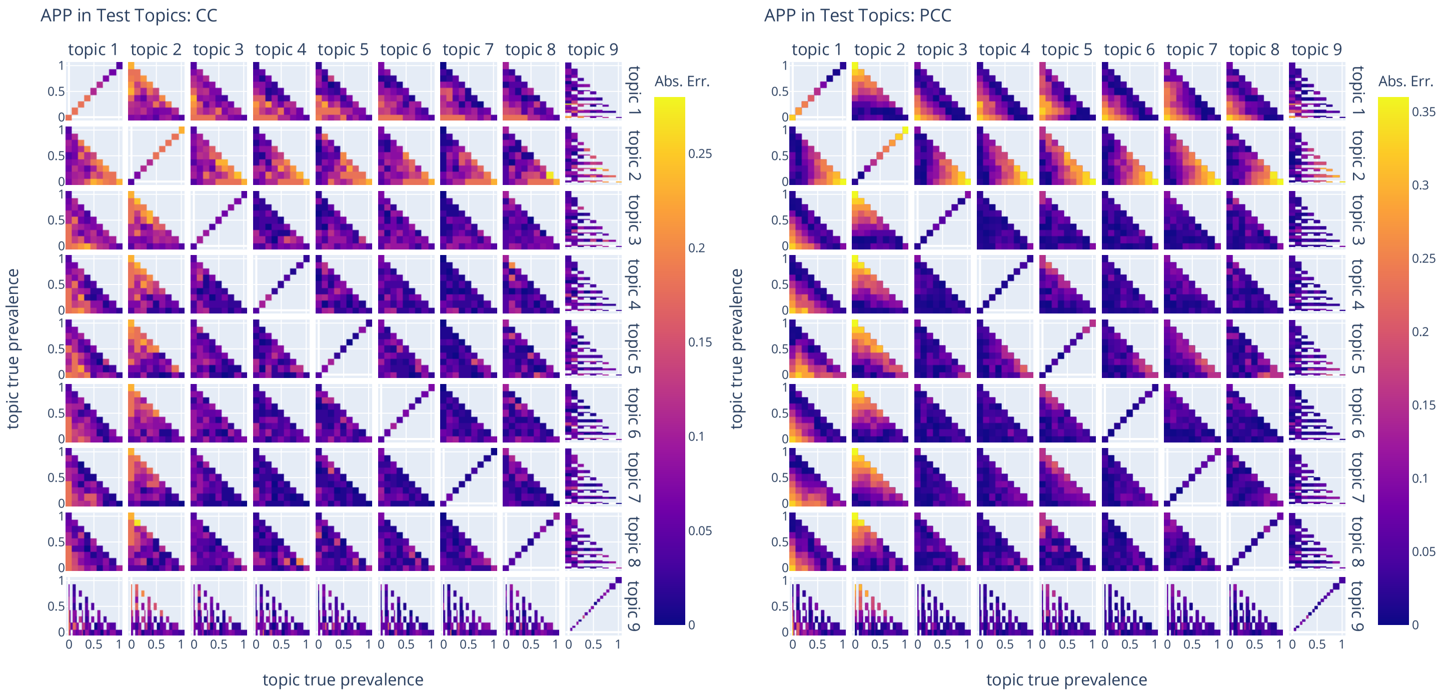


Fig. 4.11.: Absolute error when comparing topic vs. topic prevalence values artificially sampled and used as test sets (1/2).

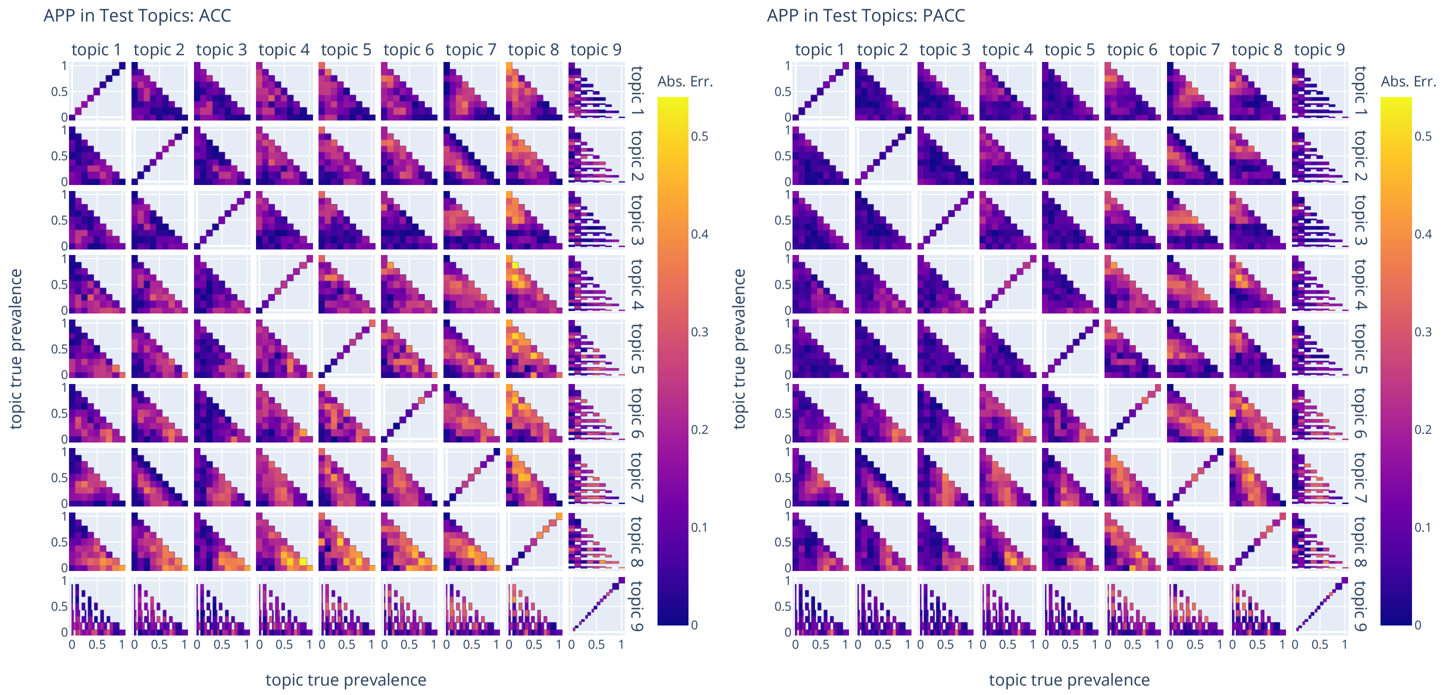


Fig. 4.12.: Absolute error when comparing topic vs. topic prevalence values artificially sampled and used as test sets (2/2).

4.6 Similarity-based Quantifiers

Covariate distribution drifts induce higher error rates when using well-known quantifiers. Multiple changes can be observed while attending to the different topics that are involved in the dataset. Figure 4.13 shows feature importance with respect to the topic. *Charlie Hebdo*, *Ferguson*, *German Wings Crash*, *Sydney Siege* and *Ottawa Shooting* have similar importance distribution, although with some differences that are more noticeable in the case of *German Wings*. Furthermore, *Gurlitt*, *Prince Toronto* and *Putin Missing* are the most different in terms of importance.

Figure 4.14 show word clouds for every topic. *http* makes reference to links, which seem to be a good predictor to distinguish between rumours and non-rumours for the *German Wings Crash* topic. Different topic involve different words, which intuitively suggest that a representative sample of all topics need to be present in the training set for the classifier to learn upon.

Attending to their features, it is possible to study how different topic instances distribute over the input space. Figure 4.15 shows a grid of 9-per-9 pairwise feature distribution with respect to the class and topic. There are several cases in which topics are linearly separable (e.g., feature 4 for *prince-toronto* and 3 for *ottawashooting*).

We thought that we could obtain better quantification performance using either (1) similarity measures to adjust posterior probabilities attending to its neighbours or (2) clustering algorithms to train several quantifiers that specialise in different inputs. We applied k-means clustering algorithm with $k = 9$ to check if unsupervised algorithms are capable of detecting each topic. Figure 4.16 show detected clusters and Table 4.2 shows different metrics to compare calculated cluster with ground-truth topics. At first sight, it seems like calculated clusters match the original topics. However, similarity measures are valued close to the mid-range point, which suggest that our clusters are noisy and not a perfect match with respect to the original topics.

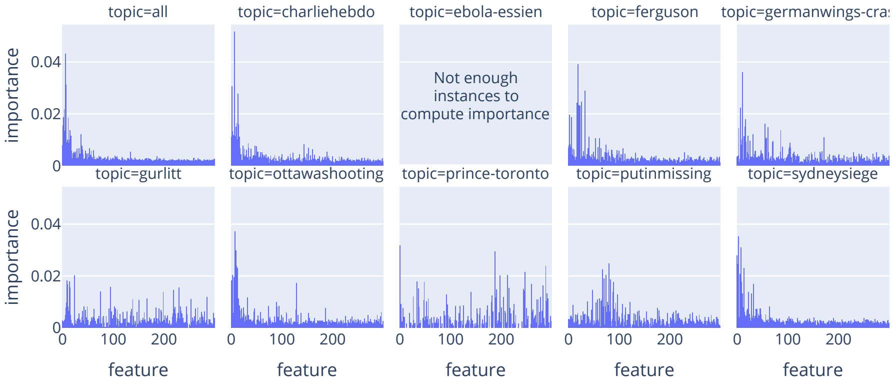


Fig. 4.13.: Per-topic impurity-based feature importance when using 300-dimensional truncated SVD features to predict tweet class.

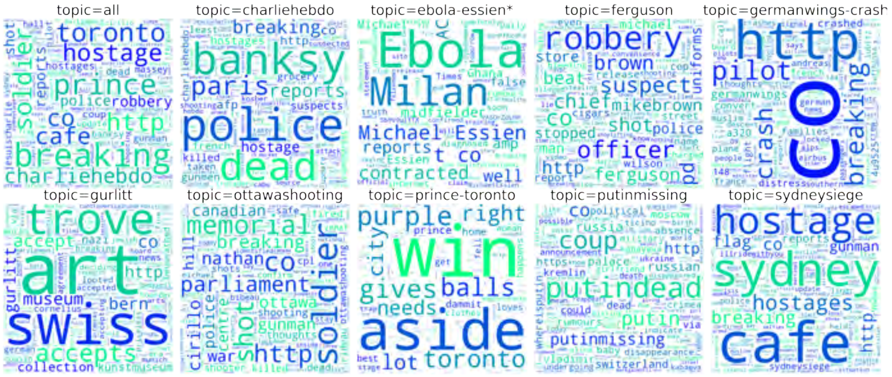


Fig. 4.14.: Per-topic TF-IDF feature importance when predicting document class. *Ebola-essien topic show word frequency instead of feature importance due to low number of instances.

Tab. 4.2.: Comparative measures between clusters and ground-truth topics.

metric	score
Adj. Rand Index	0.3859
Adj. MI	0.5486
Norm. MI	0.5550
Homogeneity	0.5601
Completeness	0.5500
V-measure	0.5550
Fowlkes-Mallows	0.5091

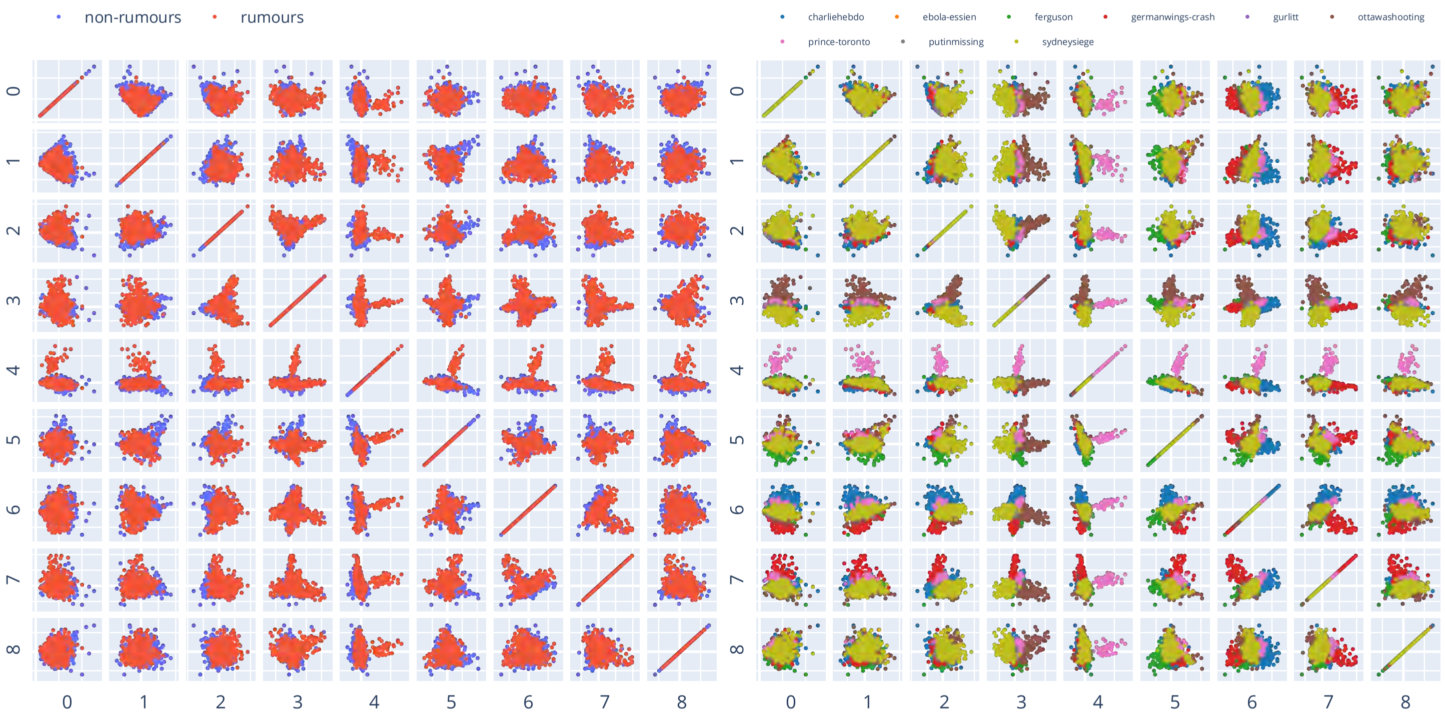


Fig. 4.15.: 9-feature pairwise projection of classes and documents. There are topics that can be easily differentiated.

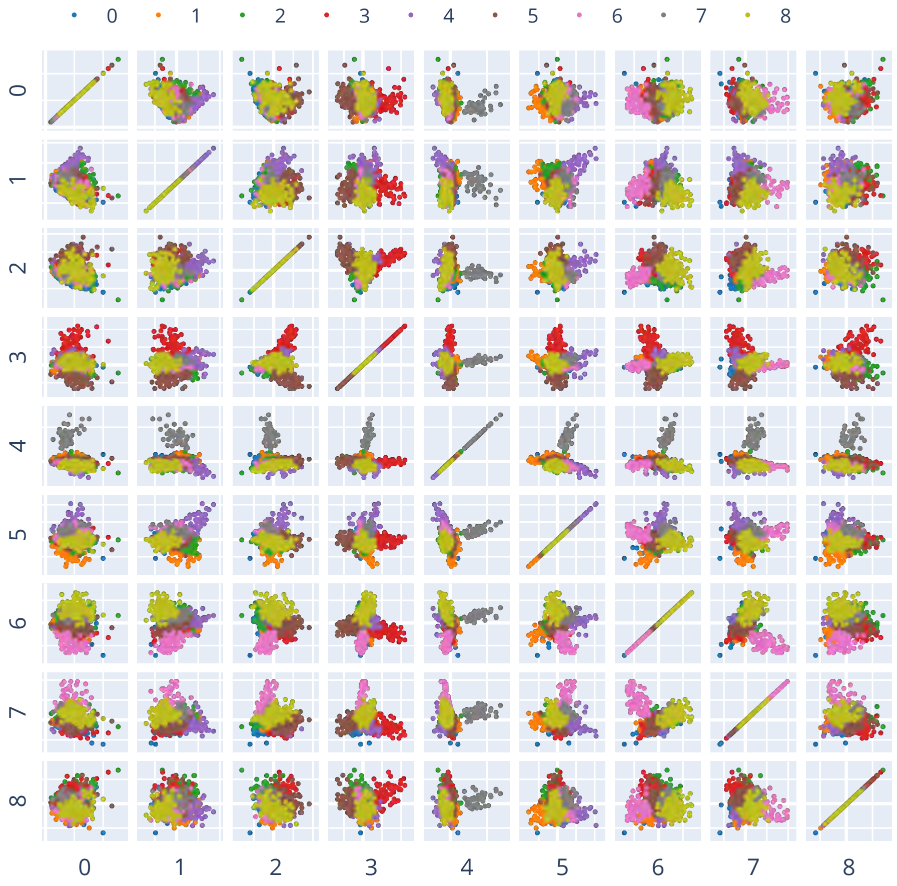


Fig. 4.16.: K-Means clustering with k=9. At first sight, there are clusters that can be easily differentiated and matched with ground-truth topics.

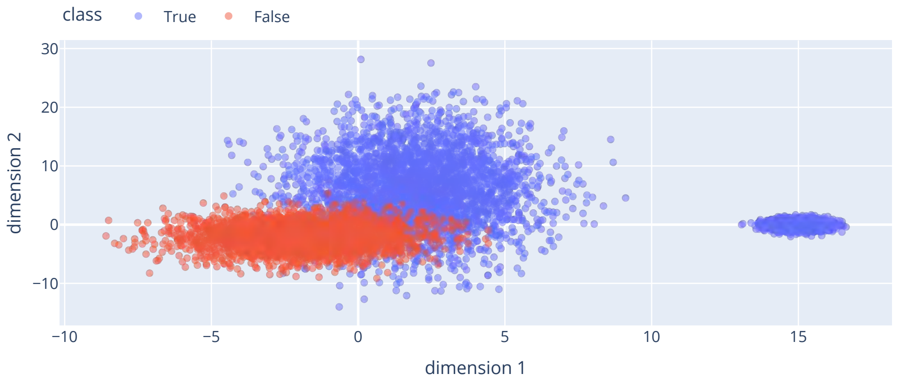


Fig. 4.17.: Artificial dataset generated from 3 data blobs

In order to study if similarity-based quantifiers are valid solutions, we generated an artificial dataset that is composed of 3 data blobs (two overlapping ones, that are from opposite classes, and another isolated one that belongs to the positive class). Figure 4.17 shows how blobs are distributed over the input space.

Distance-based similarities

Given a training sample τ and a test sample σ, such that $D = \tau \cup \sigma$, ACC adjustment is calculated as follows:

$$\hat{p}(y_i) = \frac{\hat{p}_0(y_i) - fpr}{tpr - fpr} \tag{4.14}$$

which is equivalent to:

$$\hat{p}(y_i) = \hat{p}_0(y_i)\left(\frac{1}{tpr - fpr}\right) - \frac{fpr}{tpr - fpr} = a\hat{p}_0(y_i) + b \tag{4.15}$$

being a and b two terms that define a linear combination. Since the uncorrected prevalence value $\hat{p}_0(y_i) = \frac{|\{x \in \sigma : h(x) = +1\}|}{|\sigma|}$,

$$a\hat{p}_0(y_i) + b = b + \frac{a}{|\sigma|}\sum_{x \in \sigma} h(x) = \sum_{x \in \sigma} \frac{ah(x) + b}{|\sigma|} \tag{4.16}$$

therefore the linear combination can be applied directly to classifier predictions. Analogously for the case of a *soft classifier* s and a PACC adjustment, the linear combination can be applied to classifier posterior probabilities without changing the expected result. Notice that, in such case,

$$\hat{p}_0(y_i) = \frac{1}{|\sigma|}\sum_{x \in \sigma} s(x)_i \tag{4.17}$$

From here, we tried to individually influence the adjustment for each data point such that it takes into account the similarities towards known points. Intuitively, this would work as if *true* and *false positive rates* were computed locally, within a similarity radius.

Let sim be a similarity measure and λ a similarity threshold. For each data point $x \in D$, it is possible to obtain a similarity neighbourhood of x such that

$$neigh_\lambda(x) = \{t \in \tau : sim(t, x) \geq \lambda\} \tag{4.18}$$

However, we can only compute misclassification rates with known data points, therefore we applied a cross-fold validation scheme to compute mean tpr and fpr for each point in training. Let τ and τ' be training and validation folds.

$$\forall t \in \tau', tpr(t) = \sum_{x \in D^+ \cap neigh_\lambda(t)} \frac{s(x)_1}{|D^+ \cap neigh_\lambda(t)|} \tag{4.19}$$

$$\forall t \in \tau', fpr(t) = \sum_{x \in D^- \cap neigh_\lambda(t)} \frac{s(x)_1}{|D^- \cap neigh_\lambda(t)|} \tag{4.20}$$

Therefore every training point will have local tpr and fpr attributes. When used to predict, each adjustment would be calculated as the weighted mean of their neighbours, such that:

$$L_{tpr}(x) = \frac{\sum_{t \in neigh_\lambda(x)} sim(x, t) * tpr(t)}{\sum_{t \in neigh_\lambda(x)} sim(x, t)} \tag{4.21}$$

$$L_{fpr}(x) = \frac{\sum_{t \in neigh_\lambda(x)} sim(x, t) * fpr(t)}{\sum_{t \in neigh_\lambda(x)} sim(x, t)} \tag{4.22}$$

$$\forall x \in \sigma, s'(x) = \frac{s(x) - L_{fpr}(x)}{L_{tpr}(x) - L_{fpr}(x)} \tag{4.23}$$

and, ultimately,

$$\hat{p}(y_i) = \frac{1}{|\sigma|} \sum_{x \in \sigma} s'(x)_i \tag{4.24}$$

Tab. 4.3.: Behaviour of the denominator as L_{tpr} and L_{fpr} increase or decrease.

	$L_{fpr} \downarrow$	$L_{fpr} \uparrow$
$L_{tpr} \downarrow$	den↓	den−
$L_{tpr} \uparrow$	den+	den↓

If $\lambda = 0$, values of L_{tpr} and L_{fpr} are forced to 1 and 0 respectively, so there is no adjustment and it behaves like PCC. If $\lambda = 1$, it is straightforward to prove that $D^{\{+,-\}} \cap neigh_1(t) = D^{\{+,-\}}$. Consequently, $L_{tpr} = tpr$ and $L_{fpr} = fpr$, which implies that the adjustment is equivalent to PACC, since:

$$\hat{p}(y_i) = \frac{1}{|\sigma|} \sum_{x \in \sigma} \frac{s(x)_i - fpr}{tpr - fpr} = \frac{\frac{1}{|\sigma|} \sum_{x \in \sigma} s(x)_i - fpr}{tpr - fpr} = \frac{\hat{p}_0(y_i) - fpr}{tpr - fpr} \tag{4.25}$$

We used cosine similarity and variable λ settings to study the behaviour of our proposal. Unfortunately, initial results were not promising and led us to discard this approach.

Let us start by analysing equation 4.23. L_{tpr} and L_{fpr} are local extrapolations of the misclassification rates that happen to occur in a λ radius centred in a test point. It does not have either the same interpretation nor the same behaviour than actual misclassification rates, but it is easy to understand how they influence the posterior probabilities of such test point.

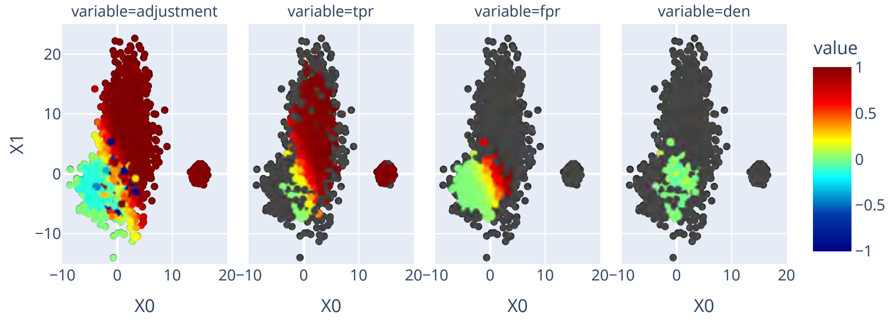


Fig. 4.18.: Local adjustment behaviour when using euclidean distance with radius 0.5 (only for illustrative purposes).

Table 4.3 ▦ shows the typical behaviour of the denominator of equation 4.23 when varying local adjustments. The expected behaviour happens when the local true positive rate is larger than the false positive rate. In such case, the

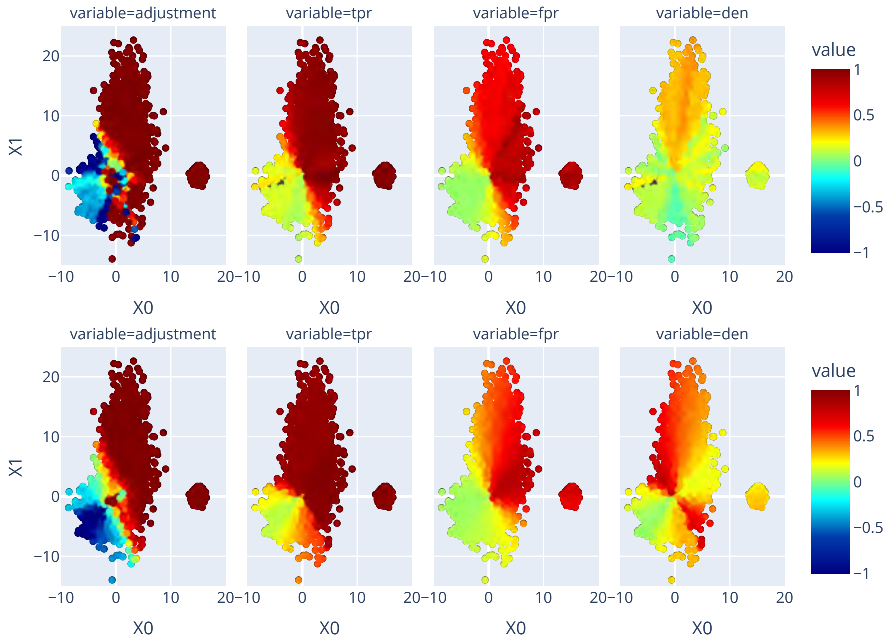


Fig. 4.19.: Local adjustment behaviour when using cosine distance (radius 0.02 and 0.5, respectively).

denominator would be greater than zero and the linear combination would adjust posterior probabilities to our needs. However, when $L_{tpr} \simeq L_{fpr}$, denominator would be too small and $s'(x)$ would be distorted.

Figure 4.18 🖼 was made for illustrative purposes. It uses euclidean distance as a similarity measure (data points in a 0.5 radius). Denominator is only defined when L_{tpr} and L_{fpr} are both defined. However, in such cases in which their values are close, denominator is close to zero and therefore the adjustment get distorted. It is possible to see data points with extreme values that would affect the prevalence estimation (see equation 4.24).

Figure 4.19 🖼 show the same setup however using cosine similarity. Results are not as dramatic yet they present the same behaviour. When local misclassification rates are close to each other, or when the L_{fpr} is greater than L_{tpr}, adjustment is extreme and induces a significant error while counting (prevalence value).

Using 0.5 radius for cosine similarity yields an adjustment that, while over optimistic, is coherent with data distribution. Unfortunately, optimal λ values highly depend on distribution attributes, such as dispersion rates, density and cardinality. Therefore, it needs to be tuned for each part of the decision boundary. Apart from intractable, λ values when performing model selection tend to be as larger as possible, since this is the overall optimal solution (and the definition of PACC).

In our numerous experiments, we could not find a setup in which similarity-based quantifiers improved the performance of traditional quantifiers.

4.7 Political Debates: a Case Study

We used what we learned from the experiments on the PHEME dataset and we applied it to one of our political debates dataset. We aim to study sentiment prevalence estimation in a dataset of tweets related to an event (tweet time-series). In order to do so, we train our quantifier over a subset of tweets uniformly sampled in time. Then, we would use time windows with different resolutions to obtain tests samples of tweets over time. This approach serves two purposes:

- The quantifier would be aware of all the topics and feature distributions. In section 4.5, we concluded that *covariate shifts* dramatically affects the quantification performance, therefore it is necessary to provide a wide scope of tweets to train the quantifier.
- Quantification error could be measure at different points of the time-series to check whether it is homogeneous or not.

We used a pre-trained BERT model to obtain the document sentiment values for all the tweets in our dataset. The quantifier would hence qualify as a surrogate model, which is not optimal for production but still useful in a test setup like ours.

We applied simple preprocessing mechanisms to tokenise the tweets, and to remove URLs and stop words. Tweet aggregation was performed using 30-second windows. Using a five-fold cross-validation scheme, we measured quantification performance on a per-window basis.

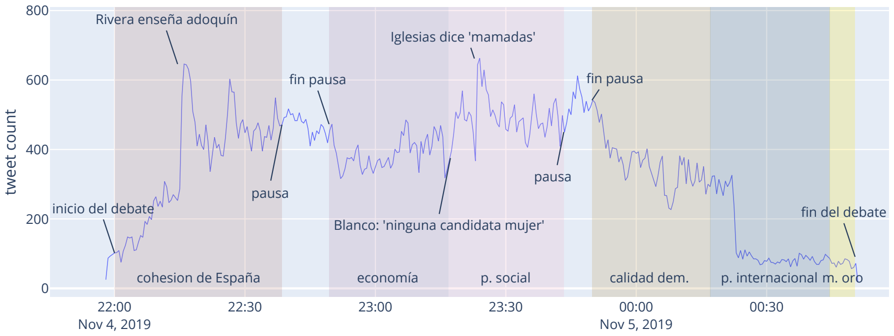


Fig. 4.20.: Debate tweet count

There are five thematic blocks during the debate:

- *Cohesión de España*, regarding Spanish domestic politics and mainly focused on Catalonia's independency. It lasted approximately 38 minutes.
- *Economía*, related to economical strategies. It lasted 27 minutes.
- *Política Social*, that was about social politics. It lasted another 27 minutes.
- *Calidad Democrática*, in which the candidates discussed how to maintain a trustworthy democracy. It lasted 28 minutes.
- *Política internacional*, regarding Spanish international politics. It lasted 27 minutes.

Figure 4.20 shows tweet count during the debate. At first, it is noticeable that the number of published tweets was gradually increasing until one of the candidates applied a controversial strategy which radically rose the number of tweets (14 minutes after the start of the debate). Another spike in popularity can be observed very close to this first one, in which the same candidate showed a multiple-page list to attack a political opponent. From here until the end of the first break, tweet count oscillated around 450 tweets per 30-second window. There was a little recession during the second and third parts of the debate, until another two consecutive events rised the tweet count. The first one was induced by one of the moderators, which accurately pointed out that there were five male candidates discussing gender equality; a few minutes later, one of the candidates mispronounced a word that resulted in another one with sexual connotations. From these events until a little before midnight, tweet count oscillated around 500 tweets per 30-second window.

Lastly, the number of tweets started to decrease. 20 minutes past midnight, people complained about the late schedule of the debate and the popularity of the event decreased to its minimum.

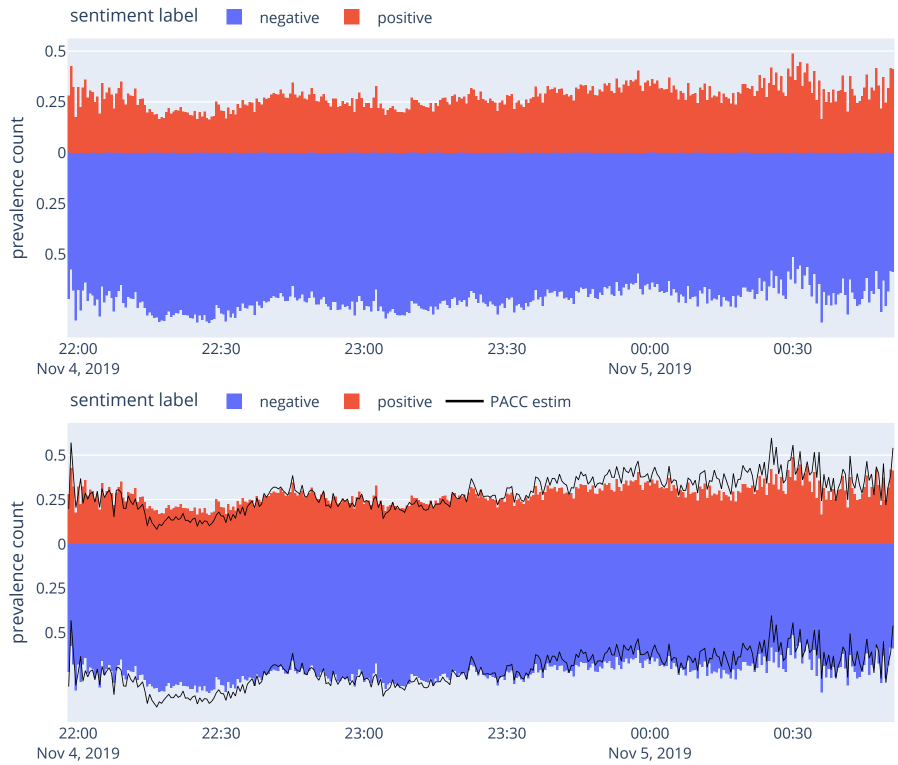


Fig. 4.21.: Sentiment prevalence count and PACC prevalence estimation

After applying pre-trained BERT sentiment analysis, we noticed tweets tend to be rather negative. Figure 4.21 🖼 shows prevalence count of positive and negative classes in 30-second windows. We also included PACC estimates as a reference. Error-prone windows are close to the events described in the paragraph above. This is actually expectable since those events are extraordinary situations that resulted in *memes* and a lot of jokes.

