CAMPUS
DE EXCELENCIA
INTERNACIONAL

Graduado en Matemáticas e Informática

Universidad Politécnica de Madrid

Facultad de Informática

TRABAJO FIN DE GRADO

Presencia en Twitter de los candidatos a las elecciones madrileñas de 2015

Autor: Ignacio Amaya de la Peña

Director: Alfonso Mateos Caballero

MADRID, JUNIO DE 2015

Índice general

Índice de figuras

Índice de tablas

Resumen

Durante los últimos años ha aumentado la presencia de personas pertenecientes al mundo de la política en la red debido a la proliferación de las redes sociales, siendo Twitter la que mayor repercusión mediática tiene en este ámbito. El estudio del comportamiento de los políticos en Twitter y de la acogida que tienen entre los ciudadanos proporciona información muy valiosa a la hora de analizar las campañas electorales. De esta forma, se puede estudiar la repercusión real que tienen sus mensajes en los resultados electorales, así como distinguir aquellos comportamientos que tienen una mayor aceptación por parte de la la ciudadanía.

Gracias a los avances desarrollados en el campo de la minería de textos, se poseen las herramientas necesarias para analizar un gran volumen de textos y extraer de ellos información de utilidad.

Este proyecto tiene como finalidad recopilar una muestra significativa de mensajes de Twitter pertenecientes a los candidatos de los principales partidos políticos que se presentan a las elecciones autonómicas de Madrid en 2015. Estos mensajes, junto con las respuestas de otros usuarios, se han analizado usando algoritmos de aprendizaje automático y aplicando las técnicas de minería de textos más oportunas.

Los resultados obtenidos para cada político se han examinado en profundidad y se han presentado mediante tablas y gráficas para facilitar su comprensión.

Abstract

During the past few years the presence on the Internet of people related with politics has increased, due to the proliferation of social networks. Among all existing social networks, Twitter is the one which has the greatest media impact in this field. Therefore, an analysis of the behaviour of politicians in this social network, along with the response from the citizens, gives us very valuable information when analysing electoral campaigns. This way it is possible to know their messages impact in the election results. Moreover, it can be inferred which behaviours have better acceptance among the citizenship.

Thanks to the advances achieved in the text mining field, its tools can be used to analyse a great amount of texts and extract from them useful information.

The present project aims to collect a significant sample of Twitter messages from the candidates of the principal political parties for the 2015 autonomic elections in Madrid. These messages, as well as the answers received by the other users, have been analysed using machine learning algorithms and applying the most suitable data mining techniques.

The results obtained for each politician have been examined in depth and have been presented using tables and graphs to make its understanding easier.

Capítulo 1

Introducción

Este proyecto se centra en el análisis de la comunicación política en redes sociales aplicando técnicas de minería de textos. Concretamente, la red social escogida fue Twitter.

Se prentende obtener información acerca del comportamiento y nivel de actividad de los principales candidatos a las elecciones autonómicas de Madrid de 2015. Esta información se obtendrá a partir del estudio de sus perfiles de Twitter y sus mensajes, así como de la interacción que realizan con el resto de usuarios de esta red social. Debido al gran volumen de mensajes que hay que analizar, será necesario diseñar herramientas que automaticen esta tarea. Para ello se aprovecharán los últimos avances en minería de textos.

El estudio se ha realizado a partir de las cuentas personales de los candidatos. Se ha escogido esta aproximación, en vez de centrarnos en las cuentas de los partidos, ya que últimamente se aprecia una tendencia hacia la personalización en la política (lo que se conoce como *americanización* de las campañas electorales [1][2]). Esto ha provocado que los partidos políticos adopten nuevas técnicas para llegar a los ciudadanos, muchas de ellas basadas en el uso de las redes sociales. De esta forma, los perfiles de los candidatos y su correcta gestión permiten dar la imagen de un trato más cercano al ciudadano y también fomentan la interacción entre el político y el votante.

Sin embargo, la presencia de los candidatos en Twitter puede ser contraproducente, ya que de gestionarse mal el perfil podría repercutir negativamente en el electorado. Los perfiles que sean creados con la única intención de hacer acto de presencia podrían provocar un descontento mayor que las ausencias, ya que los beneficios de las redes sociales comienzan a hacerse patentes cuando se deja a un lado la unidireccionalidad y aumenta la interación con los otros usuarios [3].

En las elecciones andaluzas de 2012 se realizó un estudio [4] de algunos de los *tweets* de los principales candidatos. En [4] se realiza un análisis descriptivo de los mensajes analizados para obtener información acerca de las técnicas de comuni-

cación que utilizaban. En el presente proyecto se ha realizado un análisis en mayor profundidad de estos mensajes, puesto que se han empleado herramientas de minería de textos para poder analizar un volumen mayor de mensajes. Además, este nuevo enfoque permite también estudiar la acogida que reciben los candidatos, analizando las respuestas del resto de los usuarios.

Con este estudio se pretende determinar qué comportamientos en Twitter ofrecen mejores resultados. También se persigue analizar a partir de las reacciones por parte del electorado si el estudio de las cuentas de Twitter es útil a la hora de predecir los resultados de las elecciones. Trabajos previos sobre el tema aseveran que el número de menciones de un candidato y las discusiones de los usuarios en Twitter son un reflejo del panorama político fuera de las redes sociales [5].

1.1 Objetivos

El objetivo general de este trabajo es extraer conocimiento de Twitter acerca de las elecciones autonómicas de Madrid de 2015 mediante el desarrollo de herramientas centradas en el análisis automático o semiautomático de textos.

Además, también se tienen los siguientes objetivos específicos que permiten poder lograr el objetivo general:

- Conocer el nivel de actividad e implicación en la red social Twitter por parte de los distintos candidatos de los partidos mayoritarios de cara a las elecciones madrileñas del 24 de mayo de 2015.
- Realizar una comparativa entre los candidatos en función del tipo de *tweets* publicados en la red social y de cómo enfocan su uso.
- Establecer los temas prioritarios entre los distintos candidatos.
- Estudiar el flujo de comunicación entre ciudadanos y políticos a través de la red social.
- Determinar las posibles ventajas e inconvenientes de las distintas técnicas usadas en Twitter por los candidatos e intentar inferir posibles consecuencias en los resultados de las elecciones.

1.2 Estructura del trabajo

El presente trabajo está organizado en torno a cinco capítulos:

- **Capítulo 1:** Se explica el contexto en el cual se ha desarrollado este trabajo,

se describen sus objetivos y se expone la estructura seguida.

- **Capítulo 2:** Se muestran todos aquellos fundamentos en los que se basa el trabajo. En el caso de los temas con contenido técnico se realiza un breve desarrollo teórico para introducir al lector en la materia.

- **Capítulo 3:** Se desarrollan todas las técnicas aplicadas y decisiones tomadas para poder conseguir los objetivos marcados. Esto incluye las decisiones de diseño de nuestras herramientas de análisis de textos y la descripción de las medidas de error consideradas para la evaluación de los resultados.

- **Capítulo 4:** Se presentan los resultados obtenidos del análisis de los *tweets* de los candidatos de los principales partidos políticos en las elecciones autonómicas del 24 de mayo de 2015.

- **Capítulo 5:** Se indican las conclusiones relevantes extraídas y se sugieren algunas mejoras, de forma que quedan abiertas futuras líneas de investigación sobre el tema para aquellos que se encuentren interesados en profundizar más.

Capítulo 2

Fundamentos

En este capítulo se introducen todos aquellos conceptos necesarios para comprender el trabajo realizado.

2.1 Red social Twitter

Actualmente, una gran parte de la información que se genera en Internet parte de la llamada Web 2.0. Este término se usa para designar aquellas páginas que prestan especial atención a los usuarios, ya que estos son considerados los principales creadores de contenido. Estas páginas centran sus esfuerzos en proporcionar un fácil acceso e intentan que la comunicación e interacción entre ellos sea lo más fluida y sencilla posible. Algunos ejemplos de este fenómeno son los blogs, Youtube, Facebook o Twitter.

De entre todas las redes sociales Twitter es la que ha adquirido una mayor relevancia en el panorama político actual. Sus principales características son la brevedad y la rapidez a la hora de escribir mensajes, lo que favorece que se publique un gran número de mensajes al día. Twitter se ha convertido en un espacio *online* de debate alrededor de multitud de temas. Su estructura horizontal provoca que los usuarios se sientan más cercanos a los políticos y famosos, lo cual favorece la comunicación entre ambas partes.

Formalmente, Twitter es una red de *microblogging*, donde el tamaño de los mensajes, llamados *tweets*, está limitado a 140 caracteres. Los usuarios pueden subscribirse a los *tweets* de otros usuario. A este conjunto de usuarios subscritos se los denomina *followers*. Estos también tienen la opción de compartir un *tweet* de otro usuario con todos sus *followers*, lo que se conoce como *retweet*. También se puede marcar un *tweet* como favorito, pero en este caso el mensaje no se comparte. Los nombres de todos los usuarios comienzan por el símbolo @ y se puede escribir un *tweet* dirigido a otro usuario simplemente mencionándolo en el mensaje. Este mensaje será público

para todos los usuarios de Twitter, pero el usuario mencionado recibirá una notificación. Además, un usuario también puede responder *tweets* de otro y a esto se lo conoce como *reply*. Como cualquier usuario puede participar en la conversación, debido a que casi la totalidad de los perfiles de esta red social son públicos[1], se crean debates multitudinarios acerca de muchos temas y las conversaciones pueden bifurcarse creando una compleja red de mensajes.

Otro elemento de gran importancia en Twitter son los llamados *hashtags*. Son etiquetas que permiten identificar fácilmente *tweets* sobre un tema o noticia determinados. Los usuarios no tienen más que incluir en sus *tweets* el *hashtag* precedido del símbolo #. De esta forma se puede conocer en todo momento los temas que más están dando que hablar en las distintas zonas del mundo tan sólo analizando los *hashtags* más usados. Los *hashtags* también son una herramienta de propaganda comercial y política. Actualmente, en multitud de programas de la televisión animan a los telespectadores a comentar de forma paralela el programa vía Twitter y para ello suelen sugerir un *hashtag*. Los partidos políticos están intentando copiar esta idea con fines electorales, para difundir un *slogan* o atacar a los partidos rivales, ya que los *hashtags* tienen un enorme potencial de repercusión mediática.

2.2 Candidatos de los principales partidos políticos a las elecciones autonómicas del 24 de mayo de 2015

Se han considerado los candidatos de los partidos políticos más importantes que se presentaban a las elecciones autonómicas de Madrid: el Partido Popular (PP), Ciudadanos (C's), Izquierda Unida (IU), Unión Progreso y Democracia (UPyD), Podemos y el Partido Socialista Obrero Español (PSOE). Los datos se han extraído de las páginas web de los distintos candidatos o partidos en los casos en los que ha sido posible.

1. **Cristina Cifuentes (PP):** Madrid, 1964. Licenciada en Derecho por la Universidad Complutense de Madrid (UCM) y con un máster en Administración y Dirección de Empresas del Instituto de Investigación Universitario Ortega y Gasset. Accedió por oposición al Cuerpo de Técnicos Superiores de la UCM, puesto que compatibiliza con su actividad política.

   Comenzó su andadura política en Alianza Popular, que fue refundado y pasó a llamarse Partido Popular. Actualmente forma parte del Comité de Dirección,

[1]Twitter permite que la cuenta de sus usuarios pueda ser privada, en cuyo caso sólo podrán acceder a sus *tweets* quienes sean sus seguidores con la correspondiente confirmación del usuario. Esto es poco común ya que va en contra del espíritu de Twitter, que defiende que tus mensajes lleguen al mayor número de gente posible. Con una cuenta privada sólo se tendrá acceso a los seguidores, lo que supone una clara desventaja con fines electorales o comerciales.

del Comité Ejecutivo y de la Junta Directiva Regional. También es Presidenta del Comité de Derechos y Garantías.

Figura 2.1: Captura de la página de Twitter de Cristina Cifuentes

2. **Ignacio Aguado (Ciudadanos):** Madrid, 1983. Licenciado en Derecho y Administración y Dirección de Empresas por la Universidad Pontificia de Comillas. Licenciado en Ciencias Políticas y de la Información por la Universidad Autónoma de Madrid. Realizó un Máster en Comunicación Política e Institucional por la Fundación Ortega y Gasset y es titulado en *Lobby & Advocacy* por el Instituto de Empresa (IE). En 2008 comenzó a trabajar para Unión Gas Fenosa, donde ha permanecido hasta su designación como candidato por su partido, tras lo cual pidió una excedencia.

   En 2013 se afilió a Ciudadanos cuando aún contaba con muy pocos miembros en la Comunidad de Madrid. A principios de 2014 fue elegido portavoz de este partido en dicha localidad.

Figura 2.2: Captura de la página de Twitter de Ignacio Aguado

3. **Luis García Montero (IU):** Granada, 1958. Licenciado en Filosofía y Letras por la Universidad de Granada, donde trabajó más tarde como profesor. Actualmente ostenta el título de Catedrático de Literatura Española por dicha universidad. Además, es crítico literario, poeta y ensayista. Ha recibido varios premios por sus poemarios (Premio Adonáis, Premio Loewe, Premio Nacional de Literatura y Premio Nacional de la Crítica).

De joven militó en el Partido Comunista de España (PCE) y se pasó a Izquierda Unida tras su formación. En las Elecciones europeas de 2004 se presentó en las listas de IU y en 2012 pasó a ocupar un cargo importante dentro de Izquierda Abierta, un partido de nueva creación integrado en IU.

Figura 2.3: Captura de la página de Twitter de Luis García Montero

4. **Ramón Marcos Allo (UPyD):** Burgos, 1969. Licenciado en Derecho por la Universidad de Valladolid y con *Executive Master* en Gestión Pública por el Instituto de Empresa. Es Letrado de la Administración de la Seguridad Social desde 1997.