From the application of well-known quantifiers (see figure 4.22 🖼), we discovered that PCC yields best results in terms of absolute error. CC approach, which is the most popular one [46], is the worst performing quantifier. EMQ shows good performance except in those segments related to the first popularity spike, 15 minutes after the debate. Since EMQ is based on expectation-maximisation, its behaviour is expectable for aforementioned reasons (features deviate from

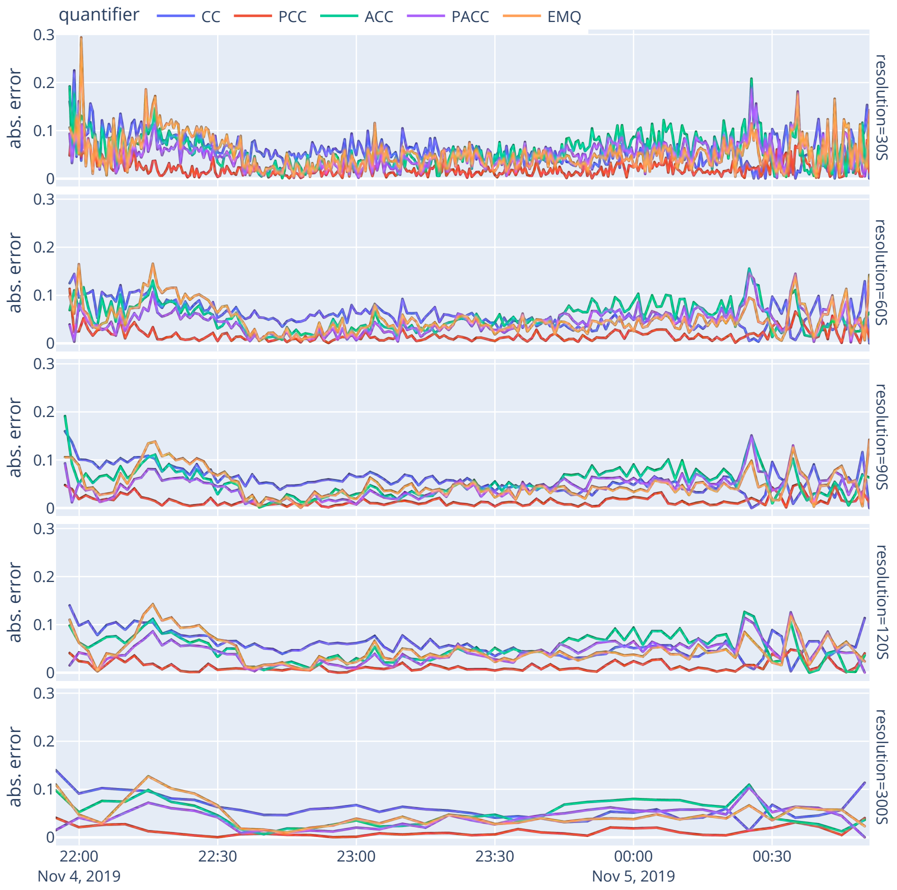


Fig. 4.22.: Absolute error when quantifying over windows of different resolutions.

the original distributions, since they are kind of unrelated to politics). Adjusted classify and count methods (ACC and PACC) show better performance than CC, however they are not able to improve PCC results.

4.7.1 Probabilistic Time-Adjusted Classify and Count

As we studied in section 4.5, covariate drift affects quantification performance. In light that the adjusted aggregation is less precise than a straightforward probabilistic count, we tried to predict the best adjustment for each window.

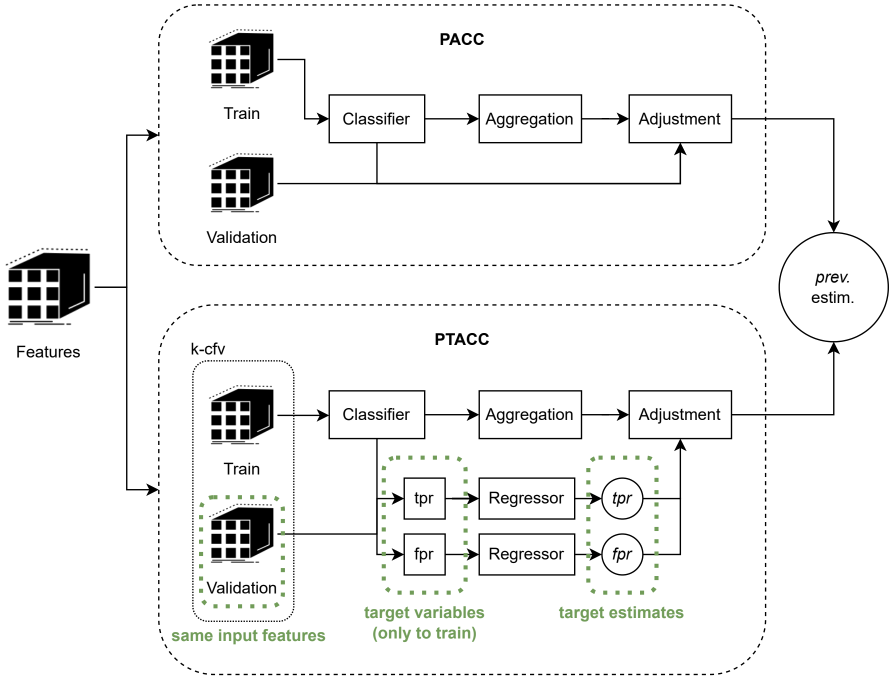


Fig. 4.23.: PTACC model architecture.

The hypothesis is that each window would present a particular drift against a full-range training set. Therefore, if we could predict both terms of the adjustment (*tpr* and *fpr*) for each window, we would be able to make better prevalence estimations. Analogously to section 4.6, our assumption is that we can model the misclassification rates to make a *local adjustment* during the aggregation phase of the quantifier.

To that end, we propose a model in which two regressors are trained over the feature matrix to predict a target variable (see figure 4.23), which are the true and false positive rates of the classifier for each window (i.e., different covariates distributions). We will refer to this strategy as *Probabilistic Time-Adjusted Classify and Count (PTACC)*.

However, PTACC introduces two new predictions ($t\hat{p}r$ and $f\hat{p}r$), and hence additional noise. The model would not only suffer from the errors of the classifier but also from the errors when trying to predict the necessary components for the prevalence count adjustment (see figure 4.24).

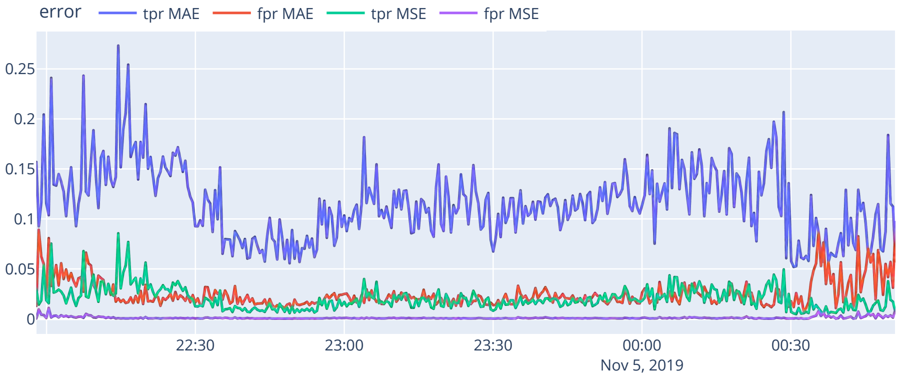


Fig. 4.24.: tpr and fpr prediction errors through time.

Especially in the case of *tpr* predictions, absolute error is high. Consequently, per-window adjustment result are not sufficient to improve PCC results. PTACC improves PACC results a little, however it does not make out for the additional overhead of the model.

Figure 4.25 show diagonal plots of the tested quantifiers. These are very explanatory figures in which true prevalence values are represented in the x-axis against predicted prevalence values (in the y-axis). The perfect quantifier would be the line resulting from the function $y = f(x) = x$, a line of unitary slope without any kind of offset. The closest the quantifier is to the diagonal, the better.

PCC shows almost perfect behaviour under samples with prevalence values below 0.4. It improves the estimations of CC, that is the most common aggregation approach in literature. On the contrary, ACC, PACC, EMQ, and our proposal (PTACC) present higher error rates and deviation from the perfect quantifier (see figure 4.26).

Comparing between folds, PTACC shows slightly less dispersion than PACC, however PCC the quantification method with the lowest standard deviation, marginally lower than CC (the naïve approach). Coherently with what we have seen in literature, classic *classify and count* can be improved just by doing a probabilistic count instead of a crisp one.

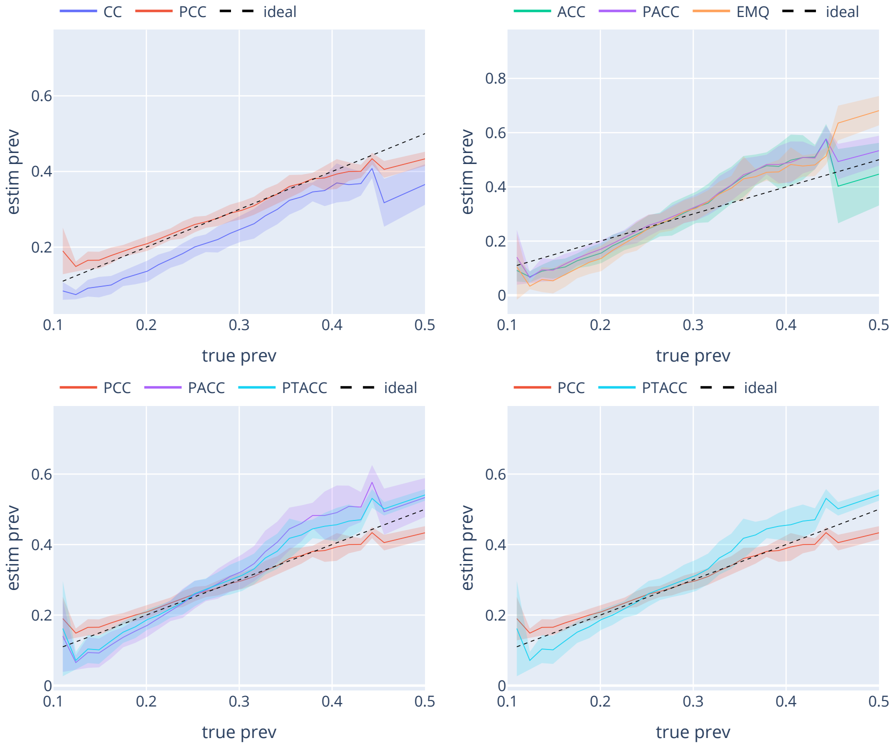


Fig. 4.25.: Diagonal plots of true prevalence against predicted prevalence for different quantifiers.

Analogously to section 4.6, local adjustment of posterior probabilities before aggregation does not improve existing methods. This conclusion suggests that, although temporal and spatial similarities can arguably be used to improve the adjustment, the most straightforward approaches are not sufficient and further research is required.

4.8 Conclusions

Real-time Twitter monitoring is a popular application of the social network's data. Although some of these application deal with individual data, the most popular task in the social network is sentiment estimation. When estimating class prevalence counts (in the case of sentiment, positive and negative classes), there are other approaches to the traditional *classify and count*.

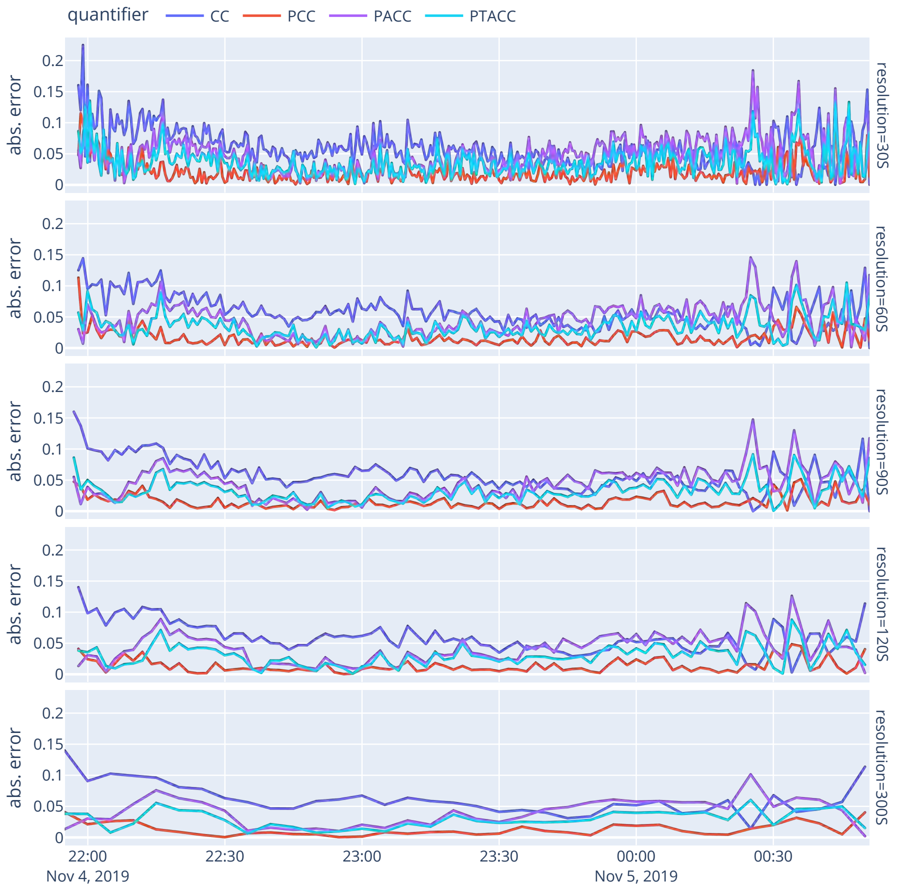


Fig. 4.26.: PTACC absolute error when quantifying over windows of different resolutions.

Previous work established that quantification methods are useful for such scenarios, and they propose a change of perspective to stop dealing with these problems as if they were classification tasks. Their proposal is not only valid for static SNS databases but also for real-time retrieval and time-based analysis.

After analysing the different type of bias that may be present in SNS, we explored current state-of-the-art in quantification, error measures and evaluation protocols. We measure the influence of *prior probability shift* and *covariate shift* both in training and test samples, and we determined that *adjusted* methods (like ACC and PACC) are the best option when dealing with *prior probability shift*, while PCC is the best performing quantifier for *feature distribution drifts*.

We tried our own proposal for prevalence count adjustment from a similarity-based perspective. Our hypothesis was that a fine-tuned adjustment based on local mislassification will improve the prevalence estimation. Unfortunately, our results did not improve the baselines. This open a new research avenue, since other methods and techniques (i.e., label propagation literature) may be applied to improve results.

Taking into account our findings, we studied the performance of quantifiers when predicting over time-based samples in one of our datasets. Since both *prior probability shift* and *covariate shift* are present in the samples (discussed topics vary over time), PCC yielded the best overall performance. Analogously to similarity-based quantifiers, we tried to adjust prevalence estimations with time-based predictions of misclassification rates. However, results did not improve the baselines. We think that both spatial- and time-based adjustments can be addressed with similar approaches and we expect to continue with this line of work in the future.

5 Exploratory Data Analysis of 2019 Spanish General Elections Dataset

During the development of this dissertation, we built several supervised datasets mostly related to politics. However, one of them was also annotated following a taxonomy of emotions [234]. In this chapter, we perform an exploratory analysis this dataset and a we explore how these features vary among parties and political wings.

The 2019 Spanish National Elections dataset is composed of more than 120k tweets that were retrieve using the hashtag #Debatea5RTVE during a main political event on November 2019. Tweets belong to more than 50k users. It was annotated in terms of sentiment towards each presidential candidate (5 labels) and also with a full taxonomy of emotions (62 labels) [234].

5.1 Feature Analysis

Political parties

Whether the tweet is positive or negative towards each political party. There are 5 possibilities, that are:

- Cs, as in Ciudadanos.
- PP, as in Partido Popular.
- PSOE, as in Partido Socialista Obrero Español.
- VOX.

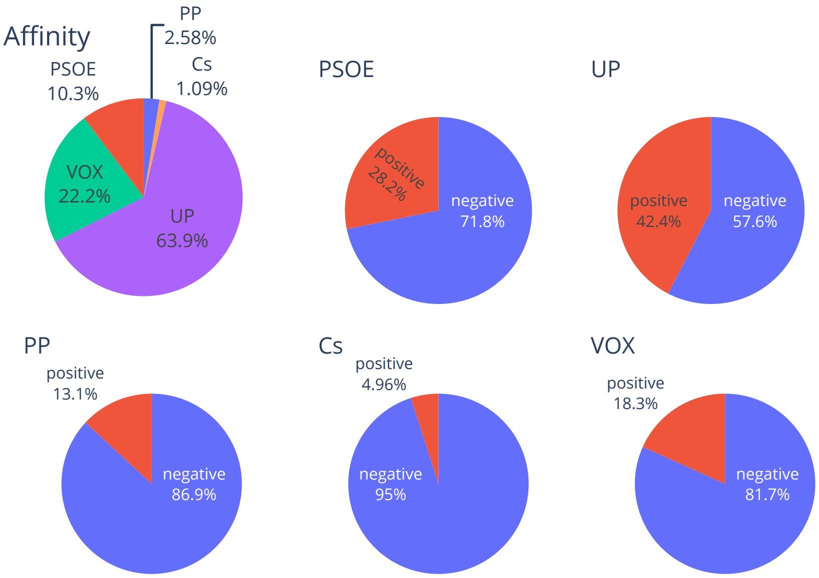


Fig. 5.1.: Affinity and sentiment count stats for the annotated tweets in the dataset.

- UP, as in Unidas Podemos.

A document is annotated as positive or negative when there is evidence for or against a political party, a candidate or any other party member. These labels may be propagated using the property expansion algorithm (please refer to section 2.4).

Figure 5.1 shows sentiment prevalence towards each political party, as well as overall tweet affinity. UP and PSOE are the parties with bigger counts of positive tweets; on the contrary, Cs and PP present the lower positive counts.

In terms of overall sentiment, UP and VOX represent roughly the 86% percent of positive tweets. Both parties are considered *populist* [235, 236], hence it may be related to its popularity in social media [237].

Binary gender

To annotate gender, we took into account the name and the description of the user, but we advised to disregard profile pictures.

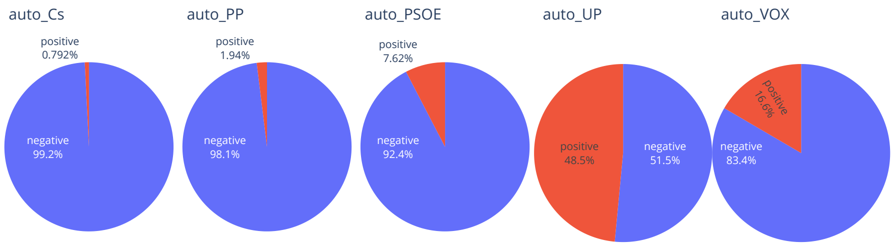


Fig. 5.2.: Sentiment count stats for the expanded annotations in the dataset.

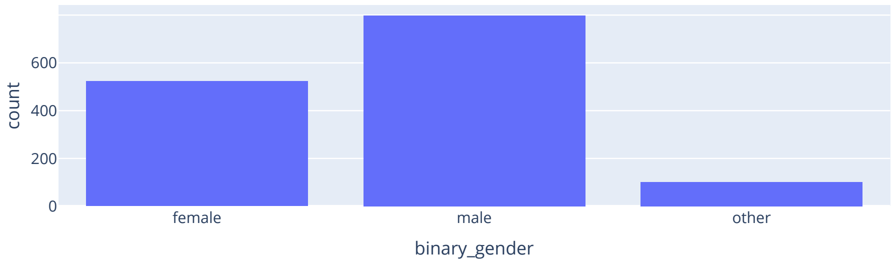


Fig. 5.3.: Histogram of *gender*

The only political party with higher prevalence of female users is PSOE. UP has the highest absolute count of female users, however data shows that they have twice as males. This also happens for VOX.

Age

User's age group. It is only annotated for those users who show strict evidence in their profile (such as birth date or a mention in their description).

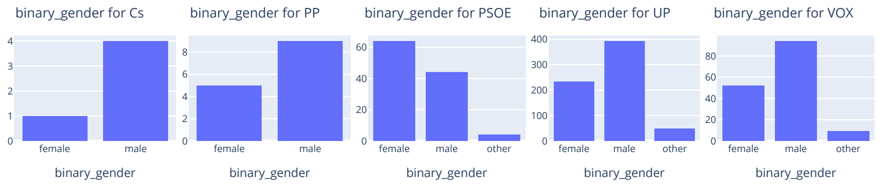


Fig. 5.4.: Per-party gender stats.

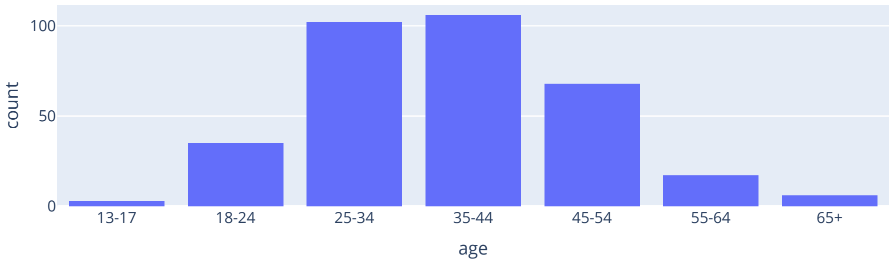


Fig. 5.5.: Histogram of *age*

SNS are known to have a higher prevalence of some age groups, depending on the platform and its target. Most of Twitter users are between 25 and 34 years old [238], however the politics topic show a higher prevalence of users between 35 and 44 years old (see figure 5.5 🖼).

In terms of age by political party, samples for PP and Cs are not representative. PSOE is linked to users in the 35-44 age group, as well as VOX. UP most popular age group is 25-34, followed by 35-44 and 45-54 years old (see figure 5.6 🖼).

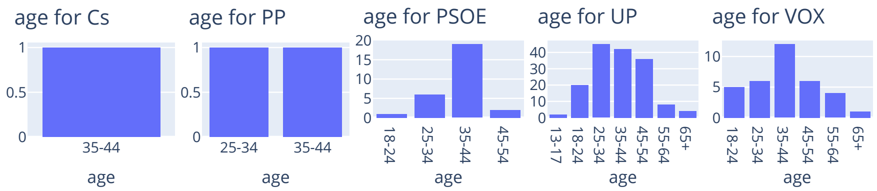


Fig. 5.6.: Per-party age stats.

Sentence type

There are four types of sentences:

- Declarative, which are used to make statements.
- Exclamative, used to express strong emotion.
- Imperative, used to order the addressee to do (or not) something.
- Interrogative, that are used to get information from the addressee.

Most common type is *declarative*, and there are not significant differences between political parties.

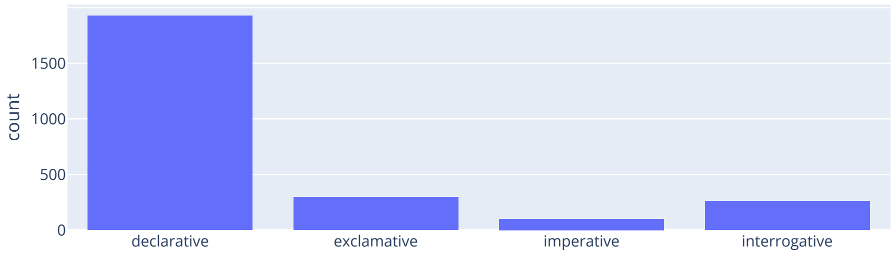


Fig. 5.7.: Histogram of *sentence type*

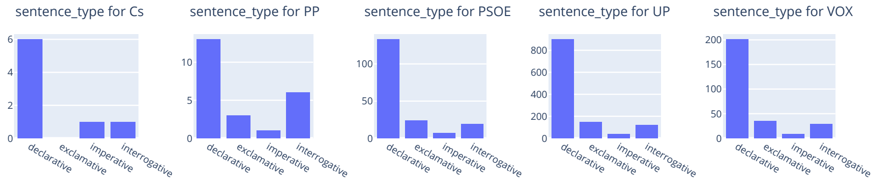


Fig. 5.8.: Per-party sentence type stats.

Speech act

Types of speech act are:

- Representative sentences, such as assertions, statements or claims.
- Commissive sentences, such as promises, oatchs or threats.
- Directive sentences, like commands, requests or invitations.
- Declaration sentences, like blessings, arrests and judicial speechs acts.
- Expressive sentences, that make assessments of attitudes, like greetings or apologies.
- Verdictive sentences, that are used to make judgement about the acts of a third person.

In line with the *sentence type*, the most common case is *representative sentences*. Once again, there are no significant differences between parties.

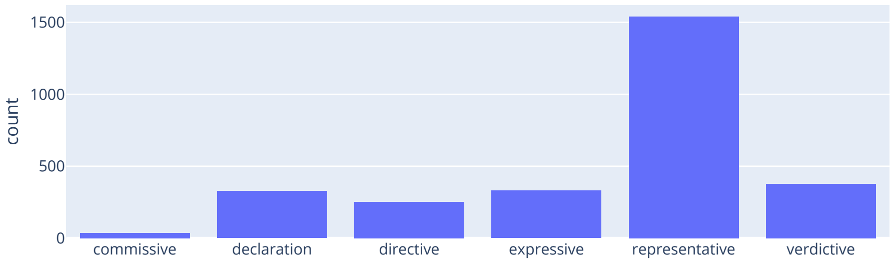


Fig. 5.9.: Histogram of *speech act*.

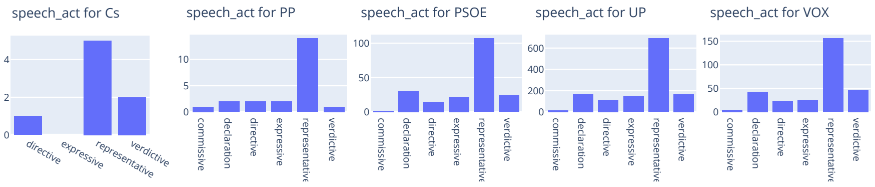


Fig. 5.10.: Per-party speech act stats.

Pragmatic function

Most difficult scenario in terms of pragmatic function is to detect irony. Written extreme statements can be easily confused with ironic statements, and sometimes it is impossible to distinguish between them without context.

- Literal, when the author wants to manifest exactly what is written.
- Metaphorical, that are symbol of something else.
- Ironic, when the author wants to manifest the opposite of what is written.
- Sarcastic, a sharp form of irony meant to ridicule someone.
- Rhetorical question, that are not meant to be taken literally and expect no answer.
- Hyperbole, an exaggerated figure of speech.

Although users with affinity towards PP show a higher prevalence of metaphorical sentences with respect to ironic ones (which is not the common case) the sample is not representative enough to assert conclusions.

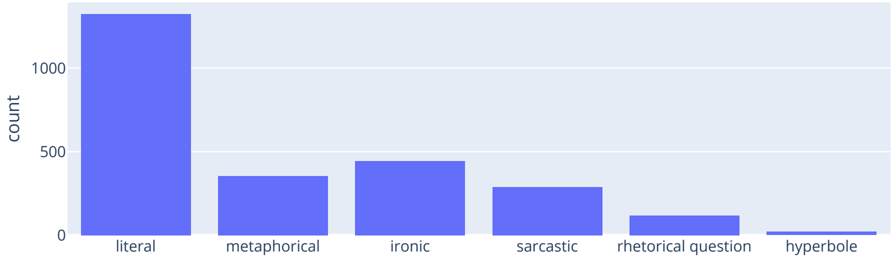


Fig. 5.11.: Histogram of *pragmatic function*.

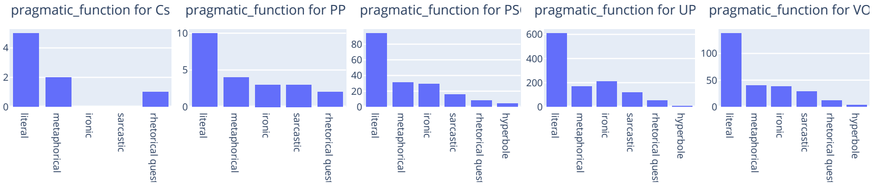


Fig. 5.12.: Per-party pragmatic function stats.

Mood

There are two possibilities:

- Realis, that are factual information or plain statements.
- Irrealis, that are situations or actions not known to have happened. It applies to imperative and interrogative clauses, desires, conditional clauses...

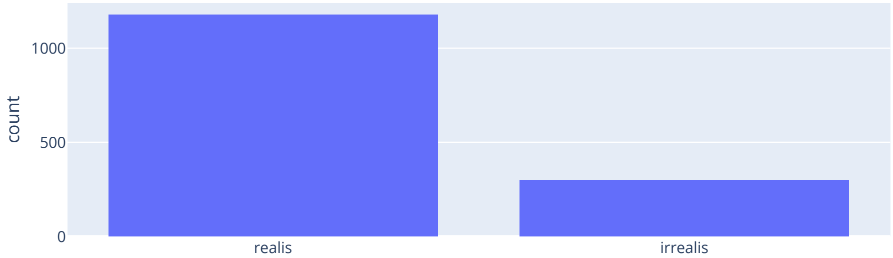


Fig. 5.13.: Histogram of *mood*.

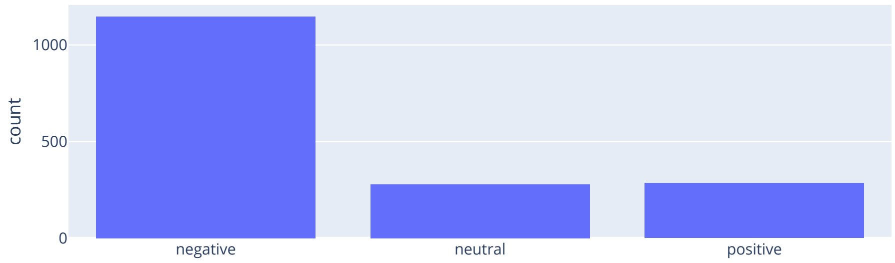


Fig. 5.14.: Histogram of *document sentiment*.

Document sentiment

It stands for the overall sentiment of the tweet. It is used to determine whether the writer's attitude towards a topic or entity is positive, negative or neutral.

There is a higher count of negative tweets, which means that the general attitude towards the event is negative. This is something expectable taking into account the overall wear of public opinion regarding politics, due to the fact that it was the second time that national elections were celebrated in 2019 as a results of the candidates inability to constitute a government.

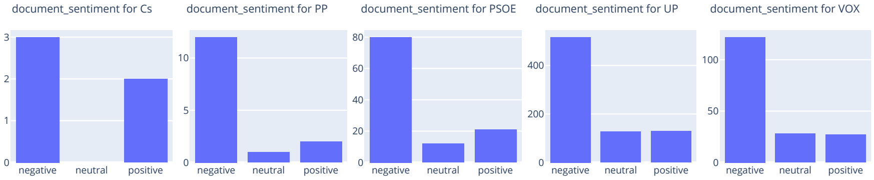


Fig. 5.15.: Per-party pragmatic function stats.

Implicit connotations

Whether a document have implicit emotions than differ from what is explicitly written. This category is used as a general label to encompass all tweets that cannot be taken literally. Subsequent labels are duplicated in terms of *connotation* and *denotation*, in order to better model what is written versus what the author means.