   Fue militante del Partido Socialista de Cataluña (PSC), que abandonó debido a su posterior oposición a ciertas ideas del partido. Ingresó durante un breve período de tiempo en Ciudadanos (C's) durante 2007. Fue uno de los fundadores de Unión Progreso y Democracia y en 2011 fue elegido diputado en la Asamblea de Madrid.

Figura 2.4: Captura de la página de Twitter de Ramón Marcos Allo

5. **José Manuel López (Podemos):** Madrid, 1966. Ingeniero Agrónomo por la Universidad Politécnica de Madrid y diplomado en Ordenación del Territorio, Desarrollo Rural y Medio Ambiente. Comenzó su carrera profesional trabajando en la consultoría privada. Ha sido director de Análisis Social y Desarrollo de Cáritas, miembro de la comisión ejecutiva de la Comisión Española de Ayuda

al Refugiado (CEAR), director de la Fundación pública Pluralismo y Convivencia y vocal del Observatorio del Pluralismo Religioso en España.

Ha participado en política desde la sociedad civil, involucrándose en asociaciones de su barrio, Hortaleza. Asimismo, es miembro del colectivo *Qué Hacemos* y del círculo 3E (economía, ecología y energía).

Figura 2.5: Captura de la página de Twitter de José Manuel López

6. **Ángel Gabilondo (PSOE):** San Sebastián, 1949. Licenciado en Filosofía y Letras por la Universidad Autónoma de Madrid (UAM). Posteriormente, obtuvo sobresaliente en su Tesis Doctoral y fue rector de la UAM, miembro del departamento de Filosofía de dicha facultad y presidente de la Conferencia de Rectores de Universidades Españolas (CRUE). Actualmente ostenta el título de Catedrático en Filosofía.

   Entre 2009 y 2011 fue ministro de Educación sustituyendo a Mercedes Cabrera, abandonando de esta manera su puesto en la UAM. No se encuentra afiliado al partido, pero es afín a las ideas del PSOE.

Figura 2.6: Captura de la página de Twitter del equipo de Ángel Gabilondo

2.3 Minería de textos

Los textos albergan una gran cantidad de información que los ordenadores no pueden analizar, ya que tradicionalmente han sido tratados como simples secuencias de caracteres. Es necesario aplicar métodos y algoritmos para procesar estos textos y extraer información de utilidad a partir de ellos. Este campo se encuentra en constante crecimiento debido a la proliferación de las redes sociales y a que se estima que alrededor del 80 % de la información relacionada con el mundo empresarial se encuentra almacenada en forma de texto [6].

La minería de textos es un área multidisciplinar que se encarga de la obtención automática de información a partir de textos, no teniendo así que depender de una persona para poder extraer información de ellos. Utiliza conocimientos procedentes de la Minería de Datos, del Aprendizaje Automático, de la Estadística y de la Lingüística Computacional. Debido a que la mayor parte de la información se encuentra almacenada en forma de texto en la red, resulta de gran utilidad automatizar la tarea de analizarla, ya que tiene un alto valor comercial. Gracias a este creciente interés de las empresas en el tema ha aumentado significativamente el número de estudios en esta materia durante los últimos años.

El término *Minería de Textos* fue mencionado por primera vez por Feldman y Dagan [7], quienes proponían una nueva forma de abordar el análisis de los textos basándose en tres componentes: el concepto de jerarquía, la categorización en conceptos y la búsqueda de patrones inusuales en las distribuciones de estos conceptos.

A partir de estos tres principios la minería de textos ha ido evolucionando hasta convertirse en un campo con gran peso dentro de la minería de datos. Se pueden distinguir tres definiciones alternativas de minería de textos en función del área en el que se esté aplicando [8]:

1. Entendida como extracción de información: Se asume que el objetivo es extraer hechos a partir de textos.

2. Entendida como minería de datos: Aplicación de algoritmos y métodos del campo del Aprendizaje Automático y de la Estadística a los textos para extraer patrones que sean de utilidad. Para ello es importante realizar un pre-proceso adecuado y, a veces, técnicas de análisis del Lenguaje Natural[2].

3. Entendida como un proceso de extracción de conocimiento (KDD[3]): Consiste en una serie de pasos orientados a extraer información subyacente en una colección grande de textos, pero que aún no ha sido descubierta. Para ello se apoya en las técnicas de las dos definiciones anteriores.

Las dos últimas aproximaciones son muy parecidas. Para entender la diferencia

[2]En inglés, *Natural Language Proccessing* (NLP).
[3]En inglés, *Knowledge Discovery in Databases* .

entre ambas debemos comprender en qué consiste un proceso de KDD.

Un proceso de extracción de conocimiento está conformado por varias subtareas que se pueden agrupar en cinco etapas:

- *Selección de datos:* Se escogen los datos a utilizar a partir de las fuentes, separando aquellos que son relevantes de los que no nos aportan información de utilidad.

- *Preprocesamiento:* Se preparan y se limpian los datos para poder manejarlos con facilidad, algo que es necesario en las fases posteriores.

- *Transformación:* Se transforman los datos existentes y se generan nuevas variables en caso de ser necesario. En este paso se realizan operaciones de agregación o normalización.

- *Minería de datos:*[4] Se obtienen los modelos tras la aplicación de algoritmos adecuados.

- *Evaluación:* Se analizan e interpretan los resultados utilizando ciertas medidas de error, para poder otorgar un grado de fiabilidad al modelo obtenido.

Además de estas etapas suele haber otras dos: una fase preliminar para establecer los objetivos del problema de minería de textos (ya que muchas veces el problema no está definido de forma clara) y otra final donde los resultados se integran con un modelo de negocio para su explotación posterior.

La aproximación a la minería de textos basada en la minería de datos es una pequeña parte de un proceso mucho más amplio. La tercera definición es la que se ha usado a la hora de realizar este trabajo, al considerarse la más completa.

Al planificar un proyecto de extracción de conocimiento es útil aplicar una metodología bien definida que acote las distintas fases del problema correctamente. El estándar más comúnmente utilizado es CRISP-DM[5] y divide cada una de las etapas explicadas anteriormente en tareas, lo que permite abordar más fácilmente el problema [9] (véase la Figura 2.7). Sin embargo, este estándar sólo sirve como pauta general, pues las tareas pueden variar mucho en función del problema.

A la hora de realizar el presente trabajo las directrices de CRISP-DM se han usado de forma orientativa. Las fases que se han seguido no coinciden en su totalidad con este estándar, ya que la minería de textos presenta ciertas particularidades que CRISP-DM no contempla al estar orientado a procesos generales de KDD. Tampoco ha sido contemplado el enfoque empresarial que presenta CRISP-DM por no ser

[4]En muchos casos se utiliza el término de Minería de Textos (*data mining* en inglés) para referirse al proceso de KDD. Esto está comúnmente extendido, por lo que KDD y minería de datos se suelen usar indistintamente, pero en principio la minería de textos es un paso dentro del proceso de KDD.

[5]En inglés, *Cross Industry Standard Process for Data Mining.*

pertinente.