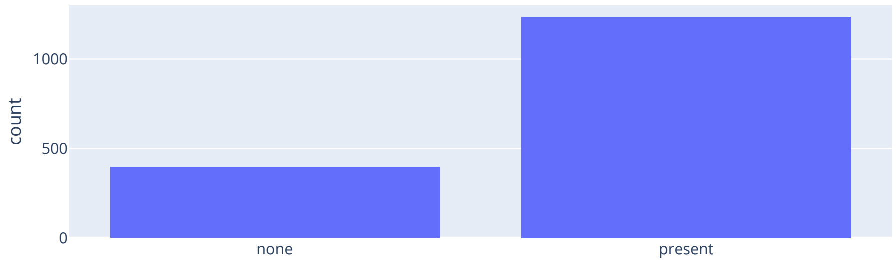


Fig. 5.16.: Histogram of *implicit connotations*.

Type of trigger

It refers to the nature of the entity that is causing the emotional reaction.

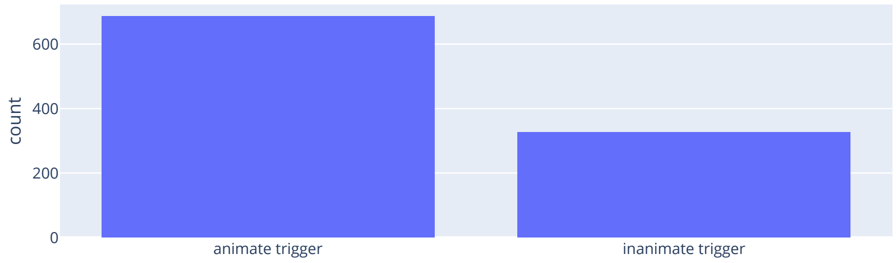


Fig. 5.17.: Histogram of *type of trigger*.

Surprise

Documents with words or expression of astonishment.

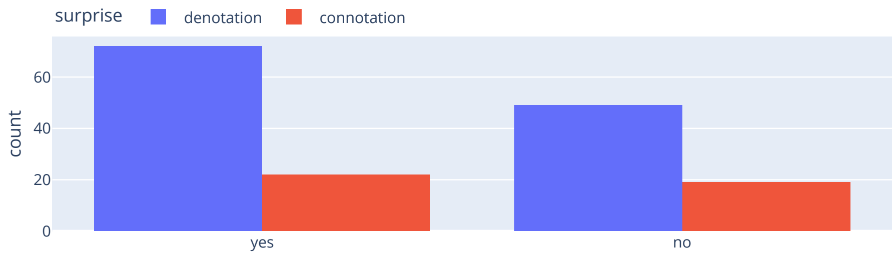


Fig. 5.18.: Histogram of *surprise*.

Interest

Documents with words or expression that express attraction towards something.

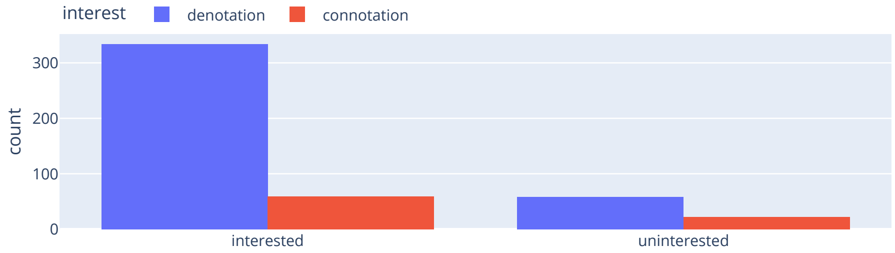


Fig. 5.19.: Histogram of *interest*.

Inclination

Documents with words or expression that express a special feeling towards something.

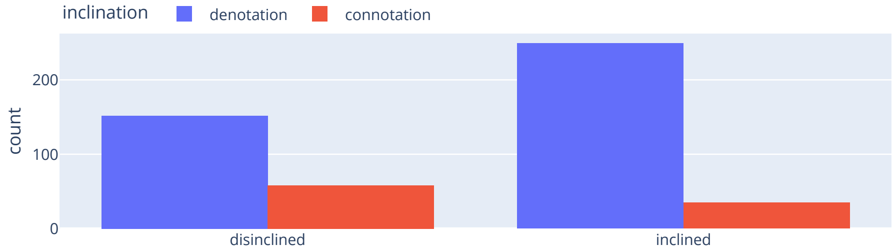


Fig. 5.20.: Histogram of *inclination*.

Security

There are nine possible values for this label:

- Anxious, that stands for eager of mental distress.
- Calm, when there are words or expressions manifesting peace or freedom from worry.

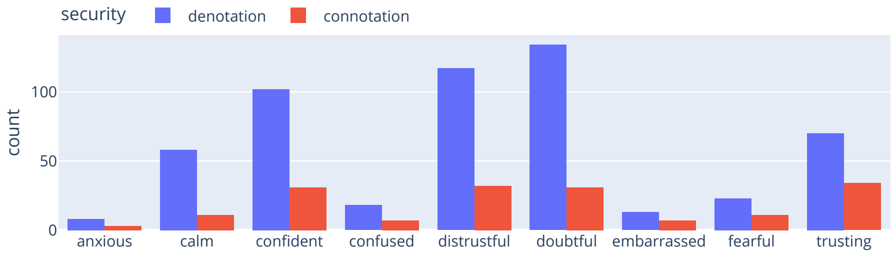


Fig. 5.21.: Histogram of *security*.

- Confident, when there is evidence of having full assurance.
- Confused, when the author manifest having difficulty understanding something.
- Distrustful, when the author express its inability to trust.
- Doubtful, that stands for uncertainty in the outcome.
- Embarrassed, when the author is ashamed.
- Fearful, when there are words or expressions of worry, concern or anxiety.
- Trusting, when the document express inclination to confide.

In this category, *doubtful* and *distrustful* are the most common emotions, followed by *confident* and *trusting* (see figure 5.21). Explicit content is significantly more popular than implicit connotations, although there are some differences among parties.

Figure 5.22 shows per-party *security* histograms. We do not present results for PP and Cs due to the small size of these samples. VOX show a higher rate of *denoted security* with respect to *connoted security*. Tweets related to this political party tend to be more explicit than the rest, while PSOE shows the highest levels of *distrust*, both in terms of denotation and connotation. *Confidence* for UP is higher than the rest, even surpassing *distrust*. *Fear* is also higher for the left-wing coalition, even more than PSOE and VOX together.

Happiness

The *happiness* category is composed of four possible values:

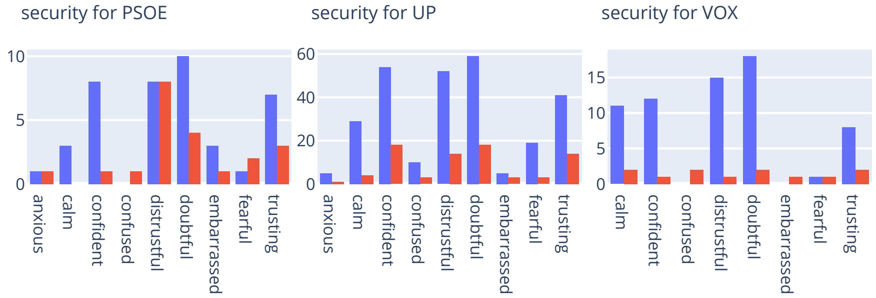


Fig. 5.22.: Per-party security stats.

- Angry, related to resentment.
- Frustrated, having a feeling of dissatisfaction.
- Happy, words or expressions or delight.
- Sad, when the author manifest unhappiness.

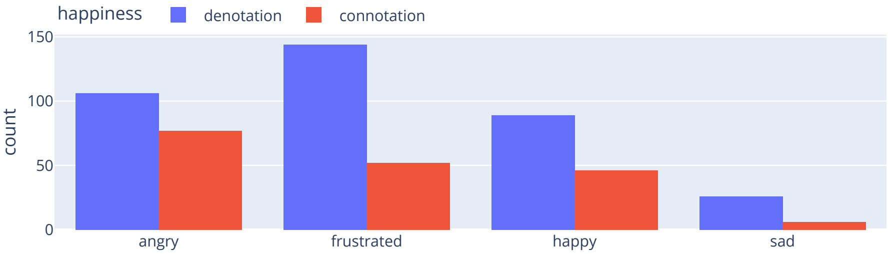


Fig. 5.23.: Histogram of *happiness*.

Figure 5.23 🖼 shows count histogram of *happiness*. *Frustration* and *anger* are the two most prevalent values, followed by *happiness* itself. Differences between political party are not as significant, however PSOE still shows higher rates of *connoted emotions* (see figure 5.24 🖼).

Liking

This is a binary class representing author's preference (or not) towards something.

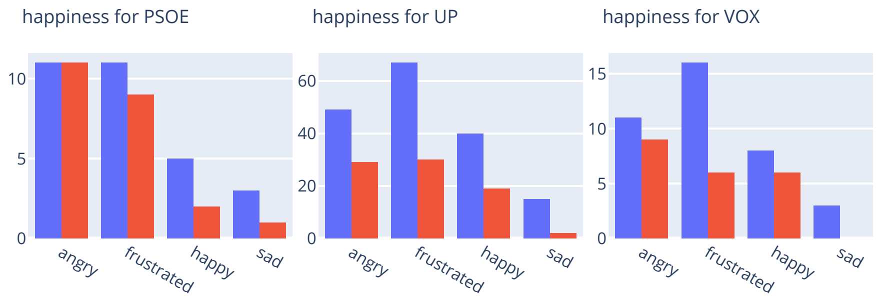


Fig. 5.24.: Per-party happiness stats.

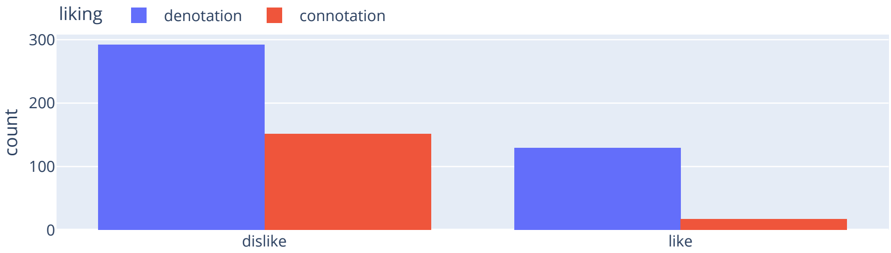


Fig. 5.25.: Histogram of *liking*.

Love

Another binary class representing author's passionate affection (or antipathy) towards something.

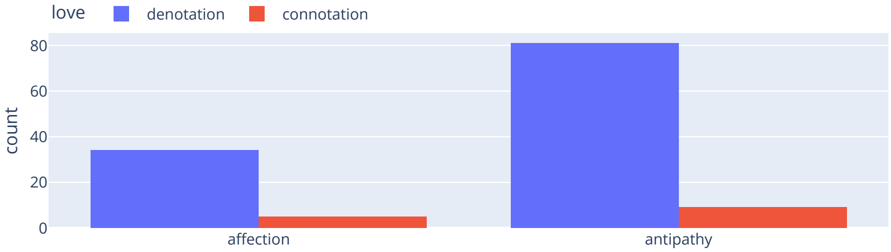


Fig. 5.26.: Histogram of *love*.

Respect

Respect is a binary category that stands for author's regard for something.

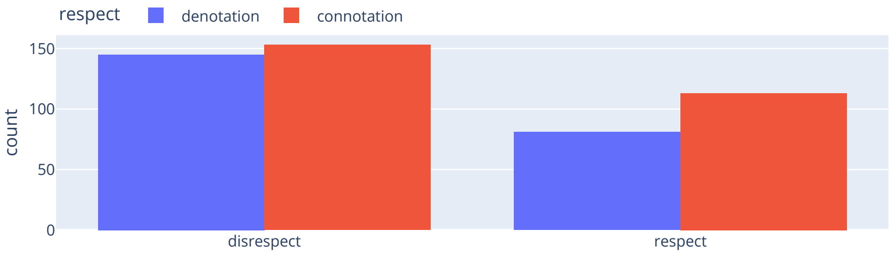


Fig. 5.27.: Histogram of *respect*.

Figure 5.28 shows that, even relative orders between category values are respected, there are some differences between parties. UP has the most similar counts between connoted and denoted emotions.

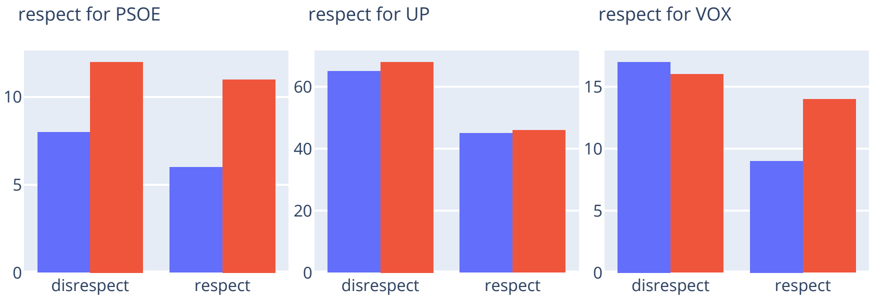


Fig. 5.28.: Per-party respect stats.

Sympathy

The *sympathy* emotion category refers to empathy or compassion, or lack thereof (indifference).

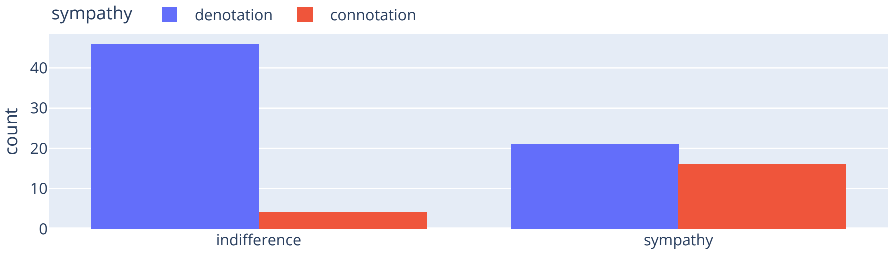


Fig. 5.29.: Histogram of *sympathy*.

In the context of this political debate, documents that show *sympathy* are half the ones that show indifference.

Tolerance

Tolerance is the quality of having an open attitude towards other people's beliefs or actions.

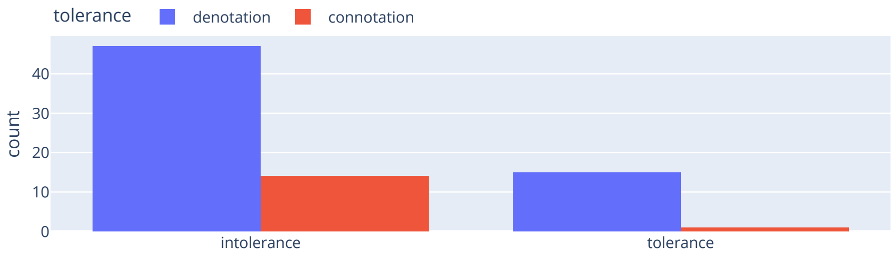


Fig. 5.30.: Histogram of *tolerance*.

Once again, *intolerance* is twice as common as tolerance. It is particularly interesting to observe that the two populist parties' related tweets are likely to be intolerant (see figure 5.31).

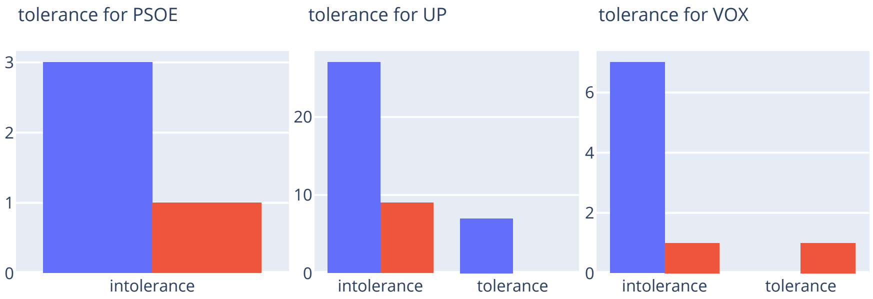


Fig. 5.31.: Per-party tolerance stats.

Type of appraised

This category refers to the nature of the entity that is assessed (both ethically and/or aesthetically).

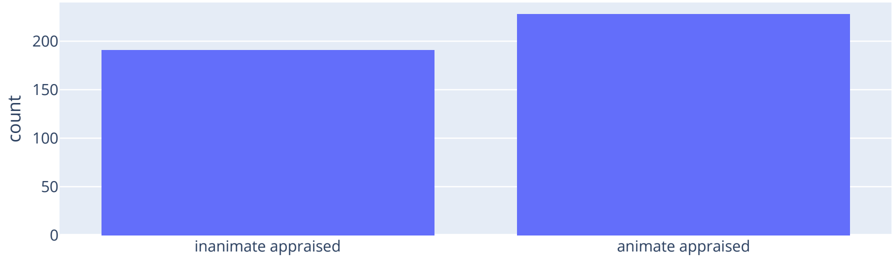


Fig. 5.32.: Histogram of *type of appraised*.

Although the prevalence of both classes are very similar, *animated entities* are slightly frequent.

Impact

Impact has two possible values, that are *dull* (boring, not lively) and *fascinating* (great attraction).

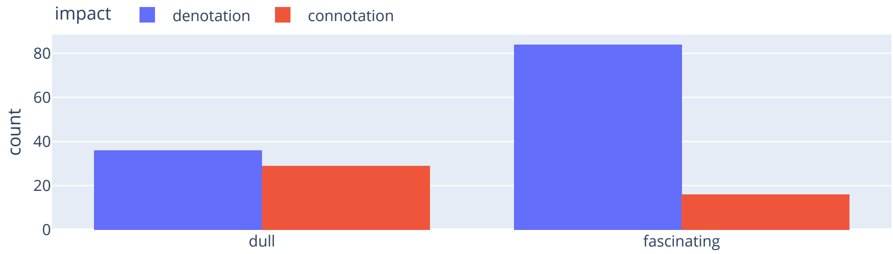


Fig. 5.33.: Histogram of *impact*.

Quality

Quality refers to a document being charming or beautiful to the senses.

Balance

Balance stands for a document that is *harmonious* (agreement of feeling or attitude) or discordant (disagreeing).

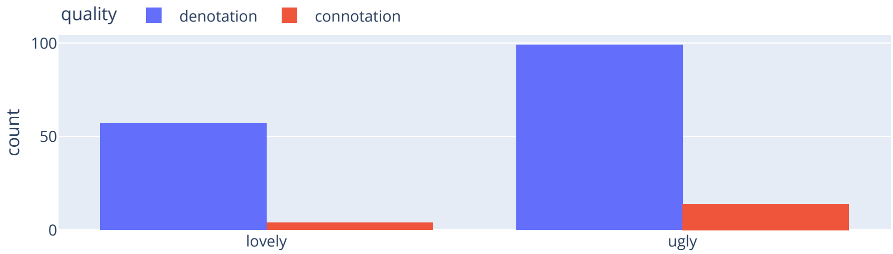


Fig. 5.34.: Histogram of *quality*.

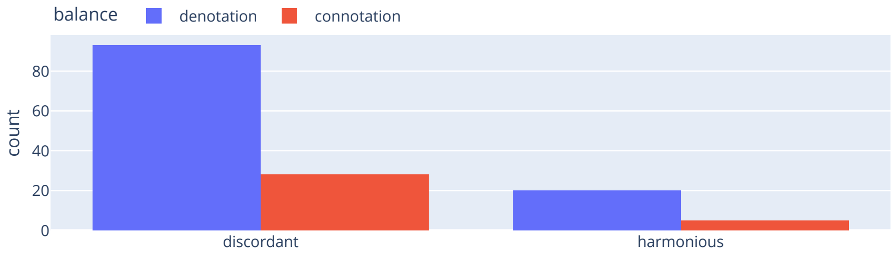


Fig. 5.35.: Histogram of *balance*.

Complexity

As its own name specifies, *complexity* stands for the author's perception of simplicity.

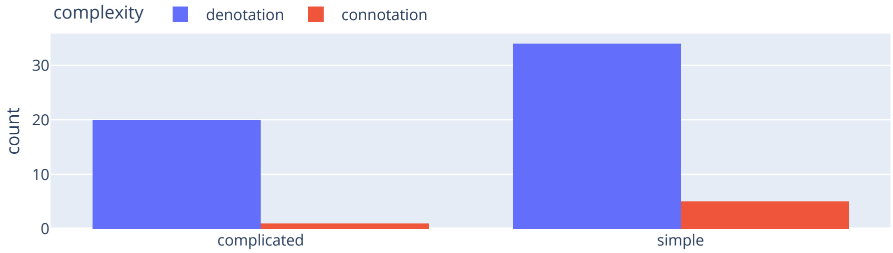


Fig. 5.36.: Histogram of *complexity*.

Significance

Significance speaks to the importance of the matter in question.

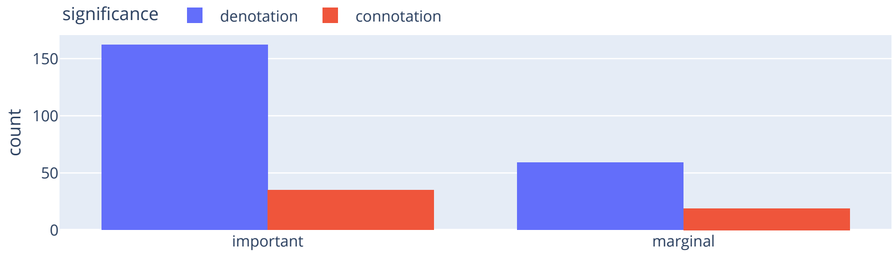


Fig. 5.37.: Histogram of *significance*.

Benefit

It relates to the purpose of the document, and whether it is constructive or not.

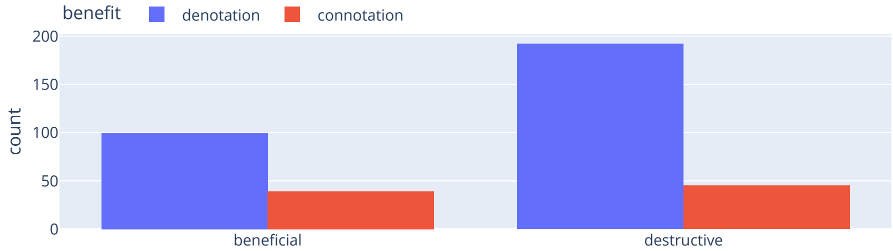


Fig. 5.38.: Histogram of *benefit*.

Propriety

This feature tries to answer the question "*how far beyond reproach?*".

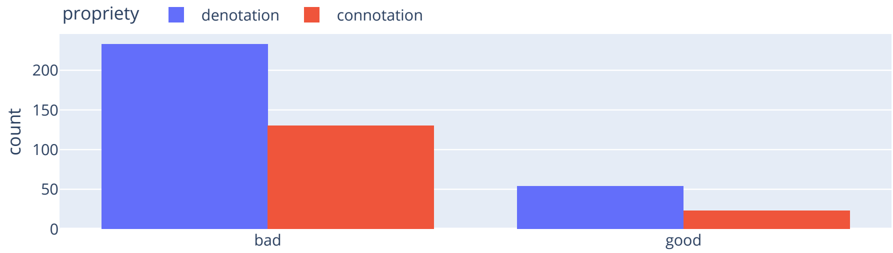


Fig. 5.39.: Histogram of *propriety*.

Documents annotated as *bad* are four times more frequent than *good* ones.

Veracity

Veracity makes reference to the observed truth in the statement.

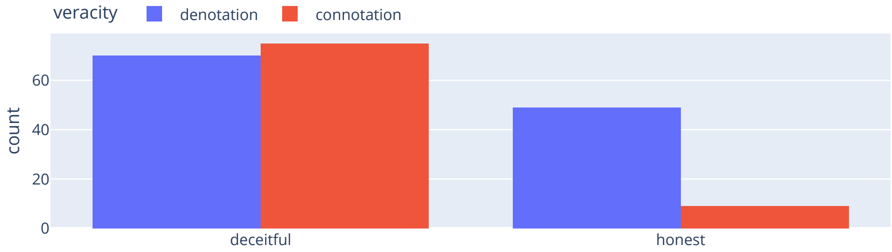


Fig. 5.40.: Histogram of *veracity*.

As expected from previous results, most authors manifest disappointment towards the topic, and there are no differences among political parties (see figure 5.41 🖼).

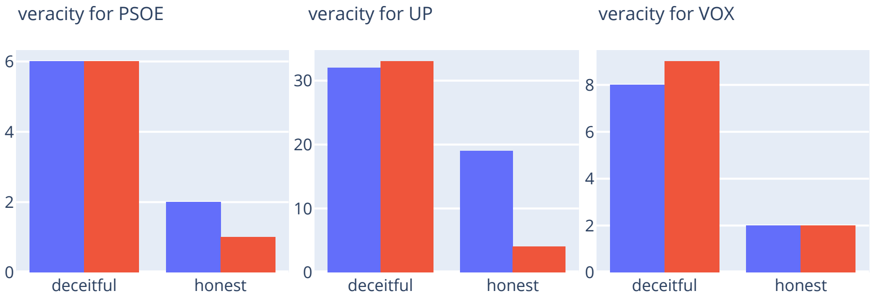


Fig. 5.41.: Per-party veracity stats.

Normality

This category tries to model the speciality of the matter.

Figure 5.42 🖼 shows that Twitter users tend to think that situations are abnormal almost twice as frequent with respect to normal statements. However, there are noticeable difference between UP and VOX, being the latter more balance with respect to the left-wing coalition and with respect to the general case (see figure 5.43 🖼).

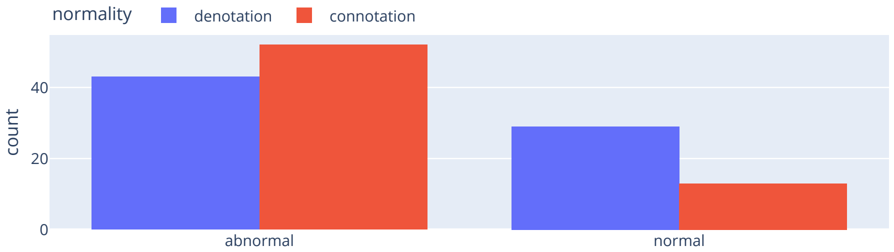


Fig. 5.42.: Histogram of *normality*.

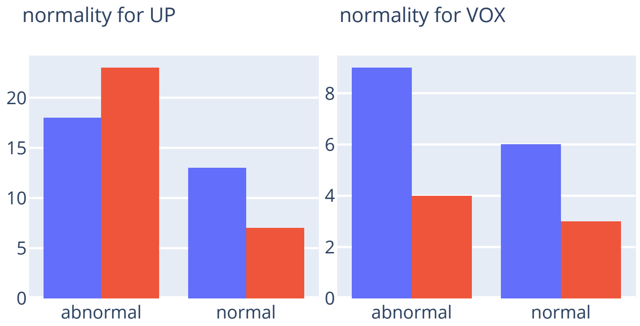


Fig. 5.43.: Per-party normality stats.

Capacity

Capacity is designed to label those documents that manifest the ability (or lack thereof) of someone to deal with certain situation.

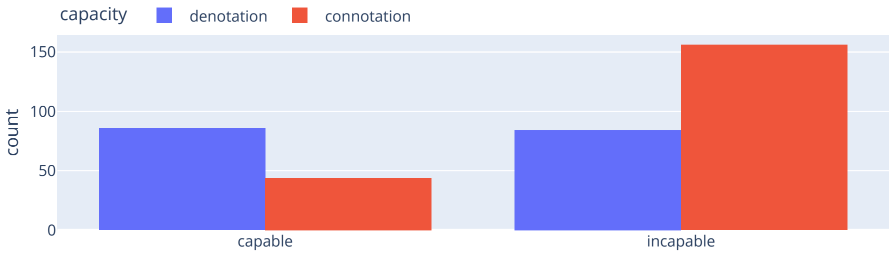


Fig. 5.44.: Histogram of *capacity*.

Implicitly, most tweets assess the capacity of the subject as *incapable*, however they do not explicitly say so (see figure 5.44 🖼). Figure 5.45 🖼 shows the same kind of behaviour between political parties, although UP shows a higher count of *denoted capability*.

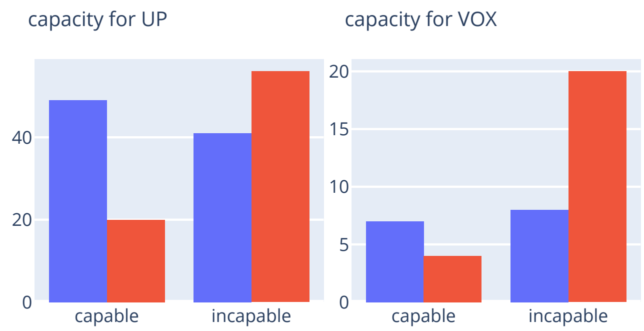


Fig. 5.45.: Per-party capacity stats.

Tenacity

This category tries to answer the question "*how dependable?*". The possible answers are *brave* (showing courage) or *cowardly* (lack of courage).

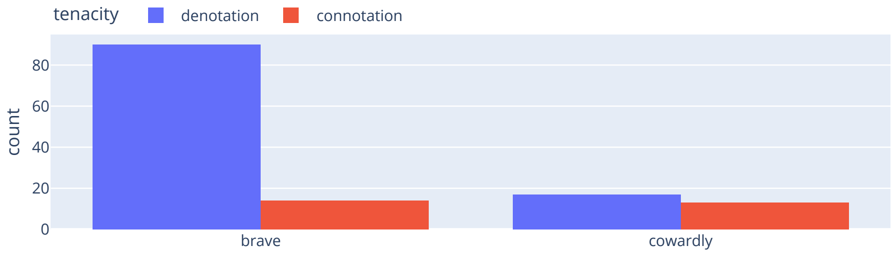


Fig. 5.46.: Histogram of *tenacity*.

Figure 5.46 shows that *braveness* is almost always manifested explicitly, and per-party aggregation shows that UP is the political party most related to the *brave* quality in terms of *tenacity* (see figure 5.47).

5.2 Correlation Analysis

There are several ways in which we can measure statistical dependencies between variables. Arguably, *Pearson correlation coefficient* is the most popular one [239]. Given two random variables a and b, *Pearson correlation coefficient* for a given sample is defined as in equation 5.1 [240].

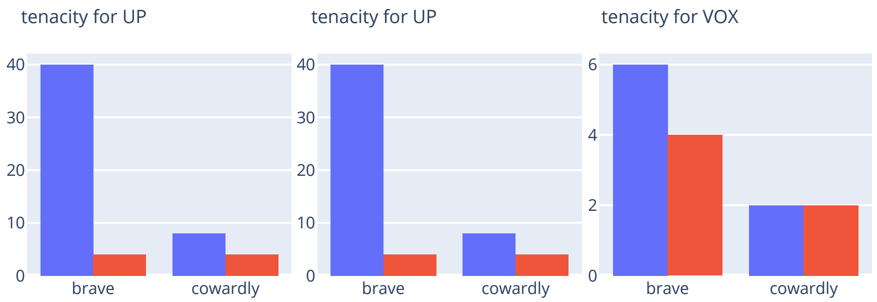


Fig. 5.47.: Per-party tenacity stats.

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}} \tag{5.1}$$

Consequently, *Pearson's correlation* is defined for numerical variables. Since our dataset is mainly categorical, it is required to use one-hot encoded features to obtain correlation coefficients. Figure 5.48 shows *Pearson's correlation* matrix between variables.

Clearly, there are four blocks of correlated features. We will focus on the upper triangle of the matrix, since it is symmetrical with respect to its diagonal. The first block is composed of general document features, such as gender and document sentiment. In the intersection between the horizontal and the vertical stripes of non-correlated features, we can detect another block, which stands for the propagated features. Since no other feature was propagated, it is not possible to obtain correlations with them. Third block correspond to correlations between general document features and emotions, while the fourth (and biggest one), stands for correlations between emotions.

Unfortunately, since one-hot encoding results in a high number of binary features, it is more difficult if not impossible to extract conclusions from the matrix. Hence we will reduce the matrix to focus on highest correlations.

From here on, we will refer to users/documents that have positive sentiment towards a political party with the acronym of that party followed by a $+$ sign (e.g., PP+, UP+), and analogously, followed by a $-$ sign to denote negative sentiment (e.g., PSOE-, VOX-).