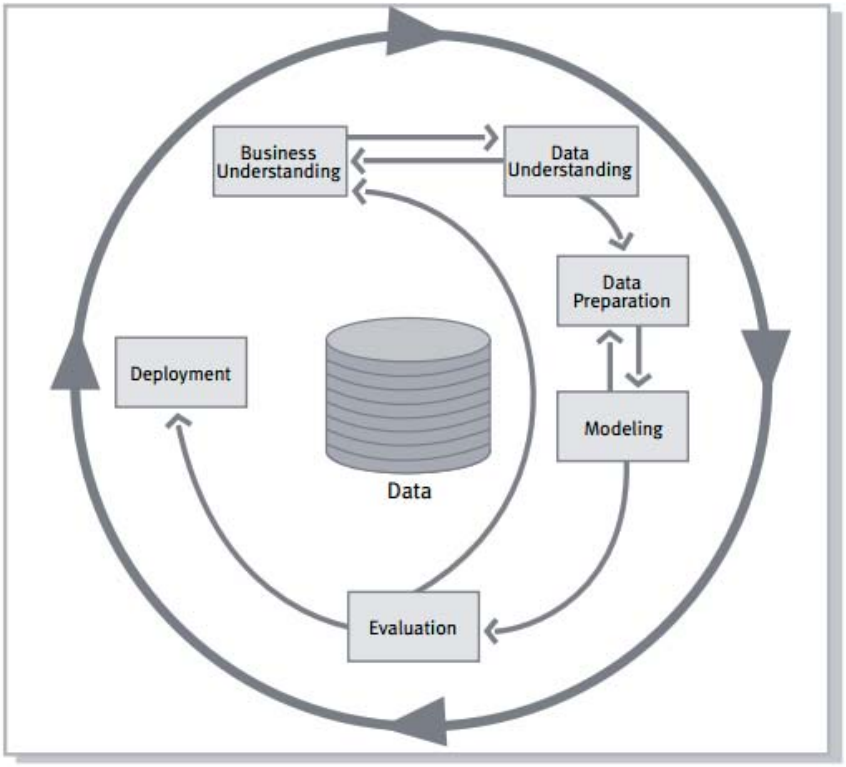


Figura 2.7: Fases de CRISP-DM
(imagen extraída de [9])

El proceso mostrado en la figura 2.7 no es lineal, ya que el orden de ejecución de las fases puede variar en función de los resultados. Esto añade bastante complejidad al proceso, ya que en la mayoría de los casos hay que avanzar y retroceder entre las diferentes fases. Los resultados que se obtienen en cada fase son determinantes para decidir cuál será la siguiente. El círculo externo representa la naturaleza cíclica del proceso de KDD, ya que una vez obtenido un modelo, los procesos posteriores se beneficiarán de los resultados. Cada vez se logrará un modelo más refinado, pues se irá contando con más información.

Para poder trabajar en el campo de la minería de textos resulta bastante útil contar con herramientas ya desarrolladas que implementen muchas de las tareas frecuentes a la hora de analizar y extraer información de los textos. En este estudio se ha usado la biblioteca nltk[6] de Python[7], que ha resultado ser muy eficaz a la hora de trabajar con el corpus que contenía los textos para analizar.

[6] *Natural Language Toolkit* (versión 3.0): `http://www.nltk.org/`

[7] Lenguaje escogido en este trabajo debido a su gran versatilidad y sencillez. Se ha utilizado la versión 3.4.

2.4 Aprendizaje automático

El Aprendizaje Automático, también llamado Aprendizaje de Máquinas, es una rama de la Inteligencia Artificial cuyo objetivo consiste en desarrollar modelos que permitan a las máquinas generalizar comportamientos partiendo de cierta información suministrada a modo de ejemplos. En algunas ocasiones se solapa con el campo de la Estadística, pero el Aprendizaje Automático tiene un enfoque mucho más orientado a la computación y resolución de problemas.

Si se relaciona este campo con los procesos de KDD tratados en el apartado anterior, se puede situar el Aprendizaje Automático dentro de la etapa de Minería de Datos, ya que sus algoritmos hacen posible obtener los modelos buscados para extraer conocimiento de los datos.

Se distinguen dos tipos de aprendizaje automático:

- *Aprendizaje supervisado o clasificación:* Usa datos con etiquetas de clase. Tiene como objetivo aprender a partir de un conjunto de entrenamiento y poder así obtener una función o modelo que sea capaz de etiquetar correctamente otros datos de entrada.

- *Aprendizaje no supervisado o clustering:* Engloba aquellas técnicas que no utilizan etiquetas de clase en el conjunto de entrenamiento (es decir, no existe conocimiento *a priori*). Estas técnicas tratan de detectar en el conjunto de datos diferentes grupos o *clusters* que comparten ciertas características en común.

En el caso de los procesos de clasificación los datos etiquetados suelen dividirse en dos subgrupos:

- *Conjunto de datos de entrenamiento*[8]*:* Datos que utiliza el algoritmo para aprender y obtener así un clasificador.

- *Conjunto de datos de prueba*[9]*:* Datos que se utilizan para estudiar la confianza del clasificador obtenido.

[8] En inglés, *training set*.
[9] En inglés, *testing set*.

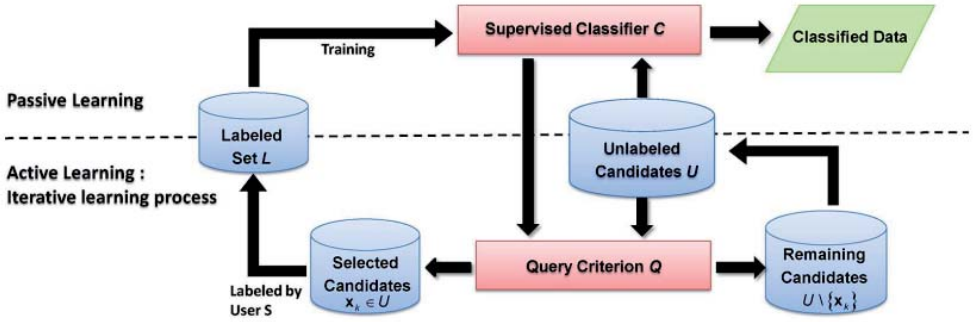


Figura 2.8: Flujo en un proceso de aprendizaje automático
(Imagen extraída de [10])

En la Figura 2.8 se utiliza un subconjunto de los datos de entrada para entrenar al algoritmo, mientras que el resto se reservan para evaluar posteriormente el clasificador obtenido. Para ello se usan valores estadísticos y otras medidas que permiten estimar la exactitud del modelo.

Existen numerosos algoritmos que se pueden emplear a la hora de realizar aprendizaje supervisado y no supervisado. Se ha decidido utilizar el método de factorización de matrices no negativas para realizar *clustering* y el método de Naïve-Bayes para realizar clasificaciones.

2.5 *Clustering*

Las técnicas de *clustering* tienen como objetivo agrupar objetos en grupos, de forma que los elementos de un grupo (también denominado *cluster*) guarden más similitudes entre ellos que con los elementos de otros grupos.

A estas técnicas también se las denomina aprendizaje automático no supervisado, ya que se parte de un conjunto de datos sin etiquetar de los que no se tiene conocimiento *a-priori* acerca de las categorías buscadas.

Para llevar a cabo un proceso de *clustering* en minería de textos es necesario estudiar los distintos algoritmos que se encuentran a nuestra disposición para elegir aquellos métodos que se adapten mejor a nuestro problema.

Muchos de estos métodos se basan en el concepto de distancia, pero debido a la dificultad de definir distancias entre los diferentes textos, se han descartado. Se ha optado por utilizar técnicas de extracción y selección de atributos que se basan en la factorización matricial. Se han tomado como atributos las palabras, aplicando al problema el modelo denominado *bag of words*, donde un texto se identifica por sus palabras más significativas, descartando la estructura gramatical y el orden de

las palabras dentro del texto, pero teniendo en cuenta el número de repeticiones de cada una de ellas.

Los métodos SVD[10] (*Descomposición en Valores Singulares*) y PCA[11] (*Análisis de Componentes Principales*) se descartaron porque no modelizaban bien el problema. En ellos están contempladas las matrices negativas, pero en el modelo *bag of words* sólo se emplean matrices no negativas (todos sus elementos mayores o iguales que cero), ya que se construyen a partir de las frecuencias de aparición de las palabras en los distintos textos de nuestro corpus.

Se ha escogido el método NMF[12](*Factorización de Matrices No Negativas*) para encontrar categorías, ya que trabaja con matrices no negativas.

El método NMF se basa en la factorización matricial, de forma que la matriz inicial se divide en dos matrices más sencillas.

Las frecuencias de aparición de las palabras en los textos nos permiten construir una matriz de frecuencias de dimensión $n \times k$, donde n es el número de textos y k el número de palabras. Cada elemento i, j de esta matriz M indica la frecuencia de aparición de la palabra j en el texto i.

$$\begin{array}{c} \\ \text{texto1} \\ \cdots \\ \text{texto n} \end{array} \begin{array}{c} \begin{array}{ccccc} \text{palabra 1} & \cdots & & \cdots & \text{palabra k} \end{array} \\ \begin{pmatrix} a_{11} & \cdots & \cdots & \cdots & a_{1k} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ a_{n1} & \cdots & \cdots & \cdots & a_{nk} \end{pmatrix} \end{array}$$

Esta matriz de frecuencias se descompone en dos matrices menores: la matriz de pesos W y la matriz de características F. El producto de estas dos matrices es aproximadamente M:

$$M \approx W \cdot F$$

La matriz de pesos W es una matriz $n \times m$ que contiene los pesos de cada una de las m características en los diferentes textos. El valor de m indica el número de categorías buscadas y debe definirse previamente. Debe escogerse con cuidado, ya que un número demasiado alto conlleva un etiquetado demasiado específico de los textos, mientras que uno muy pequeño da lugar a categorías demasiado generales que no proporcionan información de utilidad.

[10]En inglés, *Singular Value Descomposition.*
[11]En inglés, *Principal Component Analysis.*
[12]En inglés, *Non-negative Matrix Factorization*

$$\begin{array}{c} \\ \text{texto 1} \\ \cdots \\ \text{texto n} \end{array} \begin{array}{c} \begin{array}{cccccc} & \text{característica 1} & \cdots & & \text{característica m} \end{array} \\ \begin{pmatrix} a_{11} & \cdots & \cdots & \cdots & a_{1m} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ a_{n1} & \cdots & \cdots & \cdots & a_{nm} \end{pmatrix} \end{array}$$

La matriz F es una matriz $m \times k$ en donde a cada una de las características se le asignan pesos para cada una de las palabras en función de su importancia.

$$\begin{array}{c} \\ \text{característica 1} \\ \cdots \\ \text{característica m} \end{array} \begin{array}{c} \begin{array}{cccccc} & \text{palabra 1} & \cdots & & \text{palabra k} \end{array} \\ \begin{pmatrix} a_{11} & \cdots & \cdots & \cdots & a_{1k} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ a_{m1} & \cdots & \cdots & \cdots & a_{mk} \end{pmatrix} \end{array}$$

El algoritmo usado para la descomposición matricial no es trivial y utiliza técnicas de optimización multidimensional. Existen distintas técnicas para realizar la factorización. Se han usado implementaciones de la librería *scikit-learn*. El algoritmo de factorización NMF que se emplea en esta librería se basa en la proyección de gradientes [11]. También se usa un método de inicialización conocido como NNDSVD[13] [12]. Gracias a esta técnica mejora el tiempo de la factorización y se minimiza mejor el error cometido. Esto tiene una gran importancia, pues en la actualidad no existen algoritmos que nos proporcionen resultados óptimos (aquellos que garantizan un mínimo global del error) para este problema. Todos los métodos existentes proporcionan un mínimo local que, pese a ser mejorable, es útil en la mayoría de los casos.

2.6 Clasificación

Las técnicas de clasificación buscan la obtención de modelos que permitan asignar una etiqueta a una entidad a partir de ciertas propiedades que la definen. Los procesos de clasificación también son denominados aprendizaje automático supervisado, ya que es necesario contar con datos previamente etiquetados para que el algoritmo pueda aprender de ellos.

Se distinguen tres fases a la hora de realizar un proceso de clasificación: la definición de las clases, la representación de la información mediante atributos y el

[13]Nonnegative Double Singular Value Decomposition

aprendizaje mediante algoritmos [13]. En el caso concreto de los textos, se usa junto con técnicas de procesamiento del lenguaje natural. Dado un conjunto de textos etiquetados por tema o según su polaridad, se pretende construir un modelo que asigne una etiqueta otro texto nuevo basándose en lo aprendido a partir del corpus etiquetado proporcionado.

Formalmente, el objetivo de la clasificación (o aprendizaje automático supervisado) es obtener un modelo h de una función desconocida f que toma elementos de un espacio de entrada X y los transforma en elementos de un espacio discreto sin orden Y, todo ello usando un conjunto de entrenamiento S.

El conjunto $S = \{(x_1, y_1), \cdots, (x_n, y_n)\}$ contiene n muestras de entrenamiento (x, y) donde $x \in X$ e $y = f(x) \in Y$. Los valores de X pueden ser continuos o discretos y describen los atributos de la entidad que se va a clasificar. Los valores de Y son las clases en las que queremos clasificar dichas entidades.

A partir del conjunto de entrenamiento S, un algoritmo de entrenamiento calcula un clasificador h que pertenece a un conjunto de posibles funciones H, denominado *espacio de hipótesis*. En función del algoritmo empleado se considerarán diferentes conjuntos H y también variará la técnica empleada para escoger el clasificador $h \in H$ que mejor se adapte al conjunto de entrenamiento.

El clasificador obtenido nos permite asignar una clase a cada elemento nuevo de entrada. El error cometido en esta clasificación se denomina error de generalización[14]. Este error no puede minimizarse sin conocer la función f o la distribución $P(X, Y)$, por lo que el error que se va a minimizar se realiza sobre un conjunto etiquetado de entrenamiento. Cuando el conjunto de entrenamiento es lo suficientemente grande, este error supone una buena estimación. Sin embargo, en el caso de un conjunto de entrenamiento insuficiente se corre el riesgo de sobreajuste[15], que supone una generalización errónea del conjunto de entrenamiento. La presencia de ejemplos atípicos[16], inconsistentes o mal etiquetados también puede conllevar un sobreajuste.