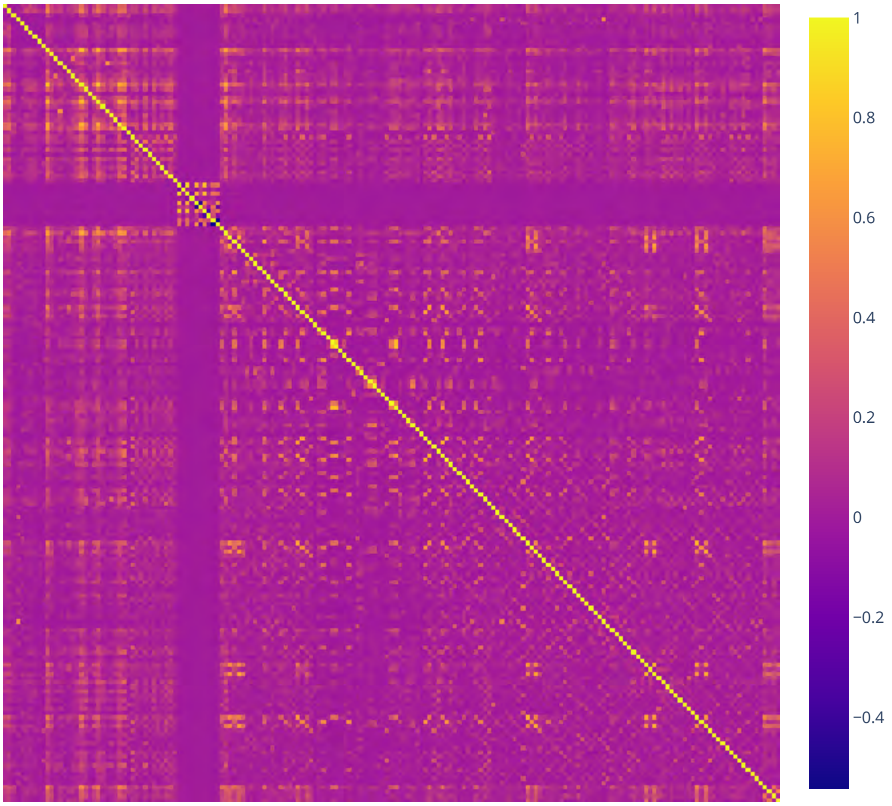


Fig. 5.48.: Pearson's Correlation Matrix using one-hot encoded features. Feature names have been removed due to size, however we will review each block separately.

Which features are correlated with propagated labels?

Figure 5.49 shows top correlations between propagated parties. Starting with *Ciudadanos*, those that have a negative sentiment towards them (Cs-) are strongly correlated with PP-, PSOE- and VOX- (with a coefficient higher than 0.7 in all three cases). Although lower, there is also a correlation between Cs- users and those who have any kind of sentiment towards UP, regardless of it being positive or negative (coefficient above 0.4 in both cases).

In the case of *Partido Popular*, PP- users also have negative sentiment for Cs, PSOE and VOX, with a coefficient higher than 0.7 in all three cases. Analogously to Cs, the party shows correlations with those users that are interested in UP, regardless of their sentiment being positive or negative.

Partido Socialista Obrero Español is the first party that presents a different behaviour. PSOE- documents are also Cs-, PP- and VOX-. However, in this case, there are less correlation with those users who are against UP, being the lowest coefficient until now. On the contrary, those that are PSOE+ present a slightly negative correlation with UP+ ones.

Unidas Podemos is the party whose sample is most representative, and its correlations are straightforward to read. Most of the users that have negative sentiment towards UP have affinity to PSOE+ or VOX+, although there are also small positive correlation coefficients for PP and Cs. Analogously, UP+ documents tend to have negative sentiment towards the right-wing parties and PSOE.

VOX behave in the same fashion. Those users that have negative sentiment towards any of the three right-wing parties are also VOX-. UP+ users are also highly correlated with VOX-. Users that have affinity to VOX are also correlated with PP and Cs, however it presents negative coefficients with PSOE and UP (left-wing).

There is almost no correlation with emotions. However, it is not possible to conclude that political parties are not correlated with them. These results were expected since these features were not propagated, therefore annotations are partial and they are not suited to extract conclusions. However, it shows that the property expansion mechanism is consistent to what we could expect from reality.

Which features are correlated with manually-annotated parties?

Figure 5.50 shows top correlations between political parties and the rest of manually-annotated labels. In this case, it is possible to find correlations with emotions, since manually-annotated tweets present the full range of features.

Appreciation categories (i.e., impact, quality, balance, complexity, significance and benefit) present mild correlation coefficients with UP+ and PP-. In the case of PSOE+, its users prefer to *denote* appreciation rather than leaving it implicit. In the case of Attention-grabbing categories (i.e., surprise, interest, inclination), only left-wing parties seem to be correlated with them.

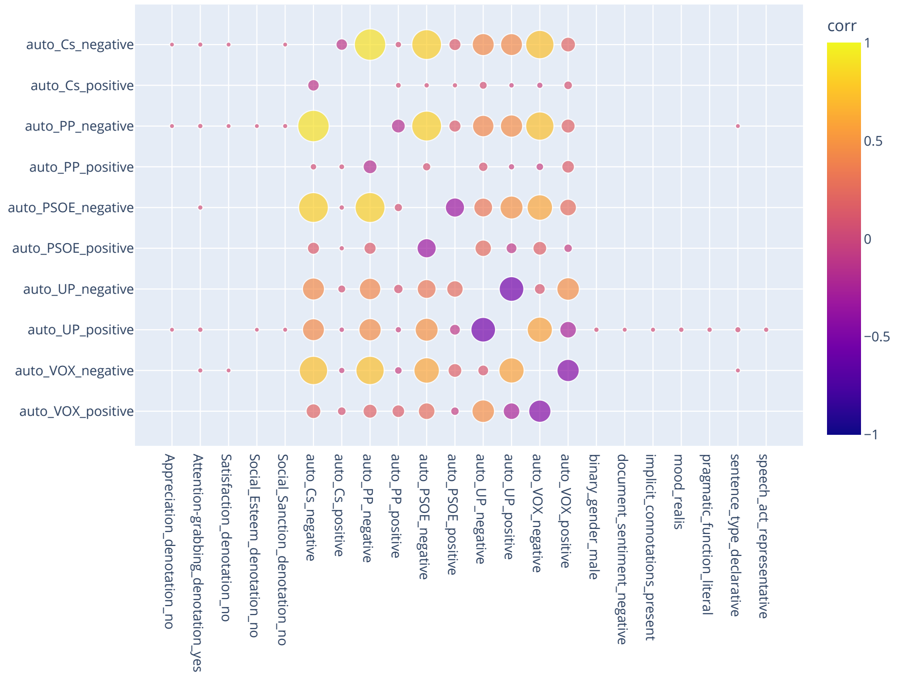


Fig. 5.49.: Highest correlations between propagated political parties and emotions.

An interesting fact, that aligns with the campaign of the left-wing parties, is that those users that are against one right-wing party are also against the other two. This was part of the left campaign, that used to call them "*trifachito*", suggesting that *a vote for one of them is the same as a vote for the other two since the three of them were fascists*.

PSOE+ users are the only ones showing *inclination*. VOX- are the only users that are correlated with *bad propriety* and *disrespect*, they use both *literal* and *metaphorical* pragmatic functions and are not only correlated with PP- and Cs- but also with PSOE-, which suggests that these user are UP+. Cs- are exclusively showing correlation with *sarcasm*, *ironic*, *disrespect* and *incapable* connotations.

Does aggregation show us something different?

If we group tweets under the *left-wing* and *right-wing* categories, as well as *populists* versus *traditional* parties, correlations decrease. The top coefficient correspond to *positive sentiment* towards *left-wing parties* with a value of 0.51,

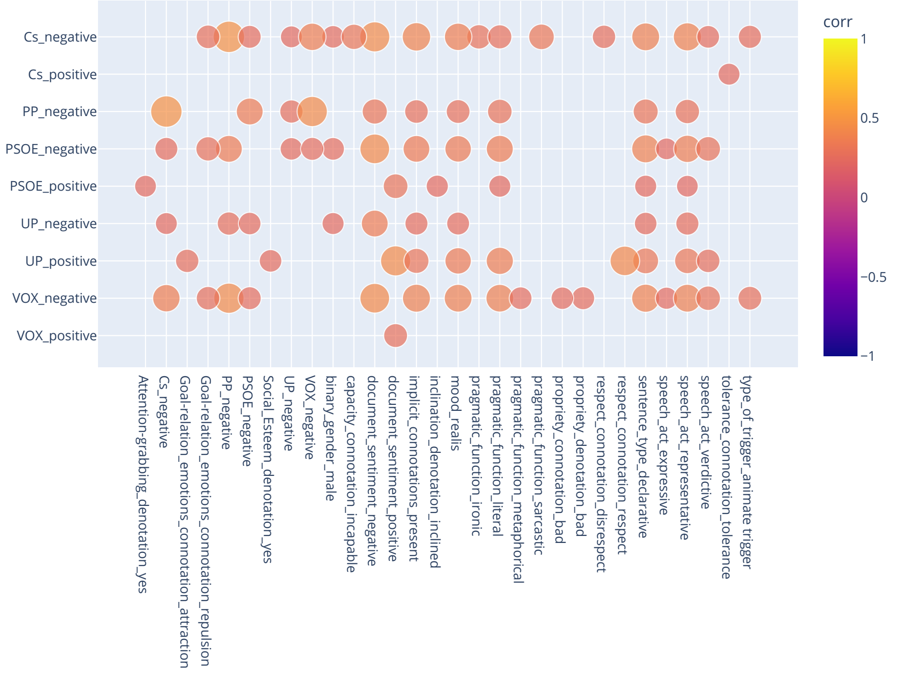


Fig. 5.50.: Highest correlations between manually-annotated political parties and emotions.

that translates to *medium dependency*. However, this cannot be used to draw any conclusions since the aggregation forces positive-sentiment tweets into the groups.

The main difference between left- and right-wing parties is *capability*. Right-wing+ tend to denote the *capacity* of their candidates, while left-wing+ users are correlated with a full range of emotions that do not include *capacity*. Samples are quite imbalanced, therefore more annotations are required to extract conclusions.

If we compare populist with traditional parties, there is a mild correlation of the former with *males* and *connoted* emotions. Traditional parties tend to be correlated with denoted emotions and literal pragmatic functions, being populist parties the only ones using functions like *metaphorical* or *verdictive* sentences.

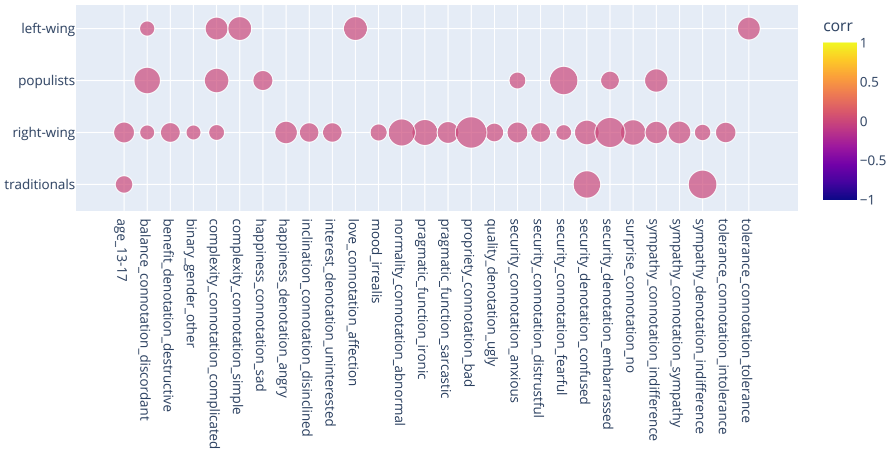


Fig. 5.51.: Highest correlations between aggregated parties and emotions.

5.3 Frequent Pattern Analysis

Frequent patterns are *itemsets* that appear in a dataset with a certain frequency [241]. Most common example is *market basket analysis*, in which customer's baskets are mined to find which products are purchased together (e.g., bread and milk) and, ultimately, build different profiles for buying habits. Association rules are often used to find and extract such patterns [242].

The first step in the process is to mine frequent itemsets. That means, finding patterns of item associations or correlations from a database. There are two popular algorithms to find those itemsets:

- Apriori algorithm, which is based on the principles that non-empty subsets of a frequent itemsets are also frequent; and that any superset of a non-frequent itemset is, also, non-frequent. Therefore, it counts the frequency of itemsets of size $i_{i=1}^{k}$ only considering those where all the subsets of length $i - 1$ are frequent. However, it present several drawbacks, such as the number of candidates may be huge and it needs k iterations over the database.

- FP-Growth (Frequent Pattern Growth), that avoids candidate generation by using a Frequent-Pattern tree (FP-tree). It first scans the database to build the data structure and the uses the FP-tree to find recurrent itemsets. Since it compresses the dataset using a tree and avoid multiple passes trough the database, it is particularly attractive for large databases.

Association Rules Quality Measures

Once frequent itemsets have been extracted, association rules are generated. For each itemset X, let $Y \subset X : Y \neq \emptyset$ be the antecedent of the rule, such that $Y \Rightarrow (X - Y)$, with $(X - Y)$ being the consequent of the rule. The frequency of the association rule in the database is called **support**. The robustness of the implication is call **confidence**.

Given a rule $A \Rightarrow C$ extracted from a dataset D, we define support and confidence such that:

$$\textit{support}(A \Rightarrow C) = P(A \cup C) = \frac{|\{t \in D : A \cup c \subseteq t\}|}{|D|} \tag{5.2}$$

$$\textit{confidence}(A \Rightarrow C) = P(C|A) = \frac{support(A \cup C)}{support(A)} \tag{5.3}$$

where t are transactions (or instances) of the database D.

In order for an association rule to be considered strong, it has to reach certain user-defined thresholds of support and confidence. However, *confidence* present a major drawback, that is that if two events are unrelated, $P(B|A)$ can be equal to $P(B)$ [243].

Lift measures the deviation of the rule from the statistical independency of the antecedent and consequent. Lift will be 1 if A and C are independent.

$$\textit{lift}(A \Rightarrow C) = \frac{\textit{confidence}(A \Rightarrow C)}{\textit{support}(C)} \tag{5.4}$$

Leverage measures the difference between the frequency of A and C co-occurring versus the expected frequency if A and C were independent. A value of 0 implies independency [244].

$$\textit{leverage}(A \Rightarrow C) = \textit{support}(A \Rightarrow C) - \textit{support}(A) \times \textit{support}(C) \tag{5.5}$$

Conviction measures the consequent dependency on the antecedent. If the clauses are independent, then conviction is 1 [243].

$$\mathit{conviction}(A \Rightarrow C) = \frac{1 - \mathit{support}(C)}{1 - \mathit{confidence}(A \Rightarrow C)} \tag{5.6}$$

Political Parties Association Rules

We applied FPGrowth and extracted association rules from frequent itemsets using our whole dataset of tweets. Table 5.1 ▦ shows association rules of one itemset in the antecedent and consequent, and it reveals some expected behaviour of our expansion algorithm.

Tab. 5.1.: Party-related association rules.

antecedents	consequents	ant. supp.	consq. supp.	supp.	conf.	lift	lvrg.	conv.
■ auto UP: positive	■ auto VOX: negative	.342	.595	.335	.982	1.651	.132	22.421
■ auto PP: negative	■ auto Cs: negative	.683	.713	.673	.986	1.382	.186	19.794
■ auto UP: positive	■ auto Cs: negative	.342	.713	.336	.983	1.378	.092	16.906
■ auto VOX: positive	■ auto UP: negative	.118	.363	.114	.960	2.648	.071	16.127
■ auto VOX: positive	■ auto PSOE: negative	.118	.655	.116	.977	1.493	.038	15.273
■ auto VOX: negative	■ auto Cs: negative	.595	.713	.583	.981	1.376	.159	15.181
■ auto PSOE: negative	■ auto Cs: negative	.655	.713	.638	.974	1.365	.171	10.941
■ auto UP: negative	■ auto Cs: negative	.363	.713	.353	.973	1.364	.094	10.660
■ auto UP: positive	■ auto PP: negative	.342	.683	.331	.969	1.420	.098	10.325
■ auto UP: positive	■ auto PSOE: negative	.342	.655	.330	.966	1.476	.106	10.158
■ auto VOX: negative	■ auto PP: negative	.595	.683	.569	.958	1.402	.163	7.466
■ auto PSOE: negative	■ auto PP: negative	.655	.683	.624	.953	1.395	.177	6.696
■ auto VOX: positive	■ auto Cs: negative	.118	.713	.113	.954	1.337	.028	6.206
■ auto Cs: negative	■ auto PP: negative	.713	.683	.673	.943	1.382	.186	5.598
■ auto UP: negative	■ auto PP: negative	.363	.683	.340	.937	1.372	.092	5.002

Continued on next page

Tab. 5.2.: Observed affinity among the candidates during the political debate. + stands for positive affinity, - for negative affinity, ± for both possibilities. For example, PSOE followed a strategy in which they were positive towards Cs and negative towards the rest of the candidates.

affinity	UP	PSOE	Cs	PP	VOX
UP+	+	±	-	-	-
PSOE+	-	+	±	-	-
Cs+	-	±	+	±	±
PP+	-	-	±	+	±
VOX+	-	-	-	-	+

Tab. 5.1.: Party-related association rules.

antecedents	consequents	ant.	consq.	supp.	conf.	lift	lvrg.	conv.
		supp.	supp.					
■ auto PP: negative	■ auto PSOE: negative	.683	.655	.624	.913	1.395	.177	3.991
■ auto VOX: positive	■ auto PP: negative	.118	.683	.107	.907	1.328	.026	3.397
■ auto Cs: negative	■ auto PSOE: negative	.713	.655	.638	.894	1.365	.171	3.252
■ auto VOX: negative	■ auto PSOE: negative	.595	.655	.525	.882	1.348	.135	2.936
■ auto PP: negative	■ auto VOX: negative	.683	.595	.569	.834	1.402	.163	2.441
■ auto UP: negative	■ auto PSOE: negative	.363	.655	.310	.855	1.306	.073	2.382
■ auto Cs: negative	■ auto VOX: negative	.713	.595	.583	.818	1.376	.159	2.226
■ auto PSOE: negative	■ auto VOX: negative	.655	.595	.525	.801	1.348	.135	2.041
■ auto UP: negative	■ auto VOX: negative	.363	.595	.241	.663	1.115	.025	1.204

Taking some examples, those users that are UP+ are also VOX- with a confidence of 98.2%. Such top confidence also occurs with PP- and Cs-, UP+ and Cs-, VOX+ and PSOE-. These rules can be summed up with (see table 5.2 ▦), that represents the affinity observed among candidates during the political event.

However, in order to extract additional rules, it is necessary to restrict the database to the set of fully-annotated tweets. Since most of our dataset is weak-labelled, those labels that are not related to political parties are mostly empty (they cannot be propagated using the same deep relation). This reduces the support of any itemset related to emotions to levels in which is unfeasible to manage all frequent itemsets (minimum support under 0.01). Therefore, from here upon, we will use the subset of tweets that were manually annotated.

Emotion-related Association Rules

If we restrict the subset of tweets to those that were fully annotated, we obtain several association rules.

Arguably, the most relevant ones evidence a clear relation between *mood realis*, *implicit connotations* and *negative sentiment*. The connection between *negative sentiment* and *implicit connotations* suggests that both qualities are closely related in political topics.

Tab. 5.3.: Association rules with non-party antecedents or consequents.

antecedents	consequents	ant. supp.	consq. supp.	supp.	conf.	lift	lvrg.	conv.
■ Appreciation denotation: no	■ Social Esteem denotation: no	.273	.282	.215	.788	2.795	.138	3.382
■ Social Esteem denotation: no	■ Appreciation denotation: no	.282	.273	.215	.764	2.795	.138	3.080
■ mood: realis ■ document sentiment: negative	■ implicit connotations: present	.313	.558	.234	.747	1.340	.059	1.751
■ document sentiment: negative ■ implicit connotations: present	■ mood: realis	.329	.534	.234	.712	1.333	.059	1.619
■ mood: realis ■ implicit connotations: present	■ document sentiment: negative	.356	.518	.234	.657	1.268	.049	1.405
■ mood: realis	■ implicit connotations: present	.534	.558	.356	.667	1.196	.058	1.328
■ implicit connotations: present	■ mood: realis	.558	.534	.356	.639	1.196	.058	1.290

Continued on next page

Tab. 5.3.: Association rules with non-party antecedents or consequents.

antecedents	consequents	ant. supp.	consq. supp.	supp.	conf.	lift	lvrg.	conv.
■ document sentiment: negative	■ implicit connotations: present	.518	.558	.329	.634	1.137	.040	1.209
■ binary gender: male	■ implicit connotations: present	.356	.558	.225	.631	1.131	.026	1.198
■ document sentiment: negative	■ mood: realis	.518	.534	.313	.604	1.131	.036	1.177

6 Conclusions

Our society is living unprecedented times in which access to information is easier than ever. The popularisation of Internet through portable devices, such as smartphones, and the ubiquitous access to Social Networking Sites are quickly shaping our habits into an hyper-connected, always-online routine. SNS help people stay informed, organise, collaborate, share knowledge and keep in touch with each other. Each and every one of these actions can be logged, and the spike of stored information is the imprint of today's generation. But the repercussions do not stop there.

The content that we generate, the *likes* that we give away, the interactions we make with each other, our location or even the sensors of our smartphones can be mined, aggregated and analysed. Machine Learning (ML) is untangling the knots of our behaviour through millions of databases that are used to train models who are able to predict things that we might not even know of ourselves. Our interest and preference can be discovered just by analysing the data that we produce but, despite that ML techniques and potential use cases are rapidly growing, they still require a lot of manual work, both to build them and also to understand them.

Precisely, this is the main goal of our work. This PhD dissertation aimed to develop techniques, methodologies and tools to detect, track, monitor and analyse groups of interest in SNS with a reduced effort and high interpretability. Our work is part of the objectives of a larger venture, The Nutcracker Project, that is a multidisciplinary project in the fields of political science, linguistics and artificial intelligence. In particular, the project's intent is to study the political and ideological radicalisation process and the nature of its discourse, that is closely related to emotive persuasion. In order to do that, we have conducted several experiments in three different fields, ranging from weak supervision to class prevalence estimation.

In terms of effort reduction, we introduced *Similarity Semantic Networks* that, apart from representing knowledge, enable a new way of deductive reasoning based on similarity measures. We proposed a materialisation of these networks for Twitter, using a *deep relation* representing *similar ideology with respect to a topic*. In order to obtain a proxy function for that high-level relation, we introduced the *co-retweet* graph. We used it to propagate labels from known users to unknown ones, with the purpose of supervising a dataset. Since the process is iterative and there may be humans involved, we introduced our weak-supervision methodology and tested it while producing 4 tweet databases. Our results show that our tool is able to reduce the initial effort by two orders of magnitude with an accuracy above 80% in all our experiments.

But that is not the only capability of our tool. Nutcracker is able to use trained models for label prediction or subject of interest **detection**. The tool includes a **tracking** utility that is able to use the *co-retweet* graph to run a preliminary property expansion and retrieve content from potentially relevant users. If any of those users are confirmed as relevant, the tool schedules them for regular updates. Therefore, it is also able to **monitor** relevant clusters. Finally, it is possible to annotate the content, both manually and using trained models, to obtain a supervised dataset that enables the **analysis**.

In terms of model comprehensibility, we determine that current interpretable models result in a large number of complex rules that are difficult for humans to read and comprehend. Since ML models look for patterns in the input feature space that are good predictors of the class, we introduce *distinguishing expressions* as a new kind of feature. These are sequence of words that are biased towards one of the classes but not the others. Consequently, it is possible to use less but more meaningful features to feed the classifier. Since the input space is smaller, the classifier is able to maintain the same performance while significantly reducing the complexity of the model by an order of magnitude.

To improve aggregation bias, we ran a comparative study between different quantifiers and types of data drift. PCC and PACC were the most competitive quantifiers, with PCC winning when *covariate shifts* are present and PACC outperforming the rest for *prior probability shifts*. We also proposed *similarity-based quantifiers*, that tried to adjust the probabilistic posterior addition using similarity measures. However, results did not improve the baselines, suggesting that other approaches are necessary.

When aggregating data using time-based windows, PCC is also the best quantifiers. This was something to expect taking into account that we used data from a political debate that explored different topics, therefore there would be a *covariate shift* present between windows from different topics. We tried a new approach that consisted on the prediction of *true* and *false positive rates* in order to make a better adjustment in the posterior probabilities with the purpose of tailoring the prevalence estimates to the specific windows. However, once again, our results were not good enough. This opens the door to new research lines, that can benefit from the applications of techniques from different fields, such as *label propagation*.

Finally, we also managed to produce 4 weak-labelled dataset in several topics and languages. We conducted an exploratory analysis on one of them, in terms of feature distribution, feature co-relations and frequent patterns. We also conducted a longitudinal study on overall sentiment during the Spanish 2019 Political Debate, and we set the landmarks to integrate such functionality in Nutcracker.

The products of this PhD dissertation are useful for researchers of multiple disciplines, such as political science, linguistic and cybersecurity. Evidence of this is that they are being used by our team's political scientists and linguists, and that it has raised the attention of the Spanish Civil Guard, that are considering our platform for a future project. The proposed techniques and the consequent methodologies and tools constitute an interpretable solution to retrieve, supervise and analyse social network data with a reduced effort.

6.1 Contributions

The applications of the techniques and methodologies presented in this work are multiple. We sum up below our main contributions:

- We introduced *Similarity Semantic Networks*, that are able to represent knowledge while enabling new reasoning mechanisms based on similarity measures.

- We developed an interpretable methodology built upon Similarity Semantic Networks with the purpose of producing weak-labelled databases through property expansion. This methodology is able to considerably reduce the effort required to assemble the dataset while maintaining an accuracy of at least 80%.

- We introduce the *co-retweet* graph, as a materialisation of a similarity semantic network using a *deep relation* that approximate the value of the high-level relation for *similar ideas*.

- We implemented a *proof-of-concept* tool that is able to *detect*, *track* and *monitor* relevant clusters of users. It is also capable of applying the aforementioned methodology to effortless produce datasets.

- We provided a comparative analysis on how comprehensible are trained interpretable models, and we established an ideal complexity based on the capabilities of human working memory.

- We presented a new ranking method for words, CF-ICF, that is able to model the utility of individual features as class predictors.

- We introduce the concept of *expression* and *distinguishing expression*, as sequence of words that are biased towards a class. These features reduce the input space and produce less complex models that are as accurate as the rest but more comprehensible.

- We studied and provided a list of different types of bias that may be present when using SNS public APIs, and we classified them with respect to the nature of the bias.

- We reviewed other's author's proposal to deal with prevalence estimation in SNS, particularly in Twitter. We ran a comparative study to determine which of the quantifiers are suitable to deal with different kinds of data drift.

- We determined that most straightforward approaches to adjust prevalence estimates based on spatial an time similarities are not good enough to improve the baselines, therefore opening a new line of research.

6.2 Future Work

Our findings opened up several future lines of work. In this section, we sum up the most important ones.

In terms of effort reduction, there is still a lot of ground to cover. Similarity Semantic Networks are able to hold representations using more than one high-level relation. The expansion of properties can benefit from a multi-layer graph by propagating labels using compatible relations and ignoring the rest. We expect that this improvement not only enhances the reach of the expansion but also the accuracy of the deductions.

As for model comprehensibility, distinguishing expressions may benefit from the use of *simsets* and/or *embeddings*, both for the word and its context. The algorithm, as is, cannot be applied to work with embeddings, and interpretability will be lost in such scenario. However, it is possible to introduce synonyms to the expressions to improve its relevance and accuracy. Unfortunately, this would also increase computational time required to extract and select those features. Experiments on the trade-off between usefulness and computational time should be conducted to explore whether this is a valid approach.

Quantification methods may also benefit from spatio-temporal similarity measures. Prevalence count estimations are adjusted taking into account misclassification rates, but these vary depending on the sample. Establishing and delimiting regions of the input space with different misclassification rates can improve the adjustment and therefore yield better prevalence estimates.

The full range of applications of the techniques and tool developed during this dissertation are yet to be established. Potential use cases are wide, including detection of profiles related to events (such as those users affected by natural disasters), detection of profiles with certain features (such as those users that are keen to facilitate the spread of fake news), delimitation of clusters of users with certain features (such as those that are vulnerable to radical or populist discourse), size estimation of those clusters, characterisation of the evolution of sentiment during an event, assessment of the effect of political strategies in public opinion, and virtually any use that can benefit from the analysis of SNS. The effort reduction that results from the application of our techniques and the interpretability of these enable many low-budget research scenarios and applications in situations that may have a social impact.

6 Conclusiones

Nuestra sociedad vive un punto de inflexión en el que el acceso a la información es más sencillo que nunca. La popularización de Internet a través de dispositivos portátiles, como nuestros *smartphones*, y el acceso a las redes sociales desde cualquier lugar, están modificando nuestros hábitos hacia un mundo hiperconectado y siempre *online*. Las redes sociales nos permiten estar informados, organizarnos, colaborar, compartir conocimiento y permanecer en contacto. Cada una de estas acciones puede ser registrada, y la explosión de información almacenada es la huella de nuestra generación. Pero las consecuencias no paran aquí.

El contenido que generamos, los *likes* que regalamos, las interacciones que hacemos con otras personas, nuestra localización e, incluso, los sensores de nuestro *smartphone*, pueden ser minados, agregados y analizados. El aprendizaje computacional (ML) está desenredando la maraña de nuestro comportamiento a través de millones de bases de datos usadas para entrenar modelos que sean capaces de predecir detalles de los que quizá ni siquiera nosotros somos conscientes. Nuestros intereses y preferencias pueden ser descubiertos simplemente analizando los datos que producimos pero, a pesar de que las técnicas y los casos de uso crecen vertiginosamente, requieren una gran cantidad de trabajo manual tanto para construirlos como para entenderlos.

Este es, precisamente, el objetivo principal de nuestro trabajo. Esta tesis pretende desarrollar técnicas, metodologías y herramientas para detectar, rastrear, monitorizar y analizar grupos de interés en redes sociales, con un esfuerzo reducido y una alta interpretabilidad. Nuestros objetivos se engloban dentro de un proyecto más grande, llamado Nutcracker, que es un proyecto multidisciplinar entre las ramas de ciencias políticas, lingüística e inteligencia artificial. En particular, el proyecto intenta estudiar el proceso de radicalización ideológica y política y la naturaleza de su discurso, que está estrechamente relacionado con la persuasión emocional. Para ello, hemos llevado a cabo numerosos experimentos en tres líneas diferentes, que van desde el etiquetado asistido de datos hasta la estimación de la prevalencia de las clases en un conjunto de datos.

En términos de *reducción del esfuerzo*, hemos presentado las *Similarity Semantic Networks* que, a parte de representar conocimiento, habilitan un nuevo mecanismo de razonamiento deductivo basado en medidas de similitud. Hemos propuesto la concreción de estas redes para Twitter, usando *relaciones profundas* que representan la *similitud de ideas con respecto a una temática*. Para obtener una función que nos permita estimar dicha relación profunda, hemos usado el concepto de *grafo de co-retweets*. Este grafo se usa para propagar etiquetas de usuarios conocidos a otros de los que no tenemos información, con el objetivo de facilitar el etiquetado de una base de datos. Como el proceso es iterativo y puede involucrar oráculos humanos, hemos presentado una metodología de etiquetado asistido que hemos probado meticulosamente durante la producción de 4 *datasets* de *tweets*. Los resultados muestran que la herramienta es capaz de reducir el esfuerzo inicial en dos órdenes de magnitud, con una precisión superior al 80 % en todos los escenarios evaluados.

Las capacidades de nuestra herramienta no acaban ahí. Nutcracker es capaz de usar modelos entrenados para predecir etiquetas o determinar la relevancia del sujeto (detección). La herramienta incluye un módulo de rastreo que usa el grafo de *co-retweets* para ejecutar una expansión preliminar de la etiqueta y únicamente recuperar contenido de aquellos usuarios que parezcan relevantes, no solo de manera instantánea, sino que planifica ciclos de recuperación de información en un tiempo preestablecido. Por consiguiente, es capaz de monitorizar grupos de usuarios relevantes. Finalmente, es posible etiquetar el contenido, usando tanto modelos entrenados como oráculos humanos, para obtener bases de datos etiquetadas que permitan el análisis de los datos.

En términos de *mejora de la comprensibilidad*, hemos podido comprobar que los modelos interpretables actuales tienen tal complejidad que entender qué hacen y por qué lo hacen se convierte en una taréa titánica. Ya que los modelos de aprendizaje computacional buscan patrones en el espacio de búsqueda que sean buenos predictores de la clase, hemos presentado las *expresiones diferenciadoras* como un nuevo tipo de característica. Dichas expresiones son secuencias de palabras sesgadas hacia una clase y, consecuentemente, es posible utilizar una menor cantidad de ellas sin perder información. Como el espacio de búsqueda se reduce, la complejidad del modelo es menor. Los resultados muestran que es posible obtener modelos un orden de magnitud menos complejos sin perder rendimiento durante la clasificación.

Para mejorar el análisis agregado, hemos ejecutado un estudio comparativo entre varios cuantificadores y datos de distinta naturaleza. PCC y PACC han sido los más competitivos, ganando PCC en experimentos donde hay presente características con diferentes distribuciones de probabilidad y PACC cuando la variabilidad se presenta en el las distribuciones de las clases. Hemos propuesto un cuantificador basado en similitud, que intentaba ajustar la suma probabilística de las predicciones utilizando medidas de distancia entre instancias. Sin embargo, nuestros resultados no han mejorado a los *baselines*, lo cual sugiere la necesidad de nuevas aproximaciones y abre una nueva línea de investigación.

Cuando agregamos datos usando ventanas temporales, PCC arroja los mejores resultados. Esto era esperable teniendo en cuenta que los datos usados son de un debate político, donde la temática varía en función al tiempo, por lo que se encuentran diferentes características de entrada entre ventanas alejadas en el tiempo. Hemos probado una aproximación que consistía en predecir los fallos en clasificación (*razón de falsos positivos* y *verdaderos positivos*) en función a la ventana, con el objetivo de ajustar el recuento a las particularidades de la misma. Sin embargo, una vez más, nuestros resultados no han sido suficientes para mejorar a las técnicas existentes. Esto abre la puerta a una nueva línea de investigación que se puede abordar adaptando técnicas de otras áreas, como la *propagación de características*.

Por último, hemos producido 4 bases de datos de diferentes temáticas e idiomas, etiquetadas utilizando nuestra herramienta y con la ayuda de colaboradores expertos. Hemos realizado un análisis exploratorio de uno de ellos, en términos de distribución de características, correlaciones entre las mismas y patrones frecuentes. También hemos desarrollado un estudio longitudinal del *sentiment* durante el debate de las Elecciones Generales de 2019, y hemos establecido las bases para integrar dicha funcionalidad en la herramienta.

Los resultados de esta tesis son útiles para investigadores de diferentes disciplinas, como las ciencias políticas, la lingüística computacional y la ciberseguridad. Prueba de ello es que ya están siendo usados por politólogos y lingüistas de nuestro equipo, y que ha llamado la atención de la Guardia Civil, por su utilidad y potencial de cara a las investigaciones en ciberseguridad. Las técnicas propuestas, y las metodologías y herramientas derivadas, constituyen una solución interpretable para recuperar, etiquetar y analizar datos de redes sociales con un esfuerzo reducido.

6.1 Aportes

Las contribuciones de las técnicas y metodologías presentadas en este trabajo son múltiples. A continuación, resumimos las principales:

- Hemos presentado las *Similarity Semantic Networks*, que son capaces de representar conocimiento y habilitar una nueva forma de razonamiento deductivo basado en medidas de similitud.

- Hemos desarollado una metodología interpretable, construida a partir de una concreción de *Similarity Semantic Network*, con el objetivo de producir datasets etiquetados a través de la propagación de características. Esta metodología es capaz de reducir significativamente el esfuerzo requerido para producir una base de datos manteniendo una precisión superior al 80 %.

- Hemos presentado el *grado de co-retweets*, como una concreción de una *Similarity Semantic Network*, utilizando una *relación profunda* que aproxima el valor de la relación de alto nivel *compartir ideas*.

- Hemos implementado una herramienta como prueba de concepto que es capaz de detectar, rastrear y monitorizar grupos de usuarios relevantes. También es capaz de aplicar la metodología mencionada anteriormente para obtener bases de datos etiquetadas con un menor esfuerzo.

- Hemos aportado un análisis comparativo de cómo de comprensibles son los modelos interpretables actuales, y hemos establecido una complejidad idónea basada en las capacidades de la memoria de trabajo humana.

- Hemos presentado un nuevo mecanismo de ponderación de palabras, CF-ICF, que es capaz de modelar la utilidad, como predictores, de características individuales.

- Hemos presentado el concepto de *expresión* y *expresión diferenciadora*, como una secuencia de palabras que actúan como predictor de una clase. Estas características reducen el espacio de búsqueda y producen modelos igual de precisos pero menos complejos y, por consiguiente, más comprensibles.

- Hemos estudiado los diferentes tipos de sesgos que se pueden presentar en datos recogidos de redes sociales, y los hemos clasificado atendiendo a la naturaleza de dicho sesgo.

- Hemos revisado las propuestas de otros autores para lidiar con la estimación de la prevalencia de las clases en muestras de datos de redes sociales y, en particular, en Twitter. Hemos llevado a cabo un estudio comparativo para determinar cuáles de los cuantificadores actuales se ajustan mejor a los diferentes sesgos en los conjuntos de validación.

- Hemos determinado que las propuestas más directas para integrar medidas de similitud espaciales y temporales no son suficientemente buenas para mejorar a las técnicas existentes, abriendo así una nueva línea de investigación.

6.2 Trabajos futuros

Nuestra investigación abre varias líneas de trabajo futuras. En esta sección, resumimos las más importantes.

En términos de *reducción del esfuerzo* hay mucho terreno que cubrir. Las *Similarity Semantic Networks* son capaces de representar distintos conceptos usando más de una relación de alto nivel. La expansión de propiedades puede beneficiarse de un grafo multicapa, que permita así la propagación de características de distinta naturaleza utilizando relaciones compatibles e ignorando el resto. Cabe esperar que dicha mejora no se refleje sólo en el alcance de la expansión sino también en la precisión de las deducciones.

Con respecto a la comprensibilidad de los modelos, las *expresiones diferenciadoras* pueden beneficiarse del uso de *simsets* y/o *embeddings*, tanto para las palabras como para el contexto. El algoritmo, tal y como lo hemos presentado, no es capaz de trabajar con *embeddings* y la interpretabilidad se perdería en tal caso. Sin embargo, sería posible el uso de sinónimos dentro de las expresiones para mejorar la relevancia estadística de las mismas, así como su precisión. Desafortunadamente, esto también aumentaría el coste computacional que se requiere para extraer y seleccionar dichas características. Así, se vuelve necesaria la ejecución de un estudio experimental para determinar el mejor balance entre *utilidad* y *coste computacional*.

Los métodos de cuantificación pueden beneficiarse del uso de medidas de similitud espaciales y temporales. La estimación de prevalencia de clases se ajusta teniendo en cuenta los fallos de clasificación, pero éstos varían dependiendo de la muestra. Establecer y delimitar regiones del espacio de búsqueda con fallos de clasificación similares pueden mejorar el ajuste y, por consiguiente, arrojar mejores estimaciones.

Los usos potenciales y aplicaciones de nuestras técnicas y herramientas no están totalmente acotados, y pueden incluir la detección de perfiles relacionados con un evento (como aquellos involucrados en un desastre natural), la detección de perfiles que cumplan una serie de características (como aquellos que sean propensos a facilitar la propagación de noticias falsas), la delimitación de grupos de usuarios con ciertas características (como aquellos que sean vulnerables al discurso de radicalización), la estimación del tamaño de los grupos de interés, la caracterización de la evolución del *sentiment* durante un evento, la evaluación del efecto de las estrategias políticas en la opinión pública y, virtualmente, cualquier caso de uso que pueda beneficiarse del uso de datos de redes sociales. La reducción del esfuerzo que resulta de la aplicación de nuestras técnicas abre puertas a escenarios de investigación de bajo presupuesto, y la interpretabilidad de las mismas habilita el uso en situaciones que puedan incurrir en un impacto social.

Bibliography

[1]Jacob Amedie. “The Impact of Social Media on Society”. In: *Pop Culture Intersections* (Sept. 2015) (cit. on pp. 1, 11).

[2]*Digital 2022: Global Overview Report*. en-GB. 2022 (cit. on pp. 1, 11).

[3]*Domo Resource - Data Never Sleeps 9.0*. en. 2021 (cit. on pp. 1, 11).

[4]J.Y. Ng, W. Abdelkader, and C. Lokker. “Tracking discussions of complementary, alternative, and integrative medicine in the context of the COVID-19 pandemic: a month-by-month sentiment analysis of Twitter data”. English. In: *BMC Complementary Medicine and Therapies* 22.1 (2022) (cit. on pp. 1, 12).

[5]P. Samanta, P. Kumar, S. Dutta, M. Chatterjee, and D. Sarkar. “Depression Detection from Twitter Data Using Two Level Multi-modal Feature Extraction”. English. In: *Lecture Notes on Data Engineering and Communications Technologies* 137 (2023), pp. 451–465 (cit. on pp. 2, 12).

[6]I. Dans-Álvarez-de-Sotomayor, P.C. Muñoz-Carril, and M. González-Sanmamed. “Uses and abuses of Social Media by Spanish Secondary Education students”. Spanish. In: *Revista Electronica Educare* 26.3 (2022), pp. 1–16 (cit. on pp. 2, 12).

[7]G. Toh, E. Pearce, J. Vines, et al. “Digital interventions for subjective and objective social isolation among individuals with mental health conditions: a scoping review”. English. In: *BMC Psychiatry* 22.1 (2022) (cit. on pp. 2, 12).

[8]F. Aliverdi, H. Farajidana, Z.M. Tourzani, et al. “Social networks and internet emotional relationships on mental health and quality of life in students: structural equation modelling”. English. In: *BMC Psychiatry* 22.1 (2022) (cit. on pp. 2, 12).

[9]S.L. Lin. “The “loneliness epidemic”, intersecting risk factors and relations to mental health help-seeking: A population-based study during COVID-19 lockdown in Canada”. English. In: *Journal of Affective Disorders* 320 (2023), pp. 7–17 (cit. on pp. 2, 12).

[10]Sandra González-Bailón, Ning Wang, Alejandro Rivero, Javier Borge-Holthoefer, and Yamir Moreno. “Assessing the bias in samples of large online networks”. en. In: *Social Networks* 38 (July 2014), pp. 16–27 (cit. on pp. 2, 12, 114).

[11]S.S. Gomasta, A. Dhali, M.M. Anwar, and I.H. Sarker. “Query-oriented topical influential users detection for top-k trending topics in twitter”. English. In: *Applied Intelligence* 52.12 (2022), pp. 13415–13434 (cit. on pp. 2, 12).

[12]M. El Tantawi, A. Al-Ansari, A. AlSubaie, et al. “Reach of messages in a dental twitter network: Cohort study examining user popularity, communication pattern, and network structure”. English. In: *Journal of Medical Internet Research* 20.9 (2018) (cit. on pp. 2, 12).

[13]Y. Kamiko, M. Yoshida, H. Ohashi, and F. Toriumi. “Uncovering information flow among users by time-series retweet data: Who is a friend of whom on Twitter?” English. In: 2016, pp. 2500–2504 (cit. on pp. 2, 12).

[14]E.U. Haq, T. Braud, Y.D. Kwon, and P. Hui. “Enemy at the Gate: Evolution of Twitter User’s Polarization during National Crisis”. English. In: 2020, pp. 212–216 (cit. on pp. 2, 12).

[15]P. Darius. “Who polarizes Twitter? Ideological polarization, partisan groups and strategic networked campaigning on Twitter during the 2017 and 2021 German Federal elections ’Bundestagswahlen’”. English. In: *Social Network Analysis and Mining* 12.1 (2022) (cit. on pp. 2, 12).

[16]T. Cicchini, S.M. del Pozo, E. Tagliazucchi, and P. Balenzuela. “News sharing on Twitter reveals emergent fragmentation of media agenda and persistent polarization”. English. In: *EPJ Data Science* 11.1 (2022) (cit. on pp. 2, 12).

[17]A. Aswathy, R. Prabha, L.S. Gopal, D. Pullarkatt, and M.V. Ramesh. “An efficient twitter data collection and analytics framework for effective disaster management”. English. In: 2022 (cit. on pp. 2, 12).

[18]S. Splendiani and A. Capriello. “Crisis communication, social media and natural disasters – the use of Twitter by local governments during the 2016 Italian earthquake”. English. In: *Corporate Communications* 27.3 (2022), pp. 509–526 (cit. on pp. 2, 12).

[19]F. Sufi. “A decision support system for extracting artificial intelligence-driven insights from live twitter feeds on natural disasters”. English. In: *Decision Analytics Journal* 5 (2022) (cit. on pp. 2, 12).

[20]S.K. Sharma, M. Daga, and B. Gemini. “Twitter Sentiment Analysis for Brand Reputation of Smart Phone Companies in India”. English. In: *Lecture Notes in Electrical Engineering* 605 (2020). ISBN: 9783030305765, pp. 841–852 (cit. on pp. 2, 12).

[21]H. Shirdastian, M. Laroche, and M.-O. Richard. “Using big data analytics to study brand authenticity sentiments: The case of Starbucks on Twitter”. English. In: *International Journal of Information Management* 48 (2019), pp. 291–307 (cit. on pp. 2, 12).

[22]D. Arora, K.F. Li, and S.W. Neville. “Consumers’ sentiment analysis of popular phone brands and operating system preference using twitter data: A feasibility study”. English. In: vol. 2015-April. ISSN: 1550-445X. 2015, pp. 680–686 (cit. on pp. 2, 12).

[23] J.K. Harris, J.B. Hawkins, L. Nguyen, et al. “Using Twitter to Identify and Respond to Food Poisoning: The Food Safety STL Project”. English. In: *Journal of Public Health Management and Practice* 23.6 (2017), pp. 577–580 (cit. on pp. 2, 12).

[24] P. Darius and F. Stephany. ““Hashjacking” the Debate: Polarisation Strategies of Germany’s Political Far-Right on Twitter”. English. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11864 LNCS (2019). ISBN: 9783030349707, pp. 298–308 (cit. on pp. 2, 12).

[25] M.M. Chiu, C.H. Park, H. Lee, Y.W. Oh, and J.-N. Kim. “Election Fraud and Misinformation on Twitter: Author, Cluster, and Message Antecedents”. English. In: *Media and Communication* 10.2 (2022), pp. 66–80 (cit. on pp. 2, 12).

[26] T. Pattewar and D. Jain. “Stock prediction analysis by customers opinion in Twitter data using an optimized intelligent model”. English. In: *Social Network Analysis and Mining* 12.1 (2022) (cit. on pp. 2, 12).

[27] J. M. Berger and Jonathon Morgan. *The ISIS Twitter census: Defining and describing the population of ISIS supporters on Twitter*. en-US. 2015 (cit. on pp. 2, 13).

[28] Miguel-Ángel Benítez Castro and Encarnación Hidalgo Tenorio. *Rethinking Martin & White’s affect taxonomy*. Mackenzie. JL y Alba Juez, L.(eds.). Emotion in discourse. Amsterdam/Philadelphia: John Benjamins, 2019 (cit. on pp. 3, 13, 72).

[29] Encarnacion Hidalgo-Tenorio and Miguel Angel Benitez-Castro. “The Language of Evaluation in the Narratives by the Magdalene Laundries Survivors: The Discourse of Female Victimhood”. In: *Applied Linguistics* 42.2 (Apr. 2021), pp. 315–341 (cit. on pp. 3, 13).

[30] Encarnación Hidalgo-Tenorio and Miguel-Ángel Benítez-Castro. “Trump’s populist discourse and affective politics, or on how to move ‘the People’ through emotion”. In: *Globalisation, Societies and Education* 20.2 (Mar. 2022), pp. 86–109 (cit. on pp. 3, 13).

[31] Miguel-Ángel Benítez-Castro and Encarnación Hidalgo-Tenorio. ““I Am Proud to Be a Traitor”: The emotion/opinion interplay in jihadist magazines:” en. In: *Pragmatics and Society* 13.3 (July 2022). Publisher: John Benjamins Publishing Company, pp. 501–531 (cit. on pp. 3, 13).

[32] Elena Block and Ralph Negrine. “The Populist Communication Style: Toward a Critical Framework”. en. In: *International Journal of Communication* 11.0 (Jan. 2017). Number: 0, p. 20 (cit. on pp. 3, 13).

[33] Ignacio-Jesús Serrano-Contreras, Javier García-Marín, and Óscar G. Luengo. “Measuring online political dialogue: does polarization trigger more deliberation?” en. In: *Media and Communication* 8.4 (2020), pp. 63–72 (cit. on pp. 3, 13).

[34] Óscar García and Javier García. “Spanish TV portrayal of terrorism during the 2008 campaign: an example of polarised pluralism?” In: *Przegląd europejski* 3 (29 (2020). Publisher: Uniwersytet Warszawski. Wydawnictwa Uniwersytetu Warszawskiego, pp. 49–65 (cit. on pp. 3, 13).

[35] Susana Salgado, Óscar G. Luengo, Stylianos Papathanassopoulos, Jane Suiter, and Agnieszka Stępińska. “Crisis and populism: a comparative study of populist and non-populist candidates and rhetoric in the news media coverage of election campaigns”. In: *European Politics and Society* 0.0 (Mar. 2021). Publisher: Routledge _eprint: https://doi.org/10.1080/23745118.2021.1896882, pp. 1–16 (cit. on pp. 3, 13).

[36] Óscar G. Luengo and Belén Fernández-García. “Digital (and Traditional) Media Usage in Spanish Electoral Campaigns”. en. In: *Digitalization of Democratic Processes in Europe: Southern and Central Europe in Comparative Perspective*. Ed. by Magdalena Musiał-Karg and Óscar G. Luengo. Studies in Digital Politics and Governance. Cham: Springer International Publishing, 2021, pp. 43–56 (cit. on pp. 3, 13).

[37] Ignacio-Jesús Serrano-Contreras, Javier García-Marín, Óscar G. Luengo, et al. “Coberturas mediáticas, polarización y reformas educativas en España”. In: *Revista de ciencia política (Santiago)* 41.3 (Dec. 2021). Publisher: Pontificia Universidad Católica de Chile. Instituto de Ciencia Política, pp. 497–514 (cit. on pp. 3, 13).

[38] Javier García-Marín and Óscar G. Luengo. “From image to function: Automated analysis of online jihadi videos”. en. In: *Pragmatics and Society* 13.3 (July 2022). Publisher: John Benjamins, pp. 383–403 (cit. on pp. 3, 13).

[39] José Manuel Moreno Mercado, Javier García Marín, and Óscar García Luengo. “Conflictos armados y la construcción de narrativas a través de Twitter. El caso de la guerra entre Armenia y Azerbaiyán”. spa. In: (July 2022). Accepted: 2022-09-23T07:36:35Z Publisher: Asociación Española de Ciencia Política y de la Administración (cit. on pp. 3, 13).

[40] FAT/ML. *Principles for Accountable Algorithms and a Social Impact Statement for Algorithms*. 2019 (cit. on pp. 4, 15, 84, 90).

[41] Cynthia Rudin. “Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead”. en. In: *Nature Machine Intelligence* 1.5 (May 2019). Number: 5 Publisher: Nature Publishing Group, pp. 206–215 (cit. on pp. 5, 15, 85, 90, 107).

[42] Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. *The Dangers of Post-hoc Interpretability: Unjustified Counterfactual Explanations*. arXiv:1907.09294 [cs, stat]. July 2019 (cit. on pp. 5, 15).

[43] Sofie Goethals, David Martens, and Theodoros Evgeniou. “The non-linear nature of the cost of comprehensibility”. In: *Journal of Big Data* 9.1 (Mar. 2022), p. 30 (cit. on pp. 5, 15).

[44] Adrien Bibal and Benoît Frénay. *Interpretability of Machine Learning Models and Representations: an Introduction*. Apr. 2016 (cit. on pp. 5, 15).

[45] Pablo González, Alberto Castaño, Nitesh V. Chawla, and Juan José Del Coz. “A Review on Quantification Learning”. en. In: *ACM Computing Surveys* 50.5 (Nov. 2017), pp. 1–40 (cit. on pp. 6, 17, 118, 120).

[46] Wei Gao and Fabrizio Sebastiani. “From classification to quantification in tweet sentiment analysis”. en. In: *Social Network Analysis and Mining* 6.1 (Apr. 2016), p. 19 (cit. on pp. 7, 18, 110, 117, 122, 147).

[47] Michael W. Berry, Azlinah Mohamed, and Bee Wah Yap, eds. *Supervised and Unsupervised Learning for Data Science*. en. Unsupervised and Semi-Supervised Learning. Cham: Springer International Publishing, 2020 (cit. on p. 21).

[48] *Datasets Over Algorithms*. en-US (cit. on p. 24).

[49] Zhi-Hua Zhou. “A brief introduction to weakly supervised learning”. In: *National Science Review* 5.1 (Jan. 2018), pp. 44–53 (cit. on p. 25).

[50] Gerard Andrews. *What is synthetic data?* en-US. June 2021 (cit. on p. 25).

[51] Maayan Frid-Adar, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. “Synthetic data augmentation using GAN for improved liver lesion classification”. In: *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*. ISSN: 1945-8452. Apr. 2018, pp. 289–293 (cit. on p. 25).

[52] Sergey I. Nikolenko. “Synthetic Data for Deep Learning”. In: *arXiv:1909.11512 [cs]* (Sept. 2019) (cit. on p. 25).

[53] Jiaqi Guan, Runzhe Li, Sheng Yu, and Xuegong Zhang. “A Method for Generating Synthetic Electronic Medical Record Text”. In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 18.1 (Jan. 2021), pp. 173–182 (cit. on p. 25).

[54] Alec Radford, Jeffrey Wu, Rewon Child, et al. “Language Models are Unsupervised Multitask Learners”. en. In: *ArXiV* (2019), p. 24 (cit. on p. 25).

[55] *OpenAI has released the largest version yet of its fake-news-spewing AI*. en (cit. on p. 25).

[56] Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. “Data Programming: Creating Large Training Sets, Quickly”. en. In: *ArXiV* (2016), p. 9 (cit. on p. 26).

[57] Alexander Ratner, Christopher De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. "Data Programming: Creating Large Training Sets, Quickly". In: *arXiv:1605.07723 [cs, stat]* (Jan. 2017) (cit. on pp. 26, 27).

[58] Geon Heo, Yuji Roh, Seonghyeon Hwang, Dayun Lee, and Steven Euijong Whang. "Inspector gadget: a data programming-based labeling system for industrial images". In: *Proceedings of the VLDB Endowment* 14.1 (Sept. 2020), pp. 28–36 (cit. on p. 26).

[59] Neil Mallinar, Abhishek Shah, Tin Kam Ho, Rajendra Ugrani, and Ayush Gupta. "Iterative Data Programming for Expanding Text Classification Corpora". In: *arXiv:2002.01412 [cs]* (Feb. 2020) (cit. on p. 26).

[60] Shakshi Sharma and Rajesh Sharma. "Identifying Possible Rumor Spreaders on Twitter: A Weak Supervised Learning Approach". In: *2021 International Joint Conference on Neural Networks (IJCNN)*. ISSN: 2161-4407. July 2021, pp. 1–8 (cit. on p. 27).

[61] B. Settles. "Active Learning Literature Survey". en. In: *undefined* (2009) (cit. on p. 27).

[62] Nils Haldenwang, Katrin Ihler, Julian Kniephoff, and Oliver Vornberger. "A Comparative Study of Uncertainty Based Active Learning Strategies for General Purpose Twitter Sentiment Analysis with Deep Neural Networks". en. In: *Language Technologies for the Challenges of the Digital Age*. Ed. by Georg Rehm and Thierry Declerck. Vol. 10713. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2018, pp. 208–215 (cit. on p. 27).

[63] Stefan Helmstetter and Heiko Paulheim. "Collecting a Large Scale Dataset for Classifying Fake News Tweets Using Weak Supervision". en. In: *Future Internet* 13.5 (May 2021). Number: 5 Publisher: Multidisciplinary Digital Publishing Institute, p. 114 (cit. on p. 27).

[64] D. Pohl, A. Bouchachia, and H. Hellwagner. "Active Online Learning for Social Media Analysis to Support Crisis Management". English. In: *IEEE Transactions on Knowledge and Data Engineering* 32.8 (2020), pp. 1445–1458 (cit. on p. 28).

[65] Rob Chew, Michael Wenger, Caroline Kery, et al. "SMART: An Open Source Data Labeling Platform for Supervised Learning". In: *Journal of Machine Learning Research* 20.82 (2019), pp. 1–5 (cit. on p. 28).

[66] Jie Yang, Yue Zhang, Linwei Li, and Xingxuan Li. "YEDDA: A Lightweight Collaborative Text Span Annotation Tool". In: *arXiv:1711.03759 [cs]* (May 2018) (cit. on p. 28).

[67] Hiroki Nakayama, Takahiro Kubo, Junya Kamura, Yasufumi Taniguchi, and Xu Liang. *doccano: Text Annotation Tool for Human*. Software available from https://github.com/doccano/doccano. 2018 (cit. on p. 29).

[68] Atta Rahman. "Knowledge Representation: A Semantic Network Approach". In: June 2016 (cit. on p. 31).

[69] Liang Guo, Fu Yan, Tian Li, Tao Yang, and Yuqian Lu. "An automatic method for constructing machining process knowledge base from knowledge graph". en. In: *Robotics and Computer-Integrated Manufacturing* 73 (Feb. 2022), p. 102222 (cit. on p. 31).

[70] Jamal Alhiyafi, Atta-ur-Rahman, Fahd Abdulsalam Alhaidari, and Mohammed Aftab Khan. "Automatic Text Categorization Using Fuzzy Semantic Network". en. In: *Proceedings of the 1st International Conference on Smart Innovation, Ergonomics and Applied Human Factors (SEAHF)*. Ed. by César Benavente-Peces, Sami Ben Slama, and Bassam Zafar. Smart Innovation, Systems and Technologies. Cham: Springer International Publishing, 2019, pp. 24–34 (cit. on p. 31).

[71] Dora-Luz Flores, Antonio Rodríguez-Díaz, Juan R. Castro, and Carelia Gaxiola. "TA-Fuzzy Semantic Networks for Interaction Representation in Social Simulation". en. In: *Evolutionary Design of Intelligent Systems in Modeling, Simulation and Control*. Ed. by Oscar Castillo, Witold Pedrycz, and Janusz Kacprzyk. Studies in Computational Intelligence. Berlin, Heidelberg: Springer, 2009, pp. 213–225 (cit. on p. 31).

[72] Mohamed Nazih Omri and Noureddine Chouigui. "Measure of Similarity Between Fuzzy Concepts for Identification of Fuzzy User's Requests in Fuzzy Semantic Networks." In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 9 (Dec. 2001), pp. 743–748 (cit. on p. 32).

[73] Danah M Boyd and Nicole B. Ellison. "Social Network Sites: Definition, History, and Scholarship". en. In: *Journal of Computer-Mediated Communication* 13.1 (Oct. 2007), pp. 210–230 (cit. on p. 34).

[74] Bishwajit Purkaystha, Tapos Datta, Md. Saiful Islam, and Marium-E-Jannat. "Rating prediction for recommendation: Constructing user profiles and item characteristics using backpropagation". In: *Applied Soft Computing* 75 (Feb. 2019), pp. 310–322 (cit. on p. 34).

[75] Manuel Francisco and Juan L. Castro. "A fuzzy model to enhance user profiles in microblogging sites using deep relations". en. In: *Fuzzy Sets and Systems*. Fuzzy Measures, Integrals and Quantification in Artificial Intelligence Problems – An Homage to Prof. Miguel Delgado 401 (Dec. 2020), pp. 133–149 (cit. on pp. 34, 45).

[76] Xin Guo, Yang Xiang, Qian Chen, Zhenhua Huang, and Yongtao Hao. "LDA-based online topic detection using tensor factorization". In: *Journal of Information Science* 39.4 (Aug. 1, 2013), pp. 459–469 (cit. on p. 36).

[77] SayyadiHassan and RaschidLouiqa. "A Graph Analytical Approach for Topic Detection". In: *ACM Transactions on Internet Technology (TOIT)* (Dec. 1, 2013) (cit. on p. 36).

[78] Ahmed Alsanad. “Arabic Topic Detection Using Discriminative Multi Nominal Naïve Bayes and Frequency Transforms”. In: *Proceedings of the 2018 International Conference on Signal Processing and Machine Learning*. SPML ’18. Shanghai, China: Association for Computing Machinery, 2018, pp. 17–21 (cit. on p. 36).

[79] Jamilah Rabeh Alharbi and Wadee S. Alhalabi. “Hybrid Approach for Sentiment Analysis of Twitter Posts Using a Dictionary-based Approach and Fuzzy Logic Methods: Study Case on Cloud Service Providers”. In: *International Journal on Semantic Web and Information Systems (IJSWIS)* 16.1 (2020), pp. 116–145 (cit. on p. 36).

[80] Jamilu Awwalu, Azuraliza Abu Bakar, and Mohd Ridzwan Yaakub. “Hybrid N-gram model using Naïve Bayes for classification of political sentiments on Twitter”. In: *Neural Computing and Applications* 31.12 (Dec. 1, 2019), pp. 9207–9220 (cit. on p. 36).

[81] Monika Arora and Vineet Kansal. “Character level embedding with deep convolutional neural network for text normalization of unstructured data for Twitter sentiment analysis”. In: *Social Network Analysis and Mining* 9.1 (2020), p. 12 (cit. on p. 36).

[82] A. Irsyad and N.A. Rakhmawati. “Community detection in twitter based on tweets similarities in indonesian using cosine similarity and louvain algorithms”. English. In: *Register: Jurnal Ilmiah Teknologi Sistem Informasi* 6.1 (2020), pp. 22–31 (cit. on p. 45).

[83] Wendel Silva, Adamo Santana, Fabio Lobato, and Marcia Pinheiro. “A methodology for community detection in Twitter”. In: *Proceedings of the International Conference on Web Intelligence*. WI ’17. New York, NY, USA: Association for Computing Machinery, Aug. 2017, pp. 1006–1009 (cit. on p. 45).

[84] Ron Artstein and Massimo Poesio. “Inter-Coder Agreement for Computational Linguistics”. en. In: *Computational Linguistics* 34.4 (Dec. 2008), pp. 555–596 (cit. on p. 74).

[85] Klaus Krippendorff. “On the Reliability of Unitizing Continuous Data”. In: *Sociological Methodology* 25 (1995). Publisher: [American Sociological Association, Wiley, Sage Publications, Inc.], pp. 47–76 (cit. on p. 74).

[86] Klaus Krippendorff. *Content analysis: an introduction to its methodology*. Fourth Edition. Los Angeles: SAGE, 2018 (cit. on p. 74).

[87] J. Richard Landis and Gary G. Koch. “The Measurement of Observer Agreement for Categorical Data”. In: *Biometrics* 33.1 (1977). Publisher: [Wiley, International Biometric Society], pp. 159–174 (cit. on p. 75).

[88] F. Pedregosa, G. Varoquaux, A. Gramfort, et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830 (cit. on pp. 75, 93).

[89] Tivadar Danka and Peter Horvath. “modAL: A modular active learning framework for Python”. In: (). available on arXiv at `https://arxiv.org/abs/1805.00979` (cit. on p. 75).

[90] Amine Abdaoui, Camille Pradel, and Grégoire Sigel. “Load What You Need: Smaller Versions of Mutlilingual BERT”. In: *SustaiNLP / EMNLP*. 2020 (cit. on p. 75).

[91] Tarleton Gillespie. *Custodians of the Internet: Platforms, Content Moderation, and the Hidden Decisions That Shape Social Media*. en. Google-Books-ID: cO-JgDwAAQBAJ. Yale University Press, June 2018 (cit. on pp. 76, 80).

[92] Megan Garcia. “Racist in the Machine: The Disturbing Implications of Algorithmic Bias”. In: *World Policy Journal* 33.4 (Dec. 2016), pp. 111–117 (cit. on p. 83).

[93] *Google apologizes after its Vision AI produced racist results*. en (cit. on p. 83).

[94] *In 2016, Microsoft’s Racist Chatbot Revealed the Dangers of Online Conversation - IEEE Spectrum* (cit. on p. 83).

[95] Avery Phillips. *The Moral Dilemma of Algorithmic Censorship*. Aug. 2018 (cit. on p. 84).

[96] Hanff, Alexander. *The Cold World of Algorithmic Censorship - Alexander Hanff - Medium*. 2018 (cit. on p. 84).

[97] M. O’Dair and A. Fry. “Beyond the black box in music streaming: the impact of recommendation systems upon artists”. In: *Popular Communication* (2019) (cit. on p. 84).

[98] Z. Zhao, M. Gao, J. Yu, et al. “Impact of the Important Users on Social Recommendation System”. In: *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST* 252 (2018), pp. 425–434 (cit. on p. 84).

[99] Dokyun Lee and K. Hosanagar. “Impact of Recommender Systems on Sales Volume and Diversity”. In: *ICIS*. 2014 (cit. on p. 84).

[100] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. “Introduction to Data Mining”. Inglés. In: Republic and Canton of Geneva, Switzerland: Pearson, 2005 (cit. on p. 84).

[101] Christoph Molnar. *Interpretable Machine Learning*. en. Leanpub, Feb. 2018 (cit. on pp. 84, 92, 104).