Se distinguen cuatro tipos principales de algoritmos de clasificación según estén basados en modelos probabilísticos, en distancias, en reglas o en *kernels*. Se contemplaron dos algoritmos a la hora de realizar este trabajo: Naïve Bayes (basado en el teorema de Bayes) y las Máquinas de Vectores Soporte(SVM)[17]. Aplicando ambos métodos al análisis de textos se obtienen valores de error muy similares [14]. El corpus y las técnicas de preproceso empleados suelen ser más determinantes a la hora de inclinar la balanza a favor o en contra de un determinado algoritmo. En algunos estudios se recomienda el uso de técnicas SVM[15], mientras que en otros Naïve Bayes arroja mejores resultados [16][17]. En los casos en los que se cuenta con un número reducido de características la técnica SVM se comporta mejor[18], pero

[14] En inglés, *generalisation* o *true error*.
[15] En inglés, *overfitting*.
[16] En inglés *outliers*.
[17] En inglés, *Support Vector Machines*.

esto no ocurre en minería de textos, donde el número de características suele ser muy elevado.

A continuación se va a profundizar en la clasificación usando Naïve Bayes, que ha sido la técnica finalmente escogida. La mayor sencillez del modelo obtenido usando Bayes ha influido bastante en esta decisión.[19]

El algoritmo de Naïve Bayes clasifica elementos $x = (x_1, \cdots, x_m)$ asignándoles la clase k que maximiza la probabilidad condicional de esa clase a partir los atributos de x.

$$\max_{\text{k}} [P(K|x_1, \cdots, x_m)] = \max_{\text{k}} \left[\frac{P(x_1, \cdots, x_m)|k)P(k)}{P(x_1, \cdots, x_m)} \right] \approx \max_{\text{k}} \left[P(k) \prod_{i=1}^{m} P(x_i|k) \right]$$

donde $P(k)$ y $P(x_i|k)$ se estiman a partir del conjunto de entrenamiento utilizando las frecuencias relativas. A esta estimación se la denomina de máxima verosimilitud[18].

Uno de los mayores problemas de este método es la asunción de independencia entre los distintos atributos. También es importante que el conjunto de entrenamiento esté balanceado (mismo número de elementos de cada una de las clases que contiene), ya que en caso contrario el clasificador tenderá a clasificar más ejemplos hacia la clase predominante del conjunto de entrenamiento. Sus mayores ventajas son su simplicidad y eficiencia computacional.

A pesar de sus limitaciones, este método ha sido uno de los más usados en clasificación, ya que cuando el conjunto de entrenamiento representa muy bien las distribuciones de probabilidad del problema se obtienen muy buenos resultados.

2.7 Análisis de sentimiento

Debido al aumento de las redes sociales, el contenido generado por los usuarios en la red se ha disparado. Conseguir identificar la polaridad de las emociones en estos textos pequeños e informales es una tarea importante que presenta un gran número de dificultades. Las incorrecciones ortográficas son uno de estos problemas, pero hay muchos otros como la falta de contexto (debido a la brevedad de los textos) o la ironía.

Según Pang y Lee [20], este campo comenzó a tener interés para la empresa en el año 2001 y, en los últimos años, ha aumentado significativamente el número de estudios realizados en este campo debido al enorme potencial que presenta. Uno de los principales motivos de este aumento ha sido el auge de Twitter, que posibilita aplicaciones en sistemas de recomendación, asistencia virtual o estudios de inteligencia comercial o política [21].

[18] En inglés, *maximum likelihood estimation*

El análisis de sentimiento surgió a raíz de la necesidad de tener que analizar un gran número de textos con opiniones que, a menudo, no se encuentran de forma explícita. Se encarga de la búsqueda, la recolección, el análisis y la visualización de puntos de vista. Se denominó minería de opiniones (OP[19]) o análisis de sentimientos(SA[20]). Este área presenta muchas similitudes con la minería de textos, pero también la dificultad añadida de tener que tratar con la subjetividad de las opiniones y los sentimientos.

Algunas de las características de Twitter hacen que este tipo de análisis sea mucho más complicado. Las más destacadas son las siguientes:

- El estilo informal de los *tweets* conlleva que las abreviaciones, las expresiones informales y el argot sean frecuentes.
- La gramática incorrecta de muchos mensajes dificulta llevar a cabo análisis lingüísticos.
- Debido al límite de 140 caracteres por *tweet* aparece un gran número de palabras acortadas y expresiones extrañas. Este problema se conoce como *sparsity*.
- La falta de contexto nos impide situar los mensajes dentro de un marco determinado, que a veces es fundamental para su comprensión. Éste es uno de los problemas más difíciles de resolver.
- El extendido uso de la ironía en los mensajes provoca que muchas veces se identifiquen *tweets* como positivos o negativos de forma incorrecta.

En la actualidad se llevan a cabo anualmente competiciones destinadas a mejorar las técnicas de análisis de sentimiento. Desde el año 2012 se ha estado realizando el taller TASS para el análisis de sentimientos en castellano, donde se proponen distintos retos relacionados con el tema.[21]

[19]Siglas de *Opinion Mining*

[20]Siglas de *Sentiment Analysis*

[21]Organizado por Daedalus, la Universidad Politécnica de Madrid y la Universidad de Jaén como un evento satélite para la conferencia anual del SEPLN (Spanish Society for Natural Language Processing).

Capítulo 3

Metodología

En esta sección se muestran las distintas técnicas analizadas y las herramientas que se han desarrollado para la consecución de los objetivos marcados.

3.1 Técnicas utilizadas

3.1.1 Obtención del corpus

La obtención de los *tweets* que conforman el corpus que se ha utilizado en este trabajo se ha realizado a través de la API[1] proporcionada por Twitter.[2] Esta librería nos proporciona las funcionalidades necesarias para filtrar *tweets* atendiendo a distintas características y descargarlos.

Para poder usar la API es necesario registrar la aplicación en Twitter. Tras realizar este registro se obtienen las siguientes claves:

- *Consumer key* y *consumer secret:* Permiten la autentificación[3] de la aplicación.
- *Access token* y *access token secret:* Permiten la autentificación[3] del desarrollador de la aplicación.

Es necesario usar ambos conjuntos de claves para registrarse, tanto a nivel de aplicación como de usuario. Esta autentificación se realiza mediante el protocolo OAuth.[4] OAuth es un protocolo de autorización[5] que permite a un usuario acceder

[1] *Application Programming Interface*. Librería con funciones desarrolladas para ser utilizadas por un programa informático con un fin específico.

[2] `https://dev.twitter.com/`

[3] Comprobación de los credenciales al realizar un intento de conexión. En este proceso un cliente envía a un servidor sus credenciales, que suelen ir cifrados.

[4] *Open Authorization*. Más información en `http://oauth.net/`

[5] Comprobación de que se da como válido un intento de conexión. Tiene lugar tras una auten-

a los datos de un proveedor sin que éste tenga que compartir toda su identidad.