[102] Finale Doshi-Velez and Been Kim. “Towards A Rigorous Science of Interpretable Machine Learning”. In: *arXiv:1702.08608 [cs, stat]* (Feb. 2017). arXiv: 1702.08608 (cit. on pp. 84, 90).

[103] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Inglés. Edición: 01. Boston: Pearson, May 2005 (cit. on p. 84).

[104]Matthew E. Peters, Mark Neumann, Mohit Iyyer, et al. “Deep contextualized word representations”. In: *arXiv:1802.05365 [cs]* (Feb. 2018). arXiv: 1802.05365 (cit. on pp. 84, 89, 91).

[105]Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *arXiv:1810.04805 [cs]* (Oct. 2018). arXiv: 1810.04805 (cit. on pp. 84, 89, 91).

[106]A. Moreo, M. Navarro, J. L. Castro, and J. M. Zurita. “A high-performance FAQ retrieval method using minimal differentiator expressions”. In: *Knowledge-Based Systems* 36 (Dec. 2012), pp. 9–20 (cit. on pp. 85, 91).

[107]Carlos Periñán-Pascual and Francisco Arcas-Túnez. “Detecting environmentally-related problems on Twitter”. In: *Biosystems Engineering*. Intelligent Systems for Environmental Applications 177 (Jan. 2019), pp. 31–48 (cit. on p. 85).

[108]A. Castellanos, J. Cigarrn, and A. Garca-Serrano. “Formal Concept Analysis for Topic Detection”. In: *Inf. Syst.* 66.C (June 2017), pp. 24–42 (cit. on p. 85).

[109]L. Nassar, R. Ibrahim, and F. Karray. “Enhancing Topic Detection in Twitter Using the Crowdsourcing Process”. In: *2016 International Conference on Collaboration Technologies and Systems (CTS)*. Oct. 2016, pp. 196–203 (cit. on p. 85).

[110]W. Xie, F. Zhu, J. Jiang, E. Lim, and K. Wang. “TopicSketch: Real-time Bursty Topic Detection from Twitter”. en. In: *IEEE Transactions on Knowledge and Data Engineering* 28.8 (2016), pp. 2216–2229 (cit. on p. 85).

[111]A. Alsaig, A. Alsaig, M. Alsadun, and S. Barghi. “Context based algorithm for social influence measurement on Twitter”. In: *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST* 266 (2019), pp. 136–149 (cit. on p. 85).

[112]Ziyaad Qasem, Marc Jansen, Tobias Hecking, and H. Ulrich Hoppe. “Detection of strong attractors in social media networks”. en. In: *Computational Social Networks* 3.1 (Dec. 2016), p. 11 (cit. on p. 85).

[113]Helge Jorgens, Nina Kolleck, and Barbara Saerbeck. “Exploring the hidden influence of international treaty secretariats: using social network analysis to analyse the Twitter debate on the Lima Work Programme on Gender”. In: *Journal of European Public Policy* 23.7 (Aug. 2016), pp. 979–998 (cit. on p. 85).

[114]Karthik Subbian, Charu Aggarwal, and Jaideep Srivastava. “Mining Influencers Using Information Flows in Social Streams”. In: *ACM Trans. Knowl. Discov. Data* 10.3 (Jan. 2016), 26:1–26:28 (cit. on p. 85).

[115]A.S.M. Alharbi and E. de Doncker. “Twitter sentiment analysis with a deep neural network: An enhanced approach using user behavioral information”. In: *Cognitive Systems Research* 54 (2019), pp. 50–61 (cit. on p. 85).

[116]Zou Xiaomei, Yang Jing, Zhang Jianpei, and Han Hongyu. “Microblog sentiment analysis with weak dependency connections”. In: *Knowledge-Based Systems* 142 (Feb. 2018), pp. 170–180 (cit. on p. 85).

[117]Sara Rosenthal, Noura Farra, and Preslav Nakov. “SemEval-2017 Task 4: Sentiment Analysis in Twitter”. In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vancouver, Canada: Association for Computational Linguistics, Aug. 2017, pp. 502–518 (cit. on p. 85).

[118]Hassan Saif, Yulan He, Miriam Fernandez, and Harith Alani. “Contextual semantics for sentiment analysis of Twitter”. In: *Information Processing & Management*. Emotion and Sentiment in Social and Expressive Media 52.1 (Jan. 2016), pp. 5–19 (cit. on p. 85).

[119]B. N. Supriya, Vish Kallimani, S. Prakash, and C. B. Akki. “Twitter Sentiment Analysis Using Binary Classification Technique”. en. In: *Nature of Computation and Communication*. Ed. by Phan Cong Vinh and Leonard Barolli. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Springer International Publishing, 2016, pp. 391–396 (cit. on p. 85).

[120]Qiudan Li, Zhipeng Jin, Can Wang, and Daniel Dajun Zeng. “Mining opinion summarizations using convolutional neural networks in Chinese microblogging systems”. In: *Knowledge-Based Systems* 107 (Sept. 2016), pp. 289–300 (cit. on p. 85).

[121]L. Gasco, C. Clavel, C. Asensio, and G. de Arcas. “Beyond sound level monitoring: Exploitation of social media to gather citizens subjective response to noise”. In: *Science of the Total Environment* 658 (2019), pp. 69–79 (cit. on p. 85).

[122]Shu Takahashi, Ayumu Sugiyama, and Youji Kohda. “A Method for Opinion Mining of Coffee Service Quality and Customer Value by Mining Twitter”. In: *Knowledge, Information and Creativity Support Systems*. Ed. by Susumu Kunifuji, George Angelos Papadopoulos, Andrzej M.J. Skulimowski, and Janusz Kacprzyk. Cham: Springer International Publishing, 2016, pp. 521–528 (cit. on p. 85).

[123]P. Kumar, T. Choudhury, S. Rawat, and S. Jayaraman. “Analysis of Various Machine Learning Algorithms for Enhanced Opinion Mining Using Twitter Data Streams”. In: *2016 International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE)*. Sept. 2016, pp. 265–270 (cit. on p. 85).

[124]Xueming Qian, Mingdi Li, Yayun Ren, and Shuhui Jiang. “Social media based event summarization by user–text–image co-clustering”. In: *Knowledge-Based Systems* 164 (Jan. 2019), pp. 107–121 (cit. on p. 85).

[125]Ignacio Arroyo-Fernández, Arturo Curiel, and Carlos-Francisco Méndez-Cruz. “Language features in extractive summarization: Humans Vs. Machines”. In: *Knowledge-Based Systems* 180 (Sept. 2019), pp. 1–11 (cit. on p. 85).

[126] Zellig S. Harris. “Distributional Structure”. In: *WORD* 10.2 (Aug. 1954), pp. 146–162 (cit. on p. 86).

[127] Karen Sparck Jones. “A statistical interpretation of term specificity and its application in retrieval”. In: *Journal of Documentation* 28.1 (Jan. 1, 1972), pp. 11–21 (cit. on p. 86).

[128] Hai-Tao Zheng, Zhe Wang, Wei Wang, et al. “Learning-based topic detection using multiple features”. English. In: *Concurrency and Computation-Practice & Experience* 30.15 (Aug. 2018). WOS:000438339700001, e4444 (cit. on pp. 86, 91).

[129] R.M. Green and J.W. Sheppard. “Comparing frequency- and style-based features for Twitter author identification”. In: *FLAIRS Conference*. FLAIRS 2013 - Proceedings of the 26th International Florida Artificial Intelligence Research Society Conference. 2013, pp. 64–69 (cit. on p. 86).

[130] Jiawei Han, Jian Pei, and Yiwen Yin. “Mining Frequent Patterns Without Candidate Generation”. In: *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. SIGMOD ’00. New York, NY, USA: ACM, 2000, pp. 1–12 (cit. on p. 86).

[131] Chenliang Li, Aixin Sun, and Anwitaman Datta. “Twevent: Segment-based Event Detection from Tweets”. In: *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. CIKM 12. New York, NY, USA: ACM, 2012, pp. 155–164 (cit. on p. 86).

[132] Hassan Sayyadi and Louiqa Raschid. “A Graph Analytical Approach for Topic Detection”. In: *ACM Trans. Internet Technol.* 13.2 (Dec. 2013), 4:1–4:23 (cit. on p. 86).

[133] Heyong Wang and Ming Hong. “Supervised Hebb rule based feature selection for text classification”. en. In: *Information Processing & Management* 56.1 (Jan. 2019), pp. 167–191 (cit. on p. 86).

[134] Bing Xue, Mengjie Zhang, and Will Browne. “Particle Swarm Optimization for Feature Selection in Classification: A Multi-Objective Approach”. In: *IEEE transactions on cybernetics* 43 (Dec. 2013), pp. 1656–1671 (cit. on p. 86).

[135] Hong Liang, Xiao Sun, Yunlei Sun, and Yuan Gao. “Text feature extraction based on deep learning: a review”. en. In: *EURASIP Journal on Wireless Communications and Networking* 2017.1 (Dec. 2017) (cit. on pp. 87, 92).

[136] Leszek Rutkowski, Ryszard Tadeusiewicz, Lofti A. Zadeh, and Jacek M. Zurada. *Artificial Intelligence and Soft Computing – ICAISC 2008: 9th International Conference Zakopane, Poland, June 22-26, 2008, Proceedings*. en. Google-Books-ID: cdBlypDgCK8C. Springer Science & Business Media, June 2008 (cit. on p. 87).

[137] Senthil Kumar B and Bhavitha Varma E. *A Different Type of Feature Selection Methods for Text Categorization on Imbalanced Data*. en. 2016 (cit. on p. 87).

[138] Zhaohui Zheng, Xiaoyun Wu, and Rohini Srihari. “Feature selection for text categorization on imbalanced data”. In: *ACM SIGKDD Explorations Newsletter* 6.1 (June 2004), pp. 80–89 (cit. on pp. 87, 88).

[139] George Forman. “An extensive empirical study of feature selection metrics for text classification [J]”. In: *Journal of Machine Learning Research - JMLR* 3 (Mar. 2003) (cit. on p. 87).

[140] Maria Fernanda Caropreso, Stan Matwin, and Fabrizio Sebastiani. *A Learner-Independent Evaluation of the Usefulness of Statistical Phrases for Automated Text Categorization*. en. 2001 (cit. on p. 87).

[141] Jianli Ding and Liyang Fu. “A Hybrid Feature Selection Algorithm Based on Information Gain and Sequential Forward Floating Search”. en. In: *Journal of Intelligent Computing* 9.3 (Sept. 2018), p. 93 (cit. on p. 87).

[142] Xuelian Deng, Yuqing Li, Jian Weng, and Jilian Zhang. “Feature selection for text classification: A review”. en. In: *Multimedia Tools and Applications* 78.3 (Feb. 2019), pp. 3797–3816 (cit. on p. 87).

[143] Christine Largeron, Christophe Moulin, and Mathias Géry. “Entropy based feature selection for text categorization”. In: *ACM Symposium on Applied Computing*. Ed. by William C. Chu, W. Eric Wong, Mathew J. Palakal, and Chih-Cheng Hung. TaiChung, Taiwan: ACM, Mar. 2011, pp. 924–928 (cit. on pp. 87, 88).

[144] Bassam Al-Salemi, Shahrul Azman Mohd Noah, and Mohd Juzaiddin Ab Aziz. “RFBoost: An improved multi-label boosting algorithm and its application to text categorisation”. In: *Knowledge-Based Systems* 103 (July 1, 2016), pp. 104–117 (cit. on pp. 87, 88, 92).

[145] Guohua Wu, Liuyang Wang, Nailiang Zhao, and Hairong Lin. “Improved Expected Cross Entropy Method for Text Feature Selection”. In: *2015 International Conference on Computer Science and Mechanical Automation (CSMA)*. ISSN: null. Oct. 2015, pp. 49–54 (cit. on p. 88).

[146] Vilmos F. Misangyi, Jeffery A. LePine, James Algina, and Jr Francis Goeddeke. “The Adequacy of Repeated-Measures Regression for Multilevel Research: Comparisons With Repeated-Measures ANOVA, Multivariate Repeated-Measures ANOVA, and Multilevel Modeling Across Various Multilevel Research Designs”. en. In: *Organizational Research Methods* (June 2016) (cit. on p. 88).

[147] Luigi Galavotti, Fabrizio Sebastiani, and Maria Simi. “Experiments on the Use of Feature Selection and Negative Evidence in Automated Text Categorization”. en. In: *Research and Advanced Technology for Digital Libraries*. Ed. by José Borbinha and Thomas Baker. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2000, pp. 59–68 (cit. on p. 88).

[148] Jennifer G. Dy and Carla E. Brodley. “Feature Selection for Unsupervised Learning”. In: *The Journal of Machine Learning Research* 5 (Dec. 2004), pp. 845–889 (cit. on p. 89).

[149] Shital C. Shah and Andrew Kusiak. "Data mining and genetic algorithm based gene/SNP selection". en. In: *Artificial Intelligence in Medicine* 31.3 (July 2004), pp. 183–196 (cit. on p. 89).

[150] Ronen Meiri and Jacob Zahavi. "Using simulated annealing to optimize the feature selection problem in marketing applications". en. In: *European Journal of Operational Research*. Feature Cluster: Heuristic and Stochastic Methods in Optimization 171.3 (June 2006), pp. 842–858 (cit. on p. 89).

[151] Chris Hans, Adrian Dobra, and Mike West. "Shotgun Stochastic Search for "Large p " Regression". In: *Journal of the American Statistical Association* 102 (Apr. 2005) (cit. on p. 89).

[152] Jain AK and Douglas Zongker. "Feature Selection: Evaluation, Application, and Small Sample Performance". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 19 (Mar. 1997), pp. 153–158 (cit. on p. 89).

[153] Jaesung Lee, Jaegyun Park, Hae-Cheon Kim, and Dae-Won Kim. "Competitive Particle Swarm Optimization for Multi-Category Text Feature Selection". In: *Entropy* 21 (June 2019), p. 602 (cit. on p. 89).

[154] Shervin Minaee, Nal Kalchbrenner, Erik Cambria, et al. "Deep Learning Based Text Classification: A Comprehensive Review". In: *arXiv:2004.03705 [cs, stat]* (Apr. 2020). arXiv: 2004.03705 version: 1 (cit. on p. 89).

[155] A. Radford, Jeffrey Wu, R. Child, et al. *Language Models are Unsupervised Multitask Learners*. en. 2019 (cit. on p. 89).

[156] Tom B. Brown, Benjamin Mann, Nick Ryder, et al. "Language Models are Few-Shot Learners". In: *arXiv:2005.14165 [cs]* (July 2020). arXiv: 2005.14165 (cit. on p. 89).

[157] José M. Alonso, Luis Magdalena, and Gil González-Rodríguez. "Looking for a good fuzzy system interpretability index: An experimental approach". en. In: *International Journal of Approximate Reasoning* 51.1 (Dec. 2009), pp. 115–134 (cit. on p. 90).

[158] C. Mencar and A. M. Fanelli. "Interpretability constraints for fuzzy information granulation". en. In: *Information Sciences* 178.24 (Dec. 2008), pp. 4585–4618 (cit. on p. 90).

[159] A. Moreo, J.L. Castro, and J.M. Zurita. "Towards portable natural language interfaces based on case-based reasoning". In: *Journal of Intelligent Information Systems* 49.2 (2017), pp. 281–314 (cit. on p. 91).

[160] A. Moreo, E.M. Eisman, J.L. Castro, and J.M. Zurita. "Learning regular expressions to template-based FAQ retrieval systems". In: *Knowledge-Based Systems* 53 (2013), pp. 108–128 (cit. on p. 91).

[161]A. Moreo, M. Romero, J.L. Castro, and J.M. Zurita. “FAQtory: A framework to provide high-quality FAQ retrieval systems”. In: *Expert Systems with Applications* 39.14 (2012), pp. 11525–11534 (cit. on p. 91).

[162]D.M. Blei, A.Y. Ng, and M.I. Jordan. “Latent Dirichlet allocation”. In: *Journal of Machine Learning Research* 3.4-5 (2003), pp. 993–1022 (cit. on p. 91).

[163]Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D. Manning. “Labeled LDA: a supervised topic model for credit attribution in multi-labeled corpora”. en. In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing Volume 1 - EMNLP ’09*. Vol. 1. Singapore: Association for Computational Linguistics, 2009, p. 248 (cit. on p. 91).

[164]Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. “Efficient Estimation of Word Representations in Vector Space”. In: *arXiv:1301.3781 [cs]* (Jan. 2013). arXiv: 1301.3781 (cit. on p. 91).

[165]Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. “Enriching Word Vectors with Subword Information”. In: *arXiv:1607.04606 [cs]* (July 2016). arXiv: 1607.04606 (cit. on p. 91).

[166]Alec Radford, Jeffrey Wu, Rewon Child, et al. “Language models are unsupervised multitask learners”. In: *OpenAI Blog* 1.8 (2019) (cit. on p. 91).

[167]Rishabh Mehrotra, Scott Sanner, Wray Buntine, and Lexing Xie. “Improving LDA topic models for microblogs via tweet pooling and automatic labeling”. en. In: *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval - SIGIR ’13*. Dublin, Ireland: ACM Press, 2013, p. 889 (cit. on p. 91).

[168]Alec Go, Richa Bhayani, and Lei Huang. “Twitter sentiment classification using distant supervision”. In: *Processing* 150 (Jan. 2009) (cit. on p. 92).

[169]Julio Villena-Roman, Sara Lana-Serrano, Eugenio Martinez-Camara, and Jose Carlos Gonzalez-Cristobal. “TASS - Workshop on Sentiment Analysis at SEPLN”. In: *Procesamiento del Lenguaje Natural* 50.0 (2015), pp. 37–44 (cit. on p. 92).

[170]George A. Miller. “The magical number seven, plus or minus two: some limits on our capacity for processing information”. In: *Psychological Review* 63.2 (1956), pp. 81–97 (cit. on p. 94).

[171]N. Cowan. “The magical number 4 in short-term memory: a reconsideration of mental storage capacity”. eng. In: *The Behavioral and Brain Sciences* 24.1 (Feb. 2001), 87–114, discussion 114–185 (cit. on p. 94).

[172]G.N. Harywanto, J.S. Veron, and D. Suhartono. “A BERTweet-based design for monitoring behaviour change based on five doors theory on coral bleaching campaign”. English. In: *Journal of Big Data* 9.1 (2022) (cit. on p. 109).

[173] I. Ajala, S. Feroze, M. El Barachi, et al. “Combining artificial intelligence and expert content analysis to explore radical views on twitter: Case study on far-right discourse”. English. In: *Journal of Cleaner Production* 362 (2022) (cit. on p. 109).

[174] L. Luo, Y. Wang, and H. Liu. “COVID-19 personal health mention detection from tweets using dual convolutional neural network”. English. In: *Expert Systems with Applications* 200 (2022) (cit. on p. 109).

[175] F.K. Sufi. “AI-SocialDisaster: An AI-based software for identifying and analyzing natural disasters from social media”. English. In: *Software Impacts* 13 (2022) (cit. on p. 109).

[176] Y. Zhang, K. Chen, Y. Weng, et al. “An intelligent early warning system of analyzing Twitter data using machine learning on COVID-19 surveillance in the US”. English. In: *Expert Systems with Applications* 198 (2022) (cit. on p. 109).

[177] H. Ji, J. Wang, B. Meng, et al. “Research on adaption to air pollution in Chinese cities: Evidence from social media-based health sensing”. English. In: *Environmental Research* 210 (2022) (cit. on p. 109).

[178] R. Dutta. “To Find the Best-Suited Model for Sentiment Analysis of Real-Time Twitter Data”. English. In: *Advances in Intelligent Systems and Computing* 1165 (2021). ISBN: 9789811551123, pp. 445–452 (cit. on p. 109).

[179] Anisha P Rodrigues, Roshan Fernandes, Aakash A, et al. “Real-Time Twitter Spam Detection and Sentiment Analysis using Machine Learning and Deep Learning Techniques”. en. In: *Computational Intelligence and Neuroscience* 2022 (Apr. 2022). Ed. by Muhammad Ahmad, pp. 1–14 (cit. on p. 110).

[180] Felipe Machorro Ramos, María Vanessa Romero Ortiz, and Nancy Maribel Arratia Martínez. “Análisis en tiempo real de los sentimientos expresados en Twitter de los votantes durante un debate presidencial en México”. Spanish. In: *Revista Ibérica de Sistemas e Tecnologias de Informação* E47 (2022). Num Pages: 414-424 Place: Lousada, Portugal Publisher: Associação Ibérica de Sistemas e Tecnologias de Informacao, pp. 414–424 (cit. on p. 110).

[181] M. El Barachi, M. AlKhatib, S. Mathew, and F. Oroumchian. “A novel sentiment analysis framework for monitoring the evolving public opinion in real-time: Case study on climate change”. English. In: *Journal of Cleaner Production* 312 (2021) (cit. on p. 110).

[182] Annamalai University, Annamalainagar, Tamil Nadu and Lakshmana Phaneendra Maguluri. “A New sentiment score based improved Bayesian networks for real-time intraday stock trend classification”. en. In: *International Journal of Advanced Trends in Computer Science and Engineering* 8.1.4 (Sept. 2019), pp. 1045–1055 (cit. on p. 110).

[183]Yong-Ting Wu, He-Yen Hsieh, Xanno K. Sigalingging, Kuan-Wu Su, and Jenq-Shiou Leu. “RIVA : A Real-Time Information Visualization and analysis platform for social media sentiment trend”. In: *2017 9th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*. ISSN: 2157-023X. Nov. 2017, pp. 256–260 (cit. on p. 110).

[184]Javier Borge-Holthoefer, Walid Magdy, Kareem Darwish, and Ingmar Weber. “Content and Network Dynamics Behind Egyptian Political Polarization on Twitter”. In: *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*. CSCW ’15. New York, NY, USA: Association for Computing Machinery, Feb. 2015, pp. 700–711 (cit. on p. 110).

[185]Mesut Kaya, Guven Fidan, and İsmail Hakkı Toroslu. “Transfer Learning Using Twitter Data for Improving Sentiment Classification of Turkish Political News”. tr. In: Oct. 2013 (cit. on p. 110).

[186]Peter Sheridan Dodds, Kameron Decker Harris, Isabel M. Kloumann, Catherine A. Bliss, and Christopher M. Danforth. “Temporal Patterns of Happiness and Information in a Global Social Network: Hedonometrics and Twitter”. en. In: *PLOS ONE* 6.12 (Dec. 2011). Publisher: Public Library of Science, e26752 (cit. on p. 110).

[187]Johan Bollen, Huina Mao, and Xiaojun Zeng. “Twitter mood predicts the stock market”. en. In: *Journal of Computational Science* 2.1 (Mar. 2011), pp. 1–8 (cit. on p. 110).

[188]Elena Kochkina, Maria Liakata, and Arkaitz Zubiaga. “PHEME dataset for Rumour Detection and Veracity Classification”. In: (June 2018) (cit. on pp. 110, 124).

[189]Radford Neal. *STA 247 - Week 7 lecture summary* (cit. on p. 111).

[190]David Blei and Rebecca Fiebrink. *COS 424: Interacting with Data*. en. 2007 (cit. on p. 111).

[191]Meelis Kull and Peter Flach. “Patterns of dataset shift”. en. In: (), p. 10 (cit. on p. 111).

[192]Alexandros Iosifidis and Anastasios Tefas. *Deep Learning for Robot Perception and Cognition*. en. Google-Books-ID: 4EU6EAAAQBAJ. Academic Press, Feb. 2022 (cit. on p. 111).

[193]Fred Morstatter and Huan Liu. “Discovering, assessing, and mitigating data bias in social media”. en. In: *Online Social Networks and Media* 1 (June 2017), pp. 1–13 (cit. on pp. 113, 114).

[194]Fred Morstatter, J. Pfeffer, Huan Liu, and Kathleen M. Carley. “Is the Sample Good Enough? Comparing Data from Twitter’s Streaming API with Twitter’s Firehose”. In: *ICWSM* (2013) (cit. on p. 114).

[195] Fred Morstatter, Jürgen Pfeffer, and Huan Liu. “When is it biased? assessing the representativeness of twitter’s streaming API”. In: *Proceedings of the 23rd International Conference on World Wide Web*. WWW ’14 Companion. New York, NY, USA: Association for Computing Machinery, Apr. 2014, pp. 555–556 (cit. on p. 114).

[196] Fred Morstatter, Harsh Dani, Justin Sampson, and Huan Liu. “Can One Tamper with the Sample API? Toward Neutralizing Bias from Spam and Bot Content”. In: *Proceedings of the 25th International Conference Companion on World Wide Web*. WWW ’16 Companion. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, Apr. 2016, pp. 81–82 (cit. on p. 114).

[197] Jürgen Pfeffer, Katja Mayer, and Fred Morstatter. “Tampering with Twitter’s Sample API”. en. In: *EPJ Data Science* 7.1 (Dec. 2018). Number: 1 Publisher: Springer Berlin Heidelberg, p. 50 (cit. on p. 114).

[198] Kenneth Joseph, Peter M. Landwehr, and Kathleen M. Carley. “Two 1%s Don’t Make a Whole: Comparing Simultaneous Samples from Twitter’s Streaming API”. en. In: *Social Computing, Behavioral-Cultural Modeling and Prediction*. Ed. by David Hutchison, Takeo Kanade, Josef Kittler, et al. Vol. 8393. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2014, pp. 75–83 (cit. on p. 114).

[199] Instagram. *See Posts You Care About First in Your Feed | Instagram Blog*. en. 2016 (cit. on p. 115).

[200] Gediminas Adomavicius, Jesse Bockstedt, Shawn Curley, and Jingjng Zhang. *Reducing Recommender Systems Biases: An Investigation of Rating Display Designs*. en. SSRN Scholarly Paper. Rochester, NY, Feb. 2019 (cit. on p. 115).

[201] Marcus O’Dair and Andrew Fry. “Beyond the black box in music streaming: the impact of recommendation systems upon artists”. In: *Popular Communication* 18.1 (Jan. 2020). Publisher: Routledge _eprint: https://doi.org/10.1080/15405702.2019.1627548, pp. 65–77 (cit. on p. 115).

[202] Paul Lewis. “’Fiction is outperforming reality’: how YouTube’s algorithm distorts truth”. en-GB. In: *The Guardian* (Feb. 2018) (cit. on p. 115).

[203] Guillaume Chaslot. *How YouTube’s A.I. boosts alternative facts*. en. Apr. 2017 (cit. on p. 115).

[204] Renjie Zhou, Samamon Khemmarat, and Lixin Gao. “The impact of YouTube recommendation system on video views”. In: *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. IMC ’10. New York, NY, USA: Association for Computing Machinery, Nov. 2010, pp. 404–410 (cit. on p. 115).

[205] Amit Sharma, Jake M. Hofman, and Duncan J. Watts. “Estimating the Causal Impact of Recommendation Systems from Observational Data”. In: *Proceedings of the Sixteenth ACM Conference on Economics and Computation*. EC ’15. New York, NY, USA: Association for Computing Machinery, June 2015, pp. 453–470 (cit. on p. 115).

[206] Sunshine Chong and Andrés Abeliuk. “Quantifying the Effects of Recommendation Systems”. In: *2019 IEEE International Conference on Big Data (Big Data)*. Dec. 2019, pp. 3008–3015 (cit. on p. 115).

[207] Mark Zuckerberg. *A Blueprint for Content Governance and Enforcement | Facebook*. en-us. 2018 (cit. on p. 115).

[208] Nikki Usher, Jesse Holcomb, and Justin Littman. “Twitter Makes It Worse: Political Journalists, Gendered Echo Chambers, and the Amplification of Gender Bias”. en. In: *The International Journal of Press/Politics* 23.3 (July 2018). Publisher: SAGE Publications Inc, pp. 324–344 (cit. on p. 116).

[209] Adam Badawy, Emilio Ferrara, and Kristina Lerman. “Analyzing the Digital Traces of Political Manipulation: The 2016 Russian Interference Twitter Campaign”. In: *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. ISSN: 2473-991X. Aug. 2018, pp. 258–265 (cit. on p. 116).

[210] Jayeon Lee and Weiai Xu. “The more attacks, the more retweets: Trump’s and Clinton’s agenda setting on Twitter”. en. In: *Public Relations Review* 44.2 (June 2018), pp. 201–213 (cit. on p. 116).

[211] datareportal. *The Latest Twitter Statistics: Everything You Need to Know*. en-GB. 2022 (cit. on p. 116).

[212] financesonline. *Number of Twitter Users 2022/2023: Demographics, Breakdowns & Predictions*. en. Mar. 2020 (cit. on p. 116).

[213] Alan Mislove, Sune Lehmann, Yong-Yeol Ahn, Jukka-Pekka Onnela, and J. Rosenquist. “Understanding the Demographics of Twitter Users”. en. In: *Proceedings of the International AAAI Conference on Web and Social Media* 5.1 (2011). Number: 1, pp. 554–557 (cit. on p. 116).

[214] Renato Miranda Filho, Jussara M. Almeida, and Gisele L. Pappa. “Twitter Population Sample Bias and its impact on predictive outcomes: a case study on elections”. In: *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*. ASONAM ’15. New York, NY, USA: Association for Computing Machinery, Aug. 2015, pp. 1254–1261 (cit. on p. 116).

[215] Ourania Kounadi, Alina Ristea, Michael Leitner, and Chad Langford. “Population at risk: using areal interpolation and Twitter messages to create population models for burglaries and robberies”. In: *Cartography and Geographic Information Science* 45.3 (May 2018). Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/15230406.2017.1304243, pp. 205–220 (cit. on p. 116).

[216] Adam Bermingham and Alan Smeaton. “On Using Twitter to Monitor Political Sentiment and Predict Election Results”. In: *Proceedings of the Workshop on Sentiment Analysis where AI meets Psychology (SAAIP 2011)*. Chiang Mai, Thailand: Asian Federation of Natural Language Processing, Nov. 2011, pp. 2–10 (cit. on p. 117).

[217] Eduardo Graells-Garrido, Ricardo Baeza-Yates, and Mounia Lalmas. “How Representative is an Abortion Debate on Twitter?” In: *Proceedings of the 10th ACM Conference on Web Science*. WebSci ’19. New York, NY, USA: Association for Computing Machinery, June 2019, pp. 133–134 (cit. on p. 117).

[218] Wei Gao and Fabrizio Sebastiani. “Tweet Sentiment: From Classification to Quantification”. In: *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*. ASONAM ’15. New York, NY, USA: Association for Computing Machinery, Aug. 2015, pp. 97–104 (cit. on pp. 117, 122).

[219] Alejandro Moreo and Fabrizio Sebastiani. *Tweet Sentiment Quantification: An Experimental Re-Evaluation*. Tech. rep. SoBigData++ and AI4Media, 2020 (cit. on pp. 117, 121, 122).

[220] David Vilares, Yerai Doval, Miguel A. Alonso, and Carlos Gómez-Rodríguez. “LyS at SemEval-2016 Task 4: Exploiting Neural Activation Values for Twitter Sentiment Classification and Quantification”. en. In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California: Association for Computational Linguistics, 2016, pp. 79–84 (cit. on p. 117).

[221] Andrea Esuli and Fabrizio Sebastiani. “Optimizing Text Quantifiers for Multivariate Loss Functions”. en. In: *ACM Transactions on Knowledge Discovery from Data* 9.4 (June 2015), pp. 1–27 (cit. on p. 117).

[222] Pablo González, Jorge Díez, Nitesh Chawla, and Juan José del Coz. “Why is quantification an interesting learning problem?” en. In: *Progress in Artificial Intelligence* 6.1 (Mar. 2017), pp. 53–58 (cit. on p. 117).

[223] George Forman. “Quantifying counts and costs via classification”. In: *Data Mining and Knowledge Discovery* 17.2 (Oct. 2008), pp. 164–206 (cit. on pp. 118, 122).

[224] Antonio Bella, Cesar Ferri, José Hernández-Orallo, and María José Ramírez-Quintana. “Quantification via Probability Estimators”. In: *2010 IEEE International Conference on Data Mining*. ISSN: 2374-8486. Dec. 2010, pp. 737–742 (cit. on pp. 119, 122).

[225] George Forman. “Quantifying trends accurately despite classifier error and class imbalance”. In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. KDD ’06. New York, NY, USA: Association for Computing Machinery, Aug. 2006, pp. 157–166 (cit. on p. 120).

[226] Letizia Milli, Anna Monreale, Giulio Rossetti, et al. “Quantification Trees”. In: *2013 IEEE 13th International Conference on Data Mining*. ISSN: 2374-8486. Dec. 2013, pp. 528–536 (cit. on p. 120).

[227] Jose Barranquero, Pablo González, Jorge Díez, and Juan José del Coz. “On the study of nearest neighbor algorithms for prevalence estimation in binary problems”. In: *Pattern Recognition* 46.2 (Feb. 2013), pp. 472–482 (cit. on pp. 120, 122, 123).

[228] Pablo Pérez-Gállego, José Ramón Quevedo, and Juan José del Coz. “Using ensembles for problems with characterizable changes in data distribution”. In: *Information Fusion* 34.C (Mar. 2017), pp. 87–100 (cit. on p. 120).

[229] Marco Saerens, Patrice Latinne, and Christine Decaestecker. “Adjusting the Outputs of a Classifier to New a Priori Probabilities: A Simple Procedure”. en. In: *Neural Computation* 14.1 (Jan. 2002), pp. 21–41 (cit. on p. 121).

[230] Slobodan Vucetic and Zoran Obradovic. “Classification on Data with Biased Class Distribution”. In: *Proceedings of the 12th European Conference on Machine Learning*. EMCL ’01. Berlin, Heidelberg: Springer-Verlag, Sept. 2001, pp. 527–538 (cit. on p. 121).

[231] George Forman. “Counting positives accurately despite inaccurate classification”. In: *Proceedings of the 16th European conference on Machine Learning*. ECML’05. Berlin, Heidelberg: Springer-Verlag, Oct. 2005, pp. 564–575 (cit. on p. 121).

[232] Jose Barranquero, Jorge Díez, and Juan José del Coz. “Quantification-oriented learning based on reliable classifiers”. en. In: *Pattern Recognition* 48.2 (Feb. 2015), pp. 591–604 (cit. on p. 122).

[233] Alejandro Moreo, Andrea Esuli, and Fabrizio Sebastiani. “QuaPy: A Python-based framework for quantification”. In: *Proceedings of the 30th ACM International Conference on Knowledge Management (CIKM 2021)*. Gold Coast, AU, 2021, pp. 4534–4543 (cit. on p. 125).

[234] Miguel Angel Benitez-Castro and Encarnacion Hidalgo-Tenorio. *Chapter 12. Rethinking Martin & White’s affect taxonomy*. English. 2019 (cit. on p. 155).

[235] Roberto Barbeito Iglesias and Ángel Iglesias Alonso. “Political emotions and digital political mobilization in the new populist parties: the cases of Podemos and Vox in Spain”. In: *International Review of Sociology* 31.2 (May 2021). Publisher: Routledge _eprint: https://doi.org/10.1080/03906701.2021.1947948, pp. 246–267 (cit. on p. 156).

[236] Eva Aladro Vico and Paula Requeijo Rey. “Discurso, estrategias e interacciones de Vox en su cuenta oficial de Instagram en las elecciones del 28-A. Derecha radical y redes sociales”. es. In: *Revista Latina de Comunicación Social* 77 (July 2020). Number: 77, pp. 203–229 (cit. on p. 156).

[237] Paolo Gerbaudo. “Social media and populism: an elective affinity?” en. In: *Media, Culture & Society* 40.5 (July 2018). Publisher: SAGE Publications Ltd, pp. 745–753 (cit. on p. 156).

[238] *Global Twitter user age distribution 2021*. en (cit. on p. 158).

[239] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. “Pearson Correlation Coefficient”. en. In: *Noise Reduction in Speech Processing*. Vol. 2. Series Title: Springer Topics in Signal Processing. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 1–4 (cit. on p. 175).

[240] Larry L. Havlicek and Nancy L. Peterson. “Robustness of the Pearson Correlation against Violations of Assumptions”. In: *Perceptual and Motor Skills* 43.3_suppl (Dec. 1976). Publisher: SAGE Publications Inc, pp. 1319–1334 (cit. on p. 175).

[241] Jiawei Han, Hong Cheng, Dong Xin, and Xifeng Yan. “Frequent pattern mining: current status and future directions”. en. In: *Data Mining and Knowledge Discovery* 15.1 (July 2007), pp. 55–86 (cit. on p. 181).

[242] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. “Mining association rules between sets of items in large databases”. In: *ACM SIGMOD Record* 22.2 (June 1993), pp. 207–216 (cit. on p. 181).

[243] Sergey Brin, Rajeev Motwani, Jeffrey D. Ullman, and Shalom Tsur. “Dynamic itemset counting and implication rules for market basket data”. In: *ACM SIGMOD Record* 26.2 (June 1997), pp. 255–264 (cit. on p. 182).

[244] Gregory Piatetsky-Shapiro. “Discovery, Analysis, and Presentation of Strong Rules”. In: *Knowledge Discovery in Databases*. Ed. by Gregory Piatetsky-Shapiro and William J. Frawley. AAAI/MIT Press, 1991, pp. 229–248 (cit. on p. 182).

List of Figures

List of Tables

Acronyms

ACC Adjusted Classify and Count. 119, 120, 125, 128, 132, 141, 148, 150, 152

AE Absolute Error. 123, 125

AI Artificial Intelligence. 21

APP Artificial Prevalence Protocol. 122, 125

CAL Continuous Active Learning. 22

CC Classify and Count. 118–120, 125, 128, 129, 147, 148, 150

DNN Deep Artificial Neural Networks. 4, 14, 15, 84

EMQ Expectation-Maximisation (*Saerens-Latinne-Decaestecker*). 121, 125, 128, 129, 132, 147, 150

KLD *Kullback-Leibler* Divergence. 123, 125

MAE Mean Absolute Error. 123

ML Machine Learning. xi, xii, 1, 4, 7, 8, 11, 21, 58, 64, 77, 83, 85, 107, 111, 118, 187, 188, 193

MRAE Mean Relative Absolute Error. 123, 124

MSE Mean Squared Error. 123

NLP Natural Language Processing. 22

NPP Natural Prevalence Protocol. 122, 125

PACC Probabilistic Adjusted Classify and Count. 119, 120, 125, 128–130, 132, 141, 143, 145, 147, 148, 150, 152, 188, 195, 223, 224

PCC Probabilistic Classify and Count. 119, 120, 125, 128–130, 132, 143, 147, 148, 150, 152, 153, 188, 189, 195, 223

RAE Relative Absolute Error. 123, 125

SE Squared Error. 123, 125

SNS Social Networking Sites. xi, 1–3, 5–9, 11, 21–23, 34, 39, 41, 78, 81, 84, 86, 107, 109, 110, 113, 115, 128, 130, 152, 158, 187, 190, 191

SVM Support Vector Machine. 117

Curriculum Vitae

Personal Information

Manuel Francisco Aparicio
ORCiD: 0000-0001-9748-2269 | francisco@decsai.ugr.es
Rafael Gómez, 2
CITIC-Univ. Granada
Granada, Spain

Education

2018-2022	PhD studies at Computational Intelligence research group (TIC210), University of Granada, Spain
2016-2017	MSc. Data Science and Computer Engineering, University of Granada, Spain
2012-2016	BSc. Computer Engineering, University of Cádiz, Spain

Teaching & Research Experience

Feb. 2022-Apr. 2022	Visiting Research Scientist, Consiglio Nazionale delle Ricerche, Pisa, Italy
Jul. 2018-Nov. 2022	FPI Research and Teaching Fellow, University of Granada
Mar. 2018-Jun. 2018	Head Teacher, Forinsur Centro de Formación, Cádiz
Sep. 2017-Ene. 2018	Research Support Staff, University of Granada
Feb. 2017-Jul. 2017	Webmaster in School for Postgraduate Studies, University of Granada

Publications List

Publications

Francisco, M., Castro, Juan L. (2020) A fuzzy model to enhance user profiles in microblogging sites using deep relations. In: Fuzzy Sets and Systems 401, pp. 133-149, `https://doi.org/10.1016/j.fss.2020.05.006`

Francisco, M., Castro, Juan L. (under review) A Methodology to Quickly Perform Opinion Mining and Build Supervised Datasets Using Social Networks Mechanics. Sent to: IEEE Transactions on Knowledge and Data Engineering (under review)

Francisco, M., Benítez-Castro, M.A., Hidalgo-Tenorio, E., Castro, J.L. (2022) A semi-supervised algorithm for detecting extremism propaganda diffusion on social media. In: Pragmatics and Society, Volume 13, Issue 3, pp. 532-554, `https://doi.org/10.1075/ps.21009.fra`

Conferences

Francisco, M., Castro, J.L. (2019) Extending clusters of Social Network Users with Deep Relations. In: 11th Conference of the European Society for Fuzzy Logic (EUSFLAT), Prague, Czech Republic.

Francisco, M., Dhiab Hassan, A., Benítez-Castro, M.A., Hidalgo-Tenorio, E., Castro, J.L. (2019) A semi-supervised algorithm for detecting extremist propaganda dissemination in social media. In: 6th Languaging Diversity International Conference (LD2019), Teruel, Spain.

García-Luengo, O., Francisco, M. (2021). Televised Debates in Times of COVID-19. In: 21st Internationla Scientific Conference "Europe of 21st Century", Słubice, Poland.

Castro, J.L., Francisco, M. (2022). Similarity Fuzzy Semantic Network for Social Media Analysis. In: Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2022), Milan, Italy.

Francisco, M., Castro, J.L. (2022). Social Media Analysis: From individual tweets to group analysis with minimum effort. In: Discourse, Politics and Extreme Ideologies, Granada, Spain.

Francisco, M., Castro, J.L. (2022). Extracting relevant expressions to help characterise user profiles in Social Networking Sites. In: Approaches to Digital Discourse Analysis (ADDA3), USF-s St. Petersburg, Florida, USA.

Francisco, M., Castro, J.L. (2022). Discriminatory Expressions to Improve Model Comprehensibility in Short Documents. In: 3rd International Conference on Pattern Recognition and Artificial Intelligence (ICPRAI 2022), Paris, France.

Castro, J.L., Francisco, M. (2022). Similarity Fuzzy Semantic Networks and Inference. An application to analysis of radical discourse in Twitter. In: 22nd International Conference on Artificial Intelligence and Soft Computing (ICAISC 2022), Zakopane, Poland.

Book Chapters

Francisco, M., Castro, J.L. (2022) Discriminatory Expressions to Improve Model Comprehensibility in Short Documents. In: Pattern Recognition and Artificial Intelligence, ICPRAI 2022 Proceedings Part I, pp. 311–322. doi: 10.1007/978-3-031-09037-0_26.

Castro, J.L., Francisco, M. (2022) Similarity Fuzzy Semantic Network for Social Media Analysis. In: Information Processing and Management of Uncertainty in Knowledge-Based Systems. IPMU 2022. Communications in Computer and Information Science, vol 1601, pp. 557-567. Springer, Cham. doi: 10.1007/978-3-031-08971-8_46.

A Guidelines for annotators

You can find below the initial guidelines used to train the initial set of experts that collaborated with us in the annotation process. Notice that both the interface and the process differ from the ones described along this document, since it was in an early development stage.

Annotation Guidelines

The Nutcracker Project

Introduction

Nowadays, information flow is huge. From traditional media such as newspapers and television to blogs and forums, sources of information have grown to a point where we seem not to be able to conceive of our reality without them. Social Media has become one of the most important sources of communication. Users from all around the world can now manifest their opinions on all sorts of matters and share content regardless of their socio-economic status. This makes Social Networking Sites (SNS) a baseline to analyse people's interest and opinion trends.

Microblogging sites are prone to multiple analysis applications. They can be understood as a combination of blogging (since general public can access your history of publications) and instant messaging (because messages are supposed to be short enough to keep a fluid communication). Messages may be multimodal and include images and videos as well.

When it comes to studying Twitter, we can observe some restrictions as to how to perform the analysis of their content, which constrains or impoverishes machine learning (ML) models capacities, namely, lack of context; abundance of misspelled words, contractions and acronyms; and new semantic units (like hashtags), among others. On top of that, there is a lack of annotated public datasets without stripped information regarding relations between tweets and/or users.

In this context, it is nearly impossible to apply supervised ML algorithms to classify users and explore clusters. Our project requires an algorithm that is capable of modelling relations between groups of users in order to identify and detect accounts that are within the boundaries of our interest (i.e. jihadist propaganda). To this end, we need to 1) design an algorithm that can explore those relations and infer knowledge of *unknown* users taking into account its environment (deep relations); 2) validate the algorithm so that it can be backed by the scientific community; and 3) apply this algorithm to known clusters of jihadist accounts in order to discover new ones. Currently, we are at the second stage, that is, algorithm validation.

Typically, when we analyse what users publish in SNS, we can only *understand* what those users explicitly say. We believe we can use deep relations in order to discover users' hidden attributes. The purpose of this work is to annotate a dataset of tweets to measure our algorithm's performance in this respect. The roadmap is as follows:

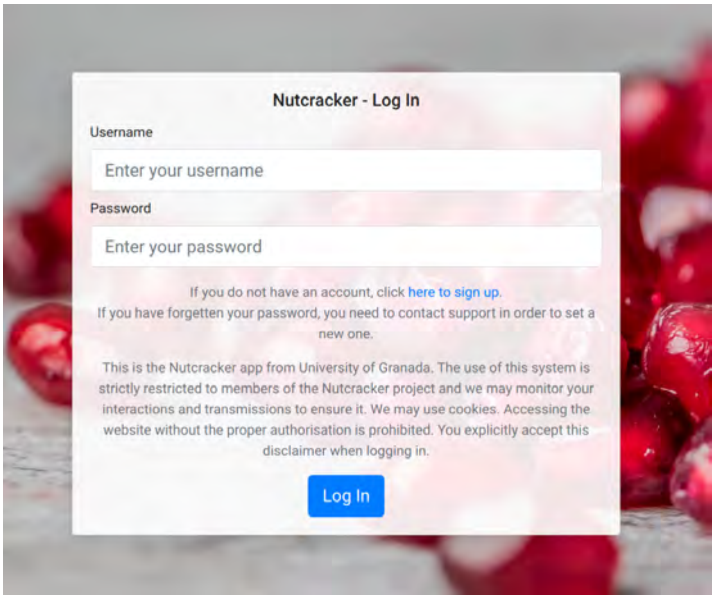
1. Put together a well-annotated dataset within a stable context (so accounts do not get banned while we do our work).
2. Split data into several partitions. Some of them will be used to *teach* (train) our algorithm.
3. Remove specific features of the remaining partitions and see how good our algorithm is when trying to infer those features from the environment.
4. Analyse performance to check if we can trust the output of the trained model.

This process may (and, in fact, should) be iterative so that we can learn from the scenarios where our model fails and improve its implementation.

Although the outcome of the annotation process will be used to validate our proposal, there is nothing that prevents further research in different areas. Ultimately, we are going to create a dataset that can serve multiple purposes. We expect that this work is useful for all of our research interests; subsequently, we expect to to help and participate in other applications of this base of knowledge. In actual fact, we will make this dataset public for the scientific community so that further research can be carried out. Please, take into account that we will need to anonymise everything before publishing, specially when commenting on the tweets.

Tagging Process

The process of tweet annotation may look simple, nevertheless, decision making is not an easy task. The app is available at https://nutracker.ugr.es. Once you enter the website, you will need to sign up.



Nutcracker - Log In

Username

Enter your username

Password

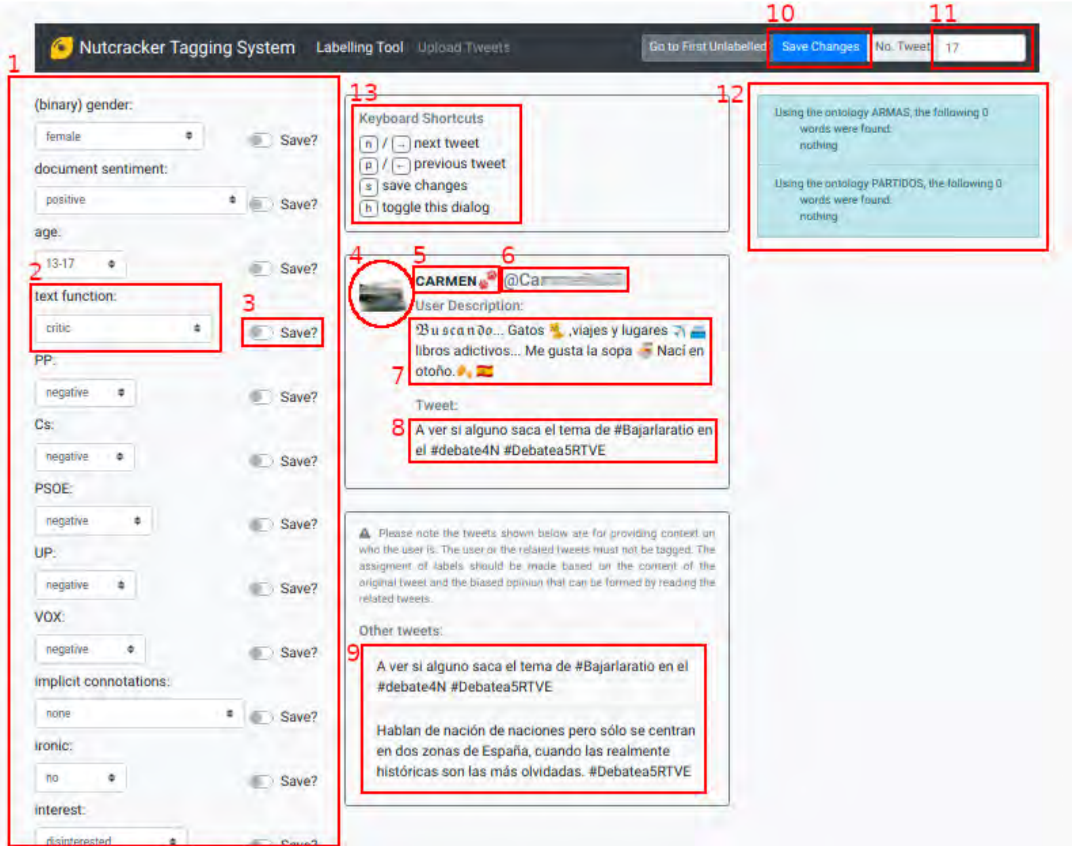
Enter your password

If you do not have an account, click here to sign up.
If you have forgetten your password, you need to contact support in order to set a new one.

This is the Nutcracker app from University of Granada. The use of this system is strictly restricted to members of the Nutcracker project and we may monitor your interactions and transmissions to ensure it. We may use cookies. Accessing the website without the proper authorisation is prohibited. You explicitly accept this disclaimer when logging in.

Notice that there is a link you can follow to create your account. Once you have an account, it needs to be approved by the administrators. For such a purpose, please contact Manuel Francisco <removed>.

Once you have signed in, you will see the following main interface:



No. Tweet: 17

(binary) gender: female ○ Save?

document sentiment: positive ○ Save?

age: 13-17 ○ Save?

text function: critic ○ Save?

PP: negative ○ Save?

Cs: negative ○ Save?

PSOE: negative ○ Save?

UP: negative ○ Save?

VOX: negative ○ Save?

implicit connotations: none ○ Save?

ironic: no ○ Save?

interest: disinterested ○ Save?

Keyboard Shortcuts
n / → next tweet
p / ← previous tweet
s save changes
h toggle this dialog

Using the ontology ARMAS, the following 0 words were found:
nothing

Using the ontology PARTIDOS, the following 0 words were found:
nothing

CARMEN @Ca[redacted]

User Description:
𝔅𝔲𝔰𝔠𝔞𝔫𝔡𝔬... Gatos ,viajes y lugares libros adictivos... Me gusta la sopa Naci en otoño. 🇪🇸

Tweet:
A ver si alguno saca el tema de #Bajarlaratio en el #debate4N #Debatea5RTVE

⚠ Please note the tweets shown below are for providing context on who the user is. The user or the related tweets must not be tagged. The assigment of labels should be made based on the content of the original tweet and the biased opinion that can be formed by reading the related tweets.

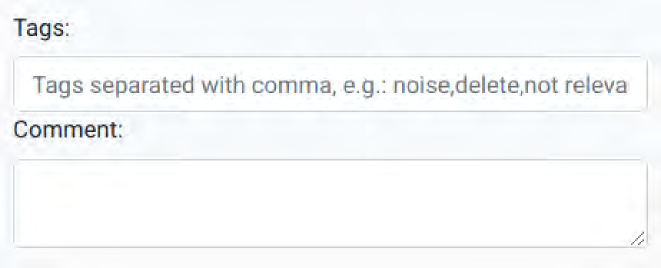
Other tweets:

A ver si alguno saca el tema de #Bajarlaratio en el #debate4N #Debatea5RTVE

Hablan de nación de naciones pero sólo se centran en dos zonas de España, cuando las realmente históricas son las más olvidadas. #Debatea5RTVE

1. Tags column. You can find here a list of features that we are going to annotate.
2. Specific attribute and value. You need to select the correct value for each feature regarding tweet content (8). Based upon the evidence provided by the user or the user account, you should decide upon which is the content of the tweet (8); that is, you will take into account the description of the user (7) and other authored tweets (9).
3. Save button for a specific attribute. You should toggle this switch when you want to save the chosen value for an attribute. If the switch is off, a null value will be saved (this means that the actual value cannot be determined or that the attribute does not apply to the context of the tweet). For example, if the user is FEMALE, you will choose this feature and then, click on SAVE just on the right hand side; if you do not know which the user's gender is, you will leave that feature un-annotated.
4. User's profile picture.
5. User's display name.
6. Username.
7. User's description. You can use the description that users give about themselves to interpret the content of the tweet as well as their socio-demographic variables.
8. Tweet content. This is what we are going to annotate in panel 1.
9. Other tweets from the same author. You can use these tweets to interpret the content of the tweet. Keep in mind that we are not annotating the user but the tweet (8).
10. Save button. After annotating all the aspects that can be applied, you should explicitly save the changes to prevent unwanted annotations. Keep in mind that there is a keyboard shortcut to do it quickly.
11. Tweet selector. You can navigate through the tweets in our database with this selector. You can also use keyboard shortcuts.
12. Ontologies. This panel shows the result of a semantic search performed with custom ontologies. We can add labels and categories to these ontologies at your discretion. Please, do not hesitate to contact us for further information.
13. Keyboard shortcuts. Help with keyboard shortcuts. You can hide this box.

At the end of panel 1, you can find two additional fields that will be useful for your work. "Tags" is a folksonomy (see Wikipedia) that can be used for anything you deem useful. Tags need to be separated with commas and you can use spaces at your discretion. “Comment” is a free text field. You can write in here whatever you want (explanations, doubts…). Both fields need to be explicitly saved (with the save changes button (10) or with its keyboard shortcut).


Tags:

Tags separated with comma, e.g.: noise,delete,not releva

Comment:

In order to avoid overwriting problems, we are going to divide our database in batches (e.g. from 1 to 200, from 201 to 400, from 401 to 600, etc.). Please, we ask you not to interfere with other people’s work by only saving changes in tweets within your assigned batches. You can check your assigned batches and auto-assign new ones to you in the following document

<removed>

Anyone who has this link can make changes to the document. You cannot share it with people who do not belong to the Nutcracker Project.

How to Deal With Errors

We run unitary and integration tests with each release of the tool. However, you may find errors when working with the website. In order to fix them, we will need a step-by-step guide to reproduce the error. Please, write down everything you were doing before the error came up and contact us as soon as possible (<removed>). You should also specify the error you encountered (attach a screenshot if possible). In the event that we cannot reproduce the error in our testing computer, we may also ask you for a meeting (online or face-to-face) so we can identify it in your computer. We apologise beforehand for any problem this may cause you.

Available Features and Values

In case there might be a drawing or a picture that might be useful to infer the meaning and function of the tweet, we should retrieve it from Twitter, annotate the features accordingly and indicate it in the Comment Section.

The features (aspects) that we decided to annotated are listed below:

- Gender (female/male/other). In this case, we will take into account the name of the user (e.g. “Carmen Fernández” will be annotated as FEMALE) or how they describe themselves (e.g. “Soy un hombre comprometido” will be annotated as MALE). Profile pictures may be confusing and misleading.

- Document Sentiment (negative/neutral/positive). Overall sentiment of the tweet; in other words, we will annotate the opinions expressed in a of text, especially in order to determine whether the writer's attitude towards a topic, a particular product, an individual, an institution, an event, someone’s attitudes, etc. is positive, negative, or neutral. We must remember that the annotator’s belief systems are not relevant here. That is, to kill Muslims or Christians may be something good from the perspective of those who hate Muslims or Christians, even though from the perspective of a normal human being, in general, killing is a wrongful act, unless what we kill is a virus like covid-19.

- Age (several discrete age groups). How old the user is. Try to find evidence.

- Pragmatic Function (literal/metaphorical/ironic/sarcastic): Given that the challenge here is perhaps whether one text is ironic or not, we will pay attention to how hyperbole (e.g. “Es listísimo”), opposition (e.g. “Cuánto lo amo”) and understatement (e.g. “Es algo malo”) operate in context so that there is some incongruence between what the intended meaning is and what the user has actually written. Although sarcasm is a harsh/sharp form of irony meant to ridicule someone, we can include it here under the umbrella term “pragmatic function” in those cases when the author clearly aims to hurt someone else. Finally, let’s indicate as well those tweets that are metaphorical. We will analyse them manually later.

- Mood (realis/irrealis): it applies to the main clause. Realis: Factual information, plain statements, statements of fact (*Something has happened / Something has not happened (both assertive and non-assertive*). Irrealis: A certain situation or action is not known to have happened at the moment the speaker is talking / Something that hasn’t yet occurred. TO CUT A LONG STORY SHORT: Irrealis would apply to any: i) imperative and interrogative clause; ii) any main clause containing an explicit/implicit epistemic or deontic marker (will, may, can, think, guess, believe, should, must, have to, etc.); iii) mental processes which by their very nature are irrealis (e.g. desiderative:

want, would like to, etc.); and iv) conditional clauses (if, unless…).

REALIS	IRREALIS
(1) You are happy (2) You are not happy (3) You are horrible (4) You are not horrible (5) You used to be my friend (6) I used to be happy (=> I was in the past, but not anymore!) (7) She asked me to be her friend (=> We focus on the main clause => Given that ask is a verbal performative process, the asking is something that really happened; hence: realis).	**Subjunctive:** (8) If I loved you, May you be happy **Conditional:** (8) I would love you **Optative:** (8) May I be loved! **Jussive (DEONTIC MODALITY):** (9) Everyone should be loved **Potential:** (10) She probably loves me; she may hate me **Imperative:** (11) Love me!!!! Do not love me!! **Desiderative:** (12) If wish she loved me! I want her to love me! **Dubitative:** (13) I think she loves me, I'm not sure she loves me **Hypothtetical:** (14) I might love you (if…) **Permissive:** (15) You can/may love me! **Hortative:** (16) Let us love each other! **Precative and interrogative:** (17) Will you love me? Does she love me?

FURTHER EXAMPLES:
(18) I used to think I loved her
(19) Be my friend!
(20) I wanted her to be my friend (cf. however I asked her to be my friend=> REALIS)

FOR FURTHER INFO ON THE REALIS/IRREALIS DISTINCTION, PLEASE SEE:

https://en.wikipedia.org/wiki/Irrealis_mood

- PP (negative/positive). If the document is negative or positive towards this political party.

- PSOE (negative/positive). If the document is negative or positive towards this political party.

- Cs (negative/positive). If the document is negative or positive towards this political party.

- UP (negative/positive). If the document is negative or positive towards this political party.

- VOX (negative/positive). If the document is negative or positive towards this political party.

- Implicit connotations (none/present). Sometimes one tweet will clearly be interpreted explicitly as one particular emotion category and implicitly as another opinion category, or the other way around: it can be interpreted explicitly as one particular opinion category and implicitly as one emotion category. In those instances where the example shows a two-tiered opinion-emotion annotation, the coder will indicate the presence of IMPLICIT CONNNOTATION and subsequently tag both, the emotion and the opinion, or the other way around. See below some examples of explicit opinion and implicit emotion:

    a. Ethical judgement about someone's veracity or normality may be read as goal achievement emotion_satisfaction or goal achievement emotion_dissatisfaction, as well as goal relation emotion_liking or goal relation emotion_disgust. For instance, if someone claims that "Some politicians are honest or normal", they may implicitly mean that they believe in them or that they like them; if they say, instead, that "Some people are deceptive or odd", they may implicitly mean that they feel very insecure about them, or that they dislike them;

    b. Aesthetic appreciation may be read as goal relation emotion_liking or goal relation emotion_disgust. For instance, if someone says that "An object is beautiful", they may implicitly mean that they love it; if they say, instead, that "One person is ugly", they may implicitly mean that they are not very fond of that particular human being;

    c. Ethical judgement about the propriety of someone's actions with an impact on the emoter may be read as goal relation emotion_affection or goal relation emotion_antipathy. For instance, “They have taken care of his niece” may implicitly mean that they love her; instead, “They have hurt that old man” may implicitly mean that they acted that way because they hated him;

d. Ethical judgement about someone’s tenacity (or capacity) be read as goal relation emotion_respect or goal relation emotion_disrespect. For instance, if someone says that “The young woman was brave”, this may implicitly mean that they admire her for her courage; if they say, instead, that “The young woman was lazy”, they may implicitly mean that they feel contempt for this female on account of her laziness.

- Surprise (no/yes). If X writes “A los del Partido Y les sorprende que el covid-19 se haya extendido”, we can annotate it as SURPRISE, because of the presence of the verb “sorprende” (or expressions of similar meaning).

- Interest (interested/uninterested). If X writes “Los del Partido Y tienen interés en que el covid-19 no se extienda más”, we can annotate it as INTERESTED, because of the presence of the expression “tienen interés”. If X writes “A los del Partido Y les aburre el covid-19”, we can annotate it as UNINTERESTED, because of the presence of the expression “aburre” (or expressions of similar meaning).

- Inclination (inclined/disinclined). If X writes “Los del Partido Y quieren terminar con el covid-19”, we can annotate it as INCLINED, because of the presence of the verb “quieren”. If X writes “Los del Partido Y no están dispuestos a terminar con el covid-19”, we can annotate it as DISINCLINED, because of the presence of the expression “no están dispuestos” (or expressions of similar meaning).

- Security (calm/confident/trusting/confused/anxious/fearful/embarrassed/doubtful). Within the SECURITY category, we can find subcategorIes that are summarised in the examples in parenthesis. So, if X writes “Los del Partido Y están tranquilos porque el covid-19 ya se ha controlado”, we can annotate it as CALM, because of the presence of the adjective “tranquilos”. So, if X writes “Los del Partido Y están nerviosos porque el covid-19 no se ha controlado”, we can annotate it as ANXIOUS, because of the presence of the adjective“nerviosos” (or expressions of similar meaning).

NOTE

If it is not possible to identify into which of the SECURITY subcategories above the tweet falls, but we still think that it may be one of them, we can use an umbrella term such as SECURITY or INSECURITY.

- Happiness (happy/angry/sad/frustrated). If X writes “Los del Partido Y están contentos de que el covid-19 se haya extinguido”, we can annotate it as HAPPY, because of the presence of the adjective “contentos”. If X writes “A los del Partido Y les frustra cómo evoluciona covid-19”, we can annotate it as FRUSTRATED, because of the presence of the verb “frustra” (or expressions of similar meaning).

NOTE

If it is not possible to identify into which of the UNHAPPINESS subcategories above the tweet falls, but we still think that it may be one of them, we can use an umbrella term such as UNHAPPINESS.

- Liking (like/dislike). If X writes “A los del Partido Y les gusta que el covid-19 se haya extendido”, we can annotate it as LIKE, because of the presence of the verb “gusta”. If X writes “Los del Partido Y no son aficionados del covid-19”, we can annotate it as DISLIKE, because of the presence of the expression “no son aficionados” (or expressions of similar meaning).
- Love (affection/antipathy). If X writes “Los del Partido Y aman a los gays”, we can annotate it as AFFECTION, because of the presence of the verb “aman”. If X writes “Los del Partido Y odian a los gays”, we can annotate it as ANTIPATHY, because of the presence of the verb “odian” (or expressions of similar meaning).
- Respect (respect/disrespect). If X writes “Los del Partido Y respetan a todo el mundo”, we can annotate it as RESPECT, because of the presence of the verb “respetan”. If X writes “Los del Partido Y le faltan el respeto a todo el mundo”, we can annotate it as DISRESPECT, because of the presence of the expression “faltan el respeto” (or expressions of similar meaning).
- Sympathy (sympathy/indifference). If X writes “Los del Partido Y han mostrado su compasión hacia los migrantes”, we can annotate it as SYMPATHY, because of the presence of the noun “compasión”. If X

writes “A los del Partido Y les dan igual los migrantes”, we can annotate it as INDIFFERENCE, because of the presence of the expression “les dan igual” (or expressions of similar meaning).

- Tolerance (tolerance/intolerance). If X writes “Los del Partido Y toleran a otros grupos”, we can annotate it as TOLERANCE, because of the presence of the verb “toleran”. If X writes “Los del Partido Y son muy intolerantes”, we can annotate it as INTOLERANCE, because of the presence of the adjective “intolerantes” (or expressions of similar meaning).