Se ha realizado en Python (con la colaboración de Steven Ortiz, alumno de la UPM) un programa que extrae los *tweets* utilizando los métodos que nos proporciona la API de Twitter.

El programa extractor cuenta con dos módulos. El primero contiene los métodos encargados de realizar la carga de los credenciales (las cuatro claves previamente explicadas) y las operaciones necesarias para almacenar los *tweets* en formato JSON[6]. El segundo es el programa ejecutable y recibe dos ficheros de texto como entrada. Uno contiene las consultas que filtran los *tweets* para descargar y el otro los credenciales. Las consultas (denominadas *queries* en inglés) se realizan utilizando una sintaxis específica desarrollada para ello, proporcionada por la Search API de Twitter[7].

Uno de los problemas de utilizar la Search API de Twitter es que no proporciona un resultado exhaustivo de los *tweets* de las consultas realizadas. Por tanto, no se puede garantizar que se estén obteniendo todos los *tweets* de un determinado usuario. Además, los *tweets* que se obtienen tienen como máximo una antigüedad de una semana[7], por lo que es necesario definir una política de extracción.

Los campos que se han extraído de los *tweets* se eligieron tomando un subconjunto de todos los que devolvía la API de Twitter. En el corpus los *tweets* tienen un número identificador, que es único. Por tanto, no hay dos *tweets* iguales repetidos.

La información de cada uno de los *tweets* se ha almacenado utilizando los siguientes campos:

- *Autor:* Se almacena su número identificador y su nombre.
- *Hashtags:* Lista de todos los *hashtags* usados en el *tweet*.
- *Número de retweets.*
- *Número de favoritos.*
- *Texto en formato UTF-8.*[8]
- *Booleano que indica si es un retweet.*
- *Booleano que indica si contiene un enlace.*

tificación correcta.

[6]Acrónimo de JavaScript Object Notation. Es un formato estándar para el intercambio de datos. Tiene la ventaja de ser muy fácil de usar. En Python, utilizando la librería JSON, se pueden convertir directamente a diccionarios con los que es muy fácil trabajar.

[7]`https://dev.twitter.com/rest/public/search`

[8]Acrónimo de 8-bit Unicode Transformation Format. Es un formato de codificación que utiliza símbolos de longitud variable. Es el más usado en la actualidad.

- *Booleano que indica si contiene un vídeo o una imagen.*

- *Usuario al que se contesta:* Se almacena su número identificador, su nombre y el identificador del *tweet* contestado (en caso de no contestarse a nadie contiene valores nulos).

- *Momento de la publicación:* Contiene la fecha y la hora.

El texto y los *hashtags* se encontraban en formato UTF-8, pero algunos contenían símbolos y acentos que se querían eliminar. Por tanto, se decidió utilizar la librería Unidecode[9] de Python, que eliminaba todos los caracteres extraños, sustituía todos los caracteres acentuados por su versión sin acento y la letra ñ por la letra n (la librería realizaba también otras conversiones, pero no son relevantes en nuestro caso).

En resumen, se ha desarrollado una herramienta de extracción de *tweets* para su estudio posterior. A partir de los datos obtenidos se han realizado métodos en Python para conformar un corpus, a partir del cual se pueden extraer datos estadísticos y donde se pueden aplicar distintas técnicas de minería de textos.

3.1.2 Detección de temas

Se ha desarrollado una herramienta que permite extraer un número determinado de temas a partir de un corpus dado. Se han tenido en cuenta varios algoritmos de clasificación no supervisada.

El método NMF (introducido en la sección 2.5) permite reducir la dimensión del problema al factorizar la matriz de las frecuencias de las palabras en otras dos: la matriz de características F y la de pesos W (se cumple que $M \approx W \cdot F$ siendo M la matriz que contiene las frecuencias). La matriz de características contiene la relación entre los temas (también denominados *clusters* o categorías) y las palabras. De esta forma, se puede saber qué palabras son las que tienen más peso entre los temas en los que se ha divido el corpus. La matriz de pesos informa acerca del grado de pertenencia de los distintos textos del corpus a las categorías encontradas.

El método NMF se ha aplicado usando la librería *scikit-learn* de Python, que trae implementados de forma eficiente una gran cantidad de algoritmos de minería de datos. Sin embargo, antes de aplicar el algoritmo ha sido necesario aplicar técnicas de preprocesado a los datos para poder obtener buenos resultados. Para realizar este preproceso se contó con la ayuda de la librería *nltk* de Python, que facilita bastante el trabajo con corpus de textos.

[9] `https://pypi.python.org/pypi/Unidecode`. Se ha utilizado la versión 0.04.17.

3.1.2.1 Técnicas de preprocesado

Uno de los mayores problemas a la hora de aplicar el método NMF consiste en que las matrices que se construían a partir de las frecuencias de las palabras en los textos eran demasiado grandes. Primero se decidió prescindir de las palabras con longitud menor que 4, ya que casi nunca proporcionaban información valiosa. Además no tenían en cuenta ni números ni signos de puntuación y todas las letras se convertían a minúsculas. También se eliminaron las denominadas *stopwords*, que son palabras que no aportan ningún significado semántico al texto y aparecen a menudo (las preposiciones o todas las conjugaciones del verbo haber, por ejemplo).

Sin embargo, este preproceso seguía sin ser suficiente, por lo que también se decidió eliminar las palabras que estuviesen en más de un determinado porcentaje de textos (un valor alrededor del 80 % parece ser el óptimo) y en un mínimo. De esta forma, se elimina información poco significativa y carente de utilidad. Para calcular el mínimo de textos en los que debía aparecer una palabra para ser tenida en cuenta no se utilizan porcentajes, ya que en el caso de procesar un número elevado de textos podría ser excesivamente alto y eliminar demasiadas palabras. Este valor varía en función del número de textos del corpus y sus características, siendo necesario hacer varias pruebas con el corpus usado para determinar el valor óptimo.

También se ha empleado una técnica denominada *stemming*, que reduce una palabra a su raíz[10] (en inglés, *stem*). Esto permite aunar palabras relacionadas en un mismo *stem*. Por ejemplo, las palabras gato, gatos y gatitos comparten el *stem gat*. Se ha usado el algoritmo de Porter para realizar el *stemming*, que en su adaptación al castellano usa una estructura conocida como *Snowball*, que se basa en la sufijación [22]. Este *stemmer* tiene poca agresividad, ya que se intenta evitar juntar en el mismo *stem* términos que no tengan nada que ver (*overstemming*).

Todas estas técnicas de preprocesado no sólo permiten reducir la dimensión de la matriz que usaremos en la factorización NMF, sino que también mejoran los resultados obtenidos, pues se pasa a trabajar con un subconjunto formado por las palabras más significativas.

3.1.2.2 Construcción de la matriz de frecuencias

Inicialmente la matriz de frecuencias iba a estar constituida por las frecuencias de aparición de las palabras en los distintos textos (siendo las filas los textos y las columnas las palabras tras el preproceso). Sin embargo, la frecuencia no constituye el mejor indicador para representar la importancia de las palabras en los documentos. Por ejemplo, si una palabra A aparece una vez en un documento y otra palabra B aparece tres veces en otro, no es razonable decir que la palabra B es el triple de importante que la palabra A. Para solucionar esto se realizó una transformación de los valores de las frecuencias aplicando logaritmos. De este modo, si una palabra

[10] No suele coincidir con el lexema.

aparece n veces más que otra su importancia no aumentará de forma lineal, sino que será $log(n)$.