NOTE

If it is not possible to identify which of the subcategories above the tweet falls into, but we still think that it may be one of them, we can use an umbrella term such as ATTRACTION or REPULSION (attraction encodes instances where X feels positively attracted to Y, and repulsion refers to cases where the emoter’s aversion is apparent).

- Impact (fascinating/dull). If X writes “Los discursos del líder del Partido Y me fascinan”, we can annotate it as FASCINATING, because of the presence of the verb “fascinan” (or expressions of similar meaning). If X writes “Los discursos del líder del Partido Y son un coñazo”, we can annotate it as DULL, because of the presence of the noun “coñazo” (or expressions of similar meaning). We must remember that this category tries to answer the question: DID IT GRAB ME? **NOTE: MANY OF THESE EXAMPLES ARE DERIVATIONALLY RELATED TO EMOTION LEXIS. THE ONLY DIFFERENCE BETWEEN THESE AND THE ABOVE CATEGORIES IS THAT IN THIS CASE THE FOCUS IS THE TRIGGER (e.g. This book is interesting) INSTEAD OF THE EMOTER (e.g. I am interested in this book)**. See below for more examples of the category IMPACT:

FASCINATING	DULL
arresting, captivating, engaging fascinating, exciting, moving lively, dramatic, intense remarkable, notable, sensational	dull, boring, tedious dry, ascetic, uninviting flat, predictable, monotonous unremarkable, pedestrian

- Quality (lovely/ugly). If X writes “El estadio de fútbol es precioso”, we can annotate it as LOVELY, because of the presence of the adjective "precioso" (or expressions of similar meaning). If X writes "El estadio de fútbol es muy feo", we can annotate it as UGLY, because of the presence of the adjective "feo" (or expressions of similar meaning) We must remember that this category tries to answer the question: DID I LIKE IT? DID IT INDICATE A PARTICULAR STANDARD? See below for more examples of both:

LOVELY	UGLY
clean, suitable, effective, conveniente, okay, fine, good lovely, beautiful, splendid appealing, enchanting, welcome	bad, yuk, nasty plain, ugly, grotesque repulsive, revolting, off-putting

- Balance (harmonious/discordant). If X writes "El argumento del Partido Y es totalmente lógico", we can annotate it as HARMONIOUS, because of the presence of the adjective "lógico" (or expressions of similar meaning). If X writes "El argumento del Partido Y tiene múltiples errores", we can annotate it as DISCORDANT, because of the presence of the noun phrase "múltiples errores" (or expressions of similar meaning). We must remember that this category tries to answer the question: DID IT HANG TOGETHER? See below for more examples of both:

HARMONIOUS	DISCORDANT
balanced, harmonious, unified, symmetrical, proportioned consistent, considered, logical shapely, curvaceous, willowly	unbalanced, discordant, irregular, uneven, flawed contradictory, disorganised shapeless, amorphous, distorted

- Complexity (simple/complicated). If X writes "Es muy fácil de seguir el discurso del líder del Partido Y", we can annotate it as SIMPLE, because of the presence of the adjective "fácil" (or expressions of similar meaning). If X writes "El discurso del líder del Partido Y es un jaleo", we can annotate it as COMPLICATED, because of the presence of the noun "jaleo" (or expressions of similar meaning). We must remember that this category tries to answer the question: WAS IT HARD TO FOLLOW? See below for more examples of both:

SIMPLE	COMPLICATED

simple, pure, elegant	ornate, extravagant, byzantine
lucid, clear, precise	arcane, unclear, woolly
intricate, rich, detailed, precise	plain, monolithic, simplistic

- Significance (important/marginal). If X writes “El Partido Y es clave en la historia del país”, we can annotate it as IMPORTANT, because of the presence of the adjective “clave” (or expressions of similar meaning). If X writes “El Partido Y no pinta nada en historia del país”, we can annotate it as MARGINAL, because of the presence of the verb phrase “no pinta” (or expressions of similar meaning). We must remember that this category tries to answer the question: WAS IT IMPORTANT? See below for more examples of both:

IMPORTANT	MARGINAL
significant, important, notable, vital, critical, momentous, noteworthy	peripheral, secondary, minor, irrelevant, unimportant, incidental

- Benefit (beneficial/destructive). If X writes “La inacción del Partido Y está matando al país”, we can annotate it as DESTRUCTIVE, because of the presence of the noun “inacción” (or expressions of similar meaning). We must remember that this category tries to answer the question: DID IT ENHANCE OR DESTROY? See below for more examples of both:

BENEFICIAL	DESTRUCTIVE
beneficial, useful, helpful, advantageous, benign, expedient, effective	dangerous, threatening, risky, alarming, hazardous, insecure

- Propriety (good/bad). If X writes “Los del Partido Y son buena gente”, we can annotate it as GOOD, because of the presence of the noun phrase “buena gente” (or expressions of similar meaning). If X writes “Los del Partido Y son un ejemplo de corrupción”, we can annotate it as BAD , because of the presence of the noun phrase “un ejemplo de corrupción” (or expressions of similar meaning). We must remember that this category tries to answer the question: HOW FAR BEYOND REPROACH? See below for more examples of both:

GOOD	BAD
good, moral, ethical	bad, immoral, evil
law abiding, fair, just	corrupt, unfair, unjust
sensitive, kind, caring	insensitive, mean, cruel
unassuming, modest, humble	vain, snobby, arrogant
polite, respectful, reverent	rude, discourteous, irreverent
altruistic, generous, charitable	selfish, greedy, avaricious

- Veracity (honest/deceitful). If X writes “Los del Partido Y nunca nos mienten”, we can annotate it as HONEST, because of the presence of the expression “nunca nos mienten” (or expressions of similar meaning). If X writes “Los del Partido Y son unos mentirosos”, we can annotate it as DECEITFUL, because of the presence of the noun “mentirosos” (or expressions of similar meaning). We must remember that this category tries to answer the question: HOW HONEST? See below for more examples of both:

HONEST	DECEITFUL
truthful, honest, credible	dishonest, deceitful, lying
frank, candid, direct	deceptive, manipulative, devious
discrete, tactful	blunt, blabbermouth

- Normality (normal/abnormal). If X writes “Lo normal es que el Partido Y gane las elecciones”, we can annotate it as NORMAL, because of the presence of the noun phrase “lo normal” (or expressions of similar meaning; see below). If X writes “Los del Partido Y han tenido la buena suerte de no tener que lidiar con el covid-19”, we can annotate it as ABNORMAL, because of the presence of the verb “ha tenido la buena suerte” (or expressions of similar meaning).We must remember that this category tries to answer the question: HOW SPECIAL? See below for more examples of both:

NORMAL	ABNORMAL
lucky, fortunate, charmed	unlucky, hapless, star-crossed
normal, natural, familiar	odd, peculiar, eccentric
cool, stable, predictable	erratic, unpredictable
fashionable, avant garde	dated, daggy, retrograde
celebrated, unsung	obscure, also-ran

- Capacity (skilled/clumsy). If X writes “Los del Partido Y serán capaces de parar el covid-19”, we can annotate it as SKILLED, because of the presence of the adjective “capaces” (or expressions of similar meaning). If X writes “Los del Partido Y han mostrado su incapacidad para parar el covid-19”, we can annotate it as CLUMSY, because of the presence of the noun “incapacidad” (or expressions of similar meaning). We must remember that this category tries to answer the question: HOW CAPABLE? See below for more examples of both:

SKILLED	CLUMSY
powerful, vigorous, robust	mild, weak, whimpy
sound, healthy, fit	unsound, sick, crippled
adult, mature, experienced	immature, childish, helpless
witty, humorous, droll	dull, dreary, grave
insightful, clever, gifted	slow, stupid, thick
balanced, together, sane	flaky, neurotic, insane
sensible, expert, shrewd	naive, inexpert, foolish
literate, educated, learned	illiterate, uneducated, ignorant
competent, accomplished	incompetent; unaccomplished
successful, productive	unsuccessful, unproductive

- Tenacity (brave/cowardly). If X writes “España está demostrando que es muy valiente durante el confinamiento”, we can annotate it as BRAVE, because of the presence of the adjective “valiente” (or expressions of similar meaning). If X writes “Los del Partido Y son unos cobardicas”, we can annotate it as COWARDLY, because of the presence of the adjective “cobardicas” (or expressions of similar meaning). We must remember that this category tries to answer the question: HOW DEPENDABLE? See below for more examples of both:

BRAVE	COWARDLY
plucky, brave, heroic	timid, cowardly, gutless
cautious, wary, patient	rash, impatient, impetuous
careful, thorough, meticulous	weak, distracted, despondent
hasty, capricious, reckless	unreliable, undependable
tireless, persevering, resolute	unfaithful, disloyal, inconstan
reliable, dependable	stubborn, obstinate, wilful
faithful, loyal, constant	
flexible, adaptable, accommodating	

Communication and Discussion

Currently, we use Slack for team communication (both private and group communications). We invite you to join (using the app or the website) so you can discuss with your teammates any doubt you may have. You can do it with the following link:
<removed>

Remember that you cannot share this link with anyone who do not belong to the Nutcracker Project. We strongly suggest you to use the app properly (use channels for specific purposes and private chats to keep conversations that are not relevant to the topic/users in the channels). You can open new channels if necessary. We may moderate the channels.

If we want the app to be useful for all of us, we need to use it properly. Take into consideration that it is easy to miss a message, so do not use it as a substitute for formal emails. We encourage you not to use channels as a personal/team repository. If you are not familiar with Slack, there are plenty of materials on how to use it. You can start with the following short video: <removed>

Disclaimer

The annotating process is subject to interpretation and it is biased by your experience. Normally, we will ask several experts to tag the same content several times, in order to aggregate the results to delimit the bias. However, the number of tweets is several times bigger than the number of experts. We do not have the resources to annotate each tweet several times. We kindly ask you to rely on the textual evidence as much as possible so that there is as less little researcher bias as possible..

The content of this document, the Nutcracker Tagging Tool, tweet databases, users information and the use that can be made of them is strictly restricted to members of the Nutcracker project and we may monitor your interactions and transmissions to ensure it. We may use cookies. Accessing the website without the proper authorisation is prohibited. You explicitly accept this disclaimer when logging in. If you encounter any problems when using this site, please, contact Manuel Francisco <removed> for support.

Acknowledgment

This research is financially supported by the Spanish Ministry of Economy and Competitiveness (MINECO), project FFI2016-79748-R, and co-funded by the European Social Fund (ESF). The author is supported by the FPI 2017 predoctoral programme, from the Spanish Ministry of Economy and Competitiveness (MINECO), grant number BES-2017-081202.

B Description of Label Sets

<political party> Possible values: *negative, positive*
Help text: *If the document is negative or positive towards this political party.*

binary gender Possible values: *female, male, other*
Help text: *In this case, we will take into account the name of the user (e.g. "Carmen Fernández" will be annotated as FEMALE) or how they describe themselves (e.g. "Soy un hombre comprometido" will be annotated as MALE). Profile pictures may be confusing and misleading.*

age Possible values: *13-17, 18-24, 25-34, 35-44, 45-54, 55-64, 65+*
Help text: *How old the user is. Try to find evidence.*

sentence type Possible values: *declarative, imperative, interrogative, exclamative*
Help text: *Declarative sentences make a statement. They tell us something and give us some information about a particular state of affairs. Normally, the usual word order for the declarative sentence is Subject + Predicator. Declarative sentences can appear both in assertive and non-assertive contexts: "Manolo did not go to the library yesterday", "Óscar studied a lot". Interrogative sentences ask a question. They are used to get information from the addressee, and always end with a question mark. The usual word order for the interrogative sentence is as follows: (Wh-word +) Auxiliary verb+ Subject + Main verb...? Interrogative sentences can appear both in assertive and non-assertive contexts: "Will Katie come over in September?","Why didn't Encarni explain that concept in the meeting?"*
Imperative sentences are used to order your addressee to do, or not to do, something. They tell us to do something, and they end with a full-stop/period (.) or exclamation mark/point (!). The usual word order for the imperative sentence is as follows: base verb.... There is usually no Subject in this type of structures,

because we understand that the Subject the second person pronoun. They can also appear both in assertive and non-assertive contexts: "Javi, don't go to the Faculty now", "Come back, Miguel Ángel".

Exclamative sentences express strong emotion; they generally end with an exclamation mark (!). The usual word order for the exclamative sentence is the following: "What (+ adjective) + noun + Subject + Predicator!", "How (+adjective/adverb) + Subject + Predicator!": "What a great musician Pascual is!", "How interesting Azzam" story was!

speech act Possible values: *representative, directive, commissive, expressive, declaration, verdictive*

Help text: *Representatives: assertions, statements, claims, hypotheses, descriptions, suggestions.*

Commissives: promises, oaths, pledges, threats, vows.

Directives: commands, requests, challenges, invitations, orders, summons, entreaties, dares.

Declarations: blessings, firings, baptisms, arrests, marrying, juridial speech acts such as sentencings, declaring a mistrial, declaring s.o.out of order, etc.

Expressives are speech acts that make assessments of psychological states or attitudes (e.g., greetings, apologies, congratulations, condolences, thanksgivings)

Verdictives are speech acts in which the speaker makes an assessment or judgement about the acts of another, usually the addressee. These include ranking, assessing, appraising, condoning. Verdictive verbs include accuse, charge, excuse, thank...

pragmatic function Possible values: *literal, metaphorical, ironic, sarcastic, rhetorical question, hyperbole*

Help text: *Given that the challenge here is perhaps whether one text is ironic or not, we will pay attention to how hyperbole (e.g. "Es listísimo"), opposition (e.g. "Cuánto lo amo") and understatement (e.g. "Es algo malo") operate in context so that there is some incongruence between what the intended meaning is and what the user has actually written. Although sarcasm is a harshsharp form of irony meant to ridicule someone, we can include it here under the umbrella term "pragmatic function" in those cases when the author clearly aims to hurt someone else. Finally, let's indicate as well those tweets that are metaphorical. We will analyse them manually later.*

Hyperbole is the use of exaggeration as a rhetorical device or figure of speech.

In rhetoric, it is also sometimes known as auxesis. In poetry and oratory, it emphasises, evokes strong feelings, and creates strong impressions. As a figure of speech, it is usually not meant to be taken literally.

mood Possible values: *irrealis, realis*
Help text: *It applies to the main clause.*
Realis: Factual information, plain statements, statements of fact (Something has happened Something has not happened (both assertive and non-assertive).
Irrealis: A certain situation or action is not known to have happened at the moment the speaker is talking Something that hasn't yet occurred.
TO CUT A LONG STORY SHORT: Irrealis would apply to any: i) imperative and interrogative clause; ii) any main clause containing an explicitimplicit epistemic or deontic marker (will, may, can, think, guess, believe, should, must, have to, etc.); iii) mental processes which by their very nature are irrealis (e.g. desiderative: want, would like to, etc.); and iv) conditional clauses (if, unless...).

document sentiment Possible values: *negative, neutral, positive*
Help text: *Overall sentiment of the tweet; in other words, we will annotate the opinions expressed in a of text, especially in order to determine whether the writer's attitude towards a topic, a particular product, an individual, an institution, an event, someone's attitudes, etc., is positive, negative, or neutral. We must remember that the annotator's belief systems are not relevant here. That is, to kill Muslims or Christians may be something good from the perspective of those who hate Muslims or Christians, even though from the perspective of a normal human being, in general, killing is a wrongful act (unless what we kill is a virus like COVID-19).*

implicit connotations Possible values: *none, present*
Help text: *Sometimes one tweet will clearly be interpreted explicitly as one particular emotion category and implicitly as another opinion category, or the other way around: it can be interpreted explicitly as one particular opinion category and implicitly as one emotion category. In those instances where the example shows a two-tiered opinion-emotion annotation, the coder will indicate the presence of IMPLICIT CONNOTATION and subsequently tag both, the emotion and the opinion, or the other way around.*

type of trigger Possible values: *inanimate trigger, animate trigger*
Help text: *This category refers to the entity that is causing some emotional reaction.*

Attention-grabbing Possible values: *no, yes*
Help text: *The following is a general category, just in case you cannot annotate specific ones. Leave blank otherwise.*

surprise Possible values: *no, yes*
Help text: *If X writes "A los del Partido Y les sorprende que el covid-19 se haya extendido", we can annotate it as SURPRISE, because of the presence of the verb "sorprende" (or expressions of similar meaning).*

interest Possible values: *uninterested, interested*
Help text: *If X writes "Los del Partido Y tienen interés en que el covid-19 no se extienda más", we can annotate it as INTERESTED, because of the presence of the expression "tienen interés". If X writes "A los del Partido Y les aburre el covid-19", we can annotate it as UNINTERESTED, because of the presence of the expression "aburre" (or expressions of similar meaning).*

inclination Possible values: *disinclined, inclined*
Help text: *If X writes "Los del Partido Y quieren terminar con el covid-19", we can annotate it as INCLINED, because of the presence of the verb "quieren". If X writes "Los del Partido Y no están dispuestos a terminar con el covid-19", we can annotate it as DISINCLINED, because of the presence of the expression "no están dispuestos" (or expressions of similar meaning).*

Satisfaction Possible values: *no, yes*
Help text: *The following is a general category, just in case you cannot annotate specific ones. Leave blank otherwise.*

security Possible values: *calm, confident, trusting, confused, anxious, fearful, embarrassed, doubtful, distrustful*

Help text: *Within the SECURITY category, we can find subcategorIes that are summarised in the examples in parenthesis. So, if X writes "Los del Partido Y están tranquilos porque el covid-19 ya se ha controlado", we can annotate it as CALM, because of the presence of the adjective "tranquilos". So, if X writes "Los del Partido Y están nerviosos porque el covid-19 no se ha controlado", we can annotate it as ANXIOUS, because of the presence of the adjective"nerviosos" (or expressions of similar meaning).*

happiness Possible values: *sad, happy, angry, frustrated*

Help text: *If X writes "Los del Partido Y están contentos de que el covid-19 se haya extinguido", we can annotate it as HAPPY, because of the presence of the adjective "contentos". If X writes "A los del Partido Y les frustra cómo evoluciona covid-19", we can annotate it as FRUSTRATED, because of the presence of the verb "frustra" (or expressions of similar meaning).*

Goal-relation emotions Possible values: *repulsion, attraction*

Help text: *The following is a general category, just in case you cannot annotate specific ones. Leave blank otherwise.*

liking Possible values: *dislike, like*

Help text: *If X writes "A los del Partido Y les gusta que el covid-19 se haya extendido", we can annotate it as LIKE, because of the presence of the verb "gusta". If X writes "Los del Partido Y no son aficionados del covid-19", we can annotate it as DISLIKE, because of the presence of the expression "no son aficionados" (or expressions of similar meaning).*

love Possible values: *antipathy, affection*

Help text: *If X writes "Los del Partido Y aman a los gays", we can annotate it as AFFECTION, because of the presence of the verb "aman". If X writes "Los del Partido Y odian a los gays", we can annotate it as ANTIPATHY, because of the presence of the verb "odian" (or expressions of similar meaning).*

respect Possible values: *disrespect, respect*

Help text: *If X writes "Los del Partido Y respetan a todo el mundo", we can annotate it as RESPECT, because of the presence of the verb "respetan". If X writes "Los del Partido Y le faltan el respeto a todo el mundo", we can annotate it as DISRESPECT, because of the presence of the expression "faltan el respeto" (or expressions of similar meaning).*

sympathy Possible values: *indifference, sympathy*

Help text: *If X writes "Los del Partido Y han mostrado su compasión hacia los migrantes", we can annotate it as SYMPATHY, because of the presence of the noun "compasión". If X writes "A los del Partido Y les dan igual los migrantes", we can annotate it as INDIFFERENCE, because of the presence of the expression "les dan igual" (or expressions of similar meaning).*

tolerance Possible values: *intolerance, tolerance*

Help text: *If X writes "Los del Partido Y toleran a otros grupos", we can annotate it as TOLERANCE, because of the presence of the verb "toleran". If X writes "Los del Partido Y son muy intolerantes", we can annotate it as INTOLERANCE, because of the presence of the adjective "intolerantes" (or expressions of similar meaning).*

type of appraised Possible values: *inanimate appraised, animate appraised*

Help text: *This category refers to the entity that is assessed either ethically or aesthetically.*

Appreciation Possible values: *no, yes*

Help text: *The following is a general category, just in case you cannot annotate specific ones. Leave blank otherwise.*

impact Possible values: *dull, fascinating*

Help text: *If X writes "Los discursos del líder del Partido Y me fascinan", we can annotate it as FASCINATING, because of the presence of the verb "fascinan" (or expressions of similar meaning). If X writes "Los discursos del líder del Partido*

Y son un coñazo", we can annotate it as DULL, because of the presence of the noun "coñazo" (or expressions of similar meaning). We must remember that this category tries to answer the question: did it grab me?

quality Possible values: *ugly, lovely*
Help text: *If X writes "El estadio de fútbol es precioso", we can annotate it as LOVELY, because of the presence of the adjective "precioso" (or expressions of similar meaning). If X writes "El estadio de fútbol es muy feo", we can annotate it as UGLY, because of the presence of the adjective "feo" (or expressions of similar meaning) We must remember that this category tries to answer the question: did I like it? Did id indicate a particular standard?*

balance Possible values: *discordant, harmonious*
Help text: *If X writes "El argumento del Partido Y es totalmente lógico", we can annotate it as HARMONIOUS, because of the presence of the adjective "lógico" (or expressions of similar meaning). If X writes "El argumento del Partido Y tiene múltiples errores", we can annotate it as DISCORDANT, because of the presence of the noun phrase "múltiples errores" (or expressions of similar meaning). We must remember that this category tries to answer the question: Did it hang together?*

complexity Possible values: *complicated, simple*
Help text: *If X writes "Es muy fácil de seguir el discurso del líder del Partido Y", we can annotate it as SIMPLE, because of the presence of the adjective "fácil" (or expressions of similar meaning). If X writes "El discurso del líder del Partido Y es un jaleo", we can annotate it as COMPLICATED, because of the presence of the noun "jaleo" (or expressions of similar meaning). We must remember that this category tries to answer the question: was it hard to follow?*

significance Possible values: *marginal, important*
Help text: *If X writes "El Partido Y es clave en la historia del país", we can annotate it as IMPORTANT, because of the presence of the adjective "clave" (or expressions of similar meaning). If X writes "El Partido Y no pinta nada en historia del país", we can annotate it as MARGINAL, because of the presence of the verb phrase "no pinta" (or expressions of similar meaning). We must remember that this category tries to answer the question: was it important?*

benefit Possible values: *destructive, beneficial*
Help text: *If X writes "La inacción del Partido Y está matando al país", we can annotate it as DESTRUCTIVE, because of the presence of the noun "inacción" (or expressions of similar meaning). We must remember that this category tries to answer the question: did it enhance or destroy?*

Social Sanction Possible values: *no, yes*
Help text: *The following is a general category, just in case you cannot annotate specific ones. Leave blank otherwise.*

propriety Possible values: *bad, good*
Help text: *If X writes "Los del Partido Y son buena gente", we can annotate it as GOOD, because of the presence of the noun phrase "buena gente" (or expressions of similar meaning). If X writes "Los del Partido Y son un ejemplo de corrupción", we can annotate it as BAD , because of the presence of the noun phrase "un ejemplo de corrupción" (or expressions of similar meaning). We must remember that this category tries to answer the question: how far beyond reproach?*

veracity Possible values: *deceitful, honest*
Help text: *If X writes "Los del Partido Y nunca nos mienten", we can annotate it as HONEST, because of the presence of the expression "nunca nos mienten" (or expressions of similar meaning). If X writes "Los del Partido Y son unos mentirosos", we can annotate it as DECEITFUL, because of the presence of the noun "mentirosos" (or expressions of similar meaning). We must remember that this category tries to answer the question: how honest?*

Social Esteem Possible values: *no, yes*
Help text: *The following is a general category, just in case you cannot annotate specific ones. Leave blank otherwis*

normality Possible values: *abnormal, normal*
Help text: *If X writes "Lo normal es que el Partido Y gane las elecciones", we can annotate it as NORMAL, because of the presence of the noun phrase "lo normal" (or expressions of similar meaning; see below). If X writes "Los del Partido Y han tenido la buena suerte de no tener que lidiar con el covid-19", we can annotate it*

as ABNORMAL, because of the presence of the verb "ha tenido la buena suerte" (or expressions of similar meaning).We must remember that this category tries to answer the question: how special?

capacity Possible values: *incapable, capable*

Help text: *If X writes "Los del Partido Y serán capaces de parar el covid-19", we can annotate it as CAPABLE, because of the presence of the adjective "capaces" (or expressions of similar meaning). If X writes "Los del Partido Y han mostrado su incapacidad para parar el COVID-19", we can annotate it as INCAPABLE, because of the presence of the noun "incapacidad" (or expressions of similar meaning). We must remember that this category tries to answer the question: how capable?*

tenacity Possible values: *cowardly, brave*

Help text: *If X writes "España está demostrando que es muy valiente durante el confinamiento", we can annotate it as BRAVE, because of the presence of the adjective "valiente" (or expressions of similar meaning). If X writes "Los del Partido Y son unos cobardicas", we can annotate it as COWARDLY, because of the presence of the adjective "cobardicas" (or expressions of similar meaning). We must remember that this category tries to answer the question: how dependable?*

C Frequent itemset mining: Association Rules

Tab. C.1.: Political parties association rules (minimun support 0.5).

antecedents	consequents	ant.	consq.	supp.	conf.	lift	lvrg.	conv.
		supp.	supp.					
■ auto PP: negative ■ auto PSOE: negative ■ auto VOX: negative	■ auto Cs: negative	.514	.713	.510	.993	1.392	.144	40.678
■ auto VOX: negative ■ auto PSOE: negative	■ auto Cs: negative	.525	.713	.520	.991	1.389	.145	30.370
■ auto PP: negative ■ auto PSOE: negative	■ auto Cs: negative	.624	.713	.617	.990	1.388	.173	28.871
■ auto PP: negative ■ auto VOX: negative	■ auto Cs: negative	.569	.713	.564	.990	1.388	.157	28.173
■ auto PP: negative	■ auto Cs: negative	.683	.713	.673	.986	1.382	.186	19.794
■ auto VOX: negative ■ auto Cs: negative ■ auto PSOE: negative	■ auto PP: negative	.520	.683	.510	.982	1.438	.155	17.252
■ auto VOX: negative ■ auto PSOE: negative	■ auto PP: negative	.525	.683	.514	.979	1.434	.156	15.286
■ auto VOX: negative	■ auto Cs: negative	.595	.713	.583	.981	1.376	.159	15.181
■ auto VOX: negative ■ auto PSOE: negative	■ auto PP: negative ■ auto Cs: negative	.525	.673	.510	.972	1.445	.157	11.829
■ auto PSOE: negative	■ auto Cs: negative	.655	.713	.638	.974	1.365	.171	10.941
■ auto Cs: negative ■ auto PSOE: negative	■ auto PP: negative	.638	.683	.617	.969	1.419	.182	10.085
■ auto VOX: negative ■ auto Cs: negative	■ auto PP: negative	.583	.683	.564	.966	1.415	.165	9.333
■ auto VOX: negative	■ auto PP: negative	.595	.683	.569	.958	1.402	.163	7.466
■ auto PSOE: negative	■ auto PP: negative	.655	.683	.624	.953	1.395	.177	6.696
■ auto VOX: negative	■ auto PP: negative ■ auto Cs: negative	.595	.673	.564	.948	1.409	.163	6.263
■ auto PSOE: negative	■ auto PP: negative ■ auto Cs: negative	.655	.673	.617	.943	1.402	.177	5.756

Continued on next page

Tab. C.1.: Political parties association rules (minimun support 0.5).

antecedents	consequents	ant.	consq.	supp.	conf.	lift	lvrg.	conv.
		supp.	supp.					
■ auto Cs: negative	■ auto PP: negative	.713	.683	.673	.943	1.382	.186	5.598
■ auto PP: negative ■ auto Cs: negative	■ auto PSOE: negative	.673	.655	.617	.918	1.402	.177	4.196
■ auto PP: negative	■ auto PSOE: negative	.683	.655	.624	.913	1.395	.177	3.991
■ auto PP: negative	■ auto Cs: negative ■ auto PSOE: negative	.683	.638	.617	.904	1.419	.182	3.792
■ auto PP: negative ■ auto Cs: negative ■ auto VOX: negative	■ auto PSOE: negative	.564	.655	.510	.905	1.383	.141	3.645
■ auto PP: negative ■ auto VOX: negative	■ auto PSOE: negative	.569	.655	.514	.902	1.378	.141	3.539
■ auto PP: negative ■ auto VOX: negative	■ auto Cs: negative ■ auto PSOE: negative	.569	.638	.510	.896	1.406	.147	3.487
■ auto Cs: negative	■ auto PSOE: negative	.713	.655	.638	.894	1.365	.171	3.252
■ auto VOX: negative ■ auto Cs: negative	■ auto PSOE: negative	.583	.655	.520	.891	1.361	.138	3.165
■ auto VOX: negative ■ auto Cs: negative	■ auto PP: negative ■ auto PSOE: negative	.583	.624	.510	.874	1.402	.146	2.999
■ auto VOX: negative	■ auto PSOE: negative	.595	.655	.525	.882	1.348	.135	2.936
■ auto VOX: negative	■ auto Cs: negative ■ auto PSOE: negative	.595	.638	.520	.874	1.371	.141	2.878
■ auto Cs: negative	■ auto PP: negative ■ auto PSOE: negative	.713	.624	.617	.866	1.388	.173	2.802
■ auto VOX: negative	■ auto PP: negative ■ auto PSOE: negative	.595	.624	.514	.864	1.386	.143	2.769
■ auto VOX: negative	■ auto PP: negative ■ auto Cs: negative ■ auto PSOE: negative	.595	.617	.510	.858	1.390	.143	2.694
■ auto PP: negative ■ auto Cs: negative	■ auto VOX: negative	.673	.595	.564	.838	1.409	.163	2.495
■ auto PP: negative	■ auto VOX: negative	.683	.595	.569	.834	1.402	.163	2.441
■ auto PP: negative	■ auto VOX: negative ■ auto Cs: negative	.683	.583	.564	.825	1.415	.165	2.386
■ auto PP: negative ■ auto Cs: negative ■ auto PSOE: negative	■ auto VOX: negative	.617	.595	.510	.826	1.390	.143	2.333
■ auto PP: negative ■ auto PSOE: negative	■ auto VOX: negative	.624	.595	.514	.824	1.386	.143	2.301
■ auto PP: negative ■ auto PSOE: negative	■ auto VOX: negative ■ auto Cs: negative	.624	.583	.510	.818	1.402	.146	2.289
■ auto Cs: negative	■ auto VOX: negative	.713	.595	.583	.818	1.376	.159	2.226

Continued on next page

Tab. C.1.: Political parties association rules (minimun support 0.5).

antecedents	consequents	ant.	consq.	supp.	conf.	lift	lvrg.	conv.
		supp.	supp.					
■ auto Cs: negative ■ auto PSOE: negative	■ auto VOX: negative	.638	.595	.520	.815	1.371	.141	2.194
■ auto Cs: negative ■ auto PSOE: negative	■ auto PP: negative ■ auto VOX: negative	.638	.569	.510	.800	1.406	.147	2.156
■ auto Cs: negative	■ auto PP: negative ■ auto VOX: negative	.713	.569	.564	.790	1.388	.157	2.051
■ auto PSOE: negative	■ auto VOX: negative	.655	.595	.525	.801	1.348	.135	2.041
■ auto PSOE: negative	■ auto VOX: negative ■ auto Cs: negative	.655	.583	.520	.794	1.361	.138	2.021
■ auto PSOE: negative	■ auto PP: negative ■ auto VOX: negative	.655	.569	.514	.785	1.378	.141	2.001
■ auto PSOE: negative	■ auto PP: negative ■ auto Cs: negative ■ auto VOX: negative	.655	.564	.510	.779	1.383	.141	1.977
■ auto PP: negative ■ auto Cs: negative	■ auto VOX: negative ■ auto PSOE: negative	.673	.525	.510	.758	1.445	.157	1.966
■ auto PP: negative	■ auto VOX: negative ■ auto PSOE: negative	.683	.525	.514	.753	1.434	.156	1.921
■ auto PP: negative	■ auto VOX: negative ■ auto Cs: negative ■ auto PSOE: negative	.683	.520	.510	.747	1.438	.155	1.900
■ auto Cs: negative	■ auto VOX: negative ■ auto PSOE: negative	.713	.525	.520	.729	1.389	.145	1.752
■ auto Cs: negative	■ auto PP: negative ■ auto PSOE: negative ■ auto VOX: negative	.713	.514	.510	.715	1.392	.144	1.707