También se decidió incluir una heurística denominada IDF (*Inverse Document Frequency*) que le asigna un valor menor a las palabras que aparecen en una gran cantidad de textos, pues son consideradas menos específicas y, por tanto, menos significativas. Dado un término t en un texto, su IDF se calcula aplicando la siguiente fórmula: $log(N/df_t)$. En esta expresión N es el número de documentos en el corpus y df_t es el número de documentos que contienen al término t (*document frequency*). Se observa que una palabra muy rara tendrá un IDF muy alto, mientras que si el término es muy frecuente su IDF será bajo.

Como las palabras que aparecen demasiado poco tampoco son muy relevantes, esta heurística no se adecuaba a nuestro problema, por lo que se decidió usar la técnica conocida como TF*IDF, que es una mejora de la anterior. En este caso, para calcular el peso de cada palabra en el documento se multiplica su frecuencia en el texto por su valor de IDF. En nuestro caso, como hemos mencionado previamente, se tomó el logaritmo de la frecuencia, ya que de otro modo, cuando una palabra era muy frecuente se anulaba el efecto que tenía la IDF. Podemos observar que siguiendo esta técnica el peso de una palabra será elevado cuando aparezca muchas veces en un número reducido de documentos y será menor cuando aparezca en un gran número de documentos. El valor que se obtiene para los casos marginales en los que una palabra aparece sólo en un documento siguen siendo altos, pero al estar presente sólo una vez, acaba siendo poco significativo en el *cluster* final.

3.1.2.3 Asignación de categorías a los textos

A partir de la matriz de características se pueden determinar las palabras que describen mejor cada uno de los temas. Las filas de esta matriz representan las distintas categorías encontradas y las columnas indican el peso de cada palabra en esa categoría (tomando las de mayor peso se tienen las más influyentes).

Analizando la matriz de pesos, los textos poseen un valor para cada categoría que indica el grado de pertenencia del texto dentro de ella. Otro método que también se empleó para determinar la pertenencia de un texto a una categoría consistió en usar la medida de similitud coseno entre las filas de la matriz de características y las de la matriz inicial. Estas dos formas de determinar las categorías de los textos arrojaron resultados similares (del orden del 95 % de coincidencia en las pruebas realizadas). Se optó por usar la matriz de pesos, ya que de ese modo los valores elevados se mantenían, mientras que realizando la similitud coseno todos los valores obtenidos eran menores que uno. Para determinar las relaciones entre categorías y textos se han realizado dos aproximaciones distintas en función de la definición de pertenencia considerada. Ambas aproximaciones se especifican a continuación:

- **Primera aproximación:** Se ha considerado que un texto pertenece sólo a una categoría, de forma cada texto pertenece sólo a la categoría en la que

tiene un valor mayor. En los casos en los que todos los valores son cero, se indica que el texto es no relevante. Esto se da cuando ninguna de las palabras del texto se encuentra entre las palabras que se obtuvieron tras el preproceso. Para diferenciar la importancia de cada texto en las distintas categorías se han discretizado estos valores atendiendo a la siguiente escala en función del valor p del peso de cada texto:

  - No relevante: $p = 0$
  - Muy poco relevante: $0 < p < 0{,}1$
  - Poco relevante: $0{,}1 \leq p < 0{,}3$
  - Relevante: $0{,}3 \leq p < 0{,}6$
  - Muy relevante: $p > 0{,}8$

  Esta escala se ha determinado de forma empírica analizando la distribución de los elementos de la matriz de pesos y realizando pruebas en distintos corpus.

- **Segunda aproximación:** Se ha considerado que un texto puede pertenecer a diferentes categorías. Para determinar los textos incluidos en cada categoría se tiene en cuenta la discretización anterior. Todos los textos no relevantes o muy poco relevantes no se incluyen en ninguna categoría. El número de textos relevantes en cada categoría aumenta, debido a la inclusión de un mismo texto en varias categorías. Así se modeliza mejor la pertenencia de un texto a una categoría, ya que se recoge información que antes se estaba perdiendo.

3.1.3 Análisis de sentimiento

El principal problema al realizar una herramienta que detectase automáticamente el sentimiento en los *tweets* ha sido la gran escasez de recursos en español. Según la literatura, la técnica que ofrece mejores resultados es el aprendizaje automático, pero para llevarlo a cabo se necesita un corpus etiquetado. Como no se ha encontrado ningún corpus etiquetado en función de su polaridad de características similares al de este estudio, se ha tenido que abordar el problema usando un diccionario con el valor afectivo de ciertas palabras. Se ha escogido una ampliación de ANEW[11], un referente en este tipo de diccionarios. Fue elaborado por Bradley y Lang [23] en 1999 y consta de 1034 palabras. La ampliación usada ha sido realizada por Warriner et al. [24], donde se han recogido cerca de 14000 palabras. Aunque este corpus estaba también en inglés, se ha usado el trabajo del profesor Daniel Gayo-Avello de la Universidad de Oviedo, que proporciona en su blog [25] una conversión semi-automática de dicho corpus al castellano (realiza una traducción al castellano y una segunda traducción al inglés como validación, además de ciertas comprobaciones manuales). Tras esta

[11] Siglas de Affective Norms for English Words.

traducción se cuenta con más de 9181 palabras (que quedan reducidas a 7901 tras la fase de *stemming*).

Se han utilizado los tres componentes que se emplean normalmente a la hora de distinguir emociones:

- Valencia[12]: Indica el agrado ante un estímulo. Permite determinar si un sujeto se siente contento o triste al leer una determinada palabra. En un extremo de esta escala se encuentran las emociones de felicidad, satisfacción o deseo. En el otro, emociones como infelicidad, molestia, melancolía, desesperación o aburrimiento.

- Excitación[13]: Indica la intensidad de la emoción que provoca el estímulo. Permite identificar si un sujeto se muestra calmado o excitado al leer una palabra. En un extremo, el lector se encuentra estimulado, excitado, frenético, nervioso, despierto o atento. En el otro, el sujeto manifiesta relajación, calma, somnolencia, aburrimiento o lentitud.

- Dominación[14]: Indica el grado de control que ejerce el estímulo sobre el individuo. En un extremo, el lector se encuentra controlado, influenciado, sumiso o sometido. En el otro extremo, las emociones implicadas son control, autonomía o autoridad.

En la Figura 3.1 se muestra el formulario[15] empleado a la hora de asignar un valor a cada uno de los indicadores anteriores. Fue utilizado en la confección del corpus ANEW por Bardley y Lang y fue rellenado por cada uno de los participantes.

[12]En inglés, *valence*
[13]En inglés, *arousal*
[14]En inglés, *dominance*
[15]Hoja de puntuaciones denominada *Self-Assesment Manikin*(SAM).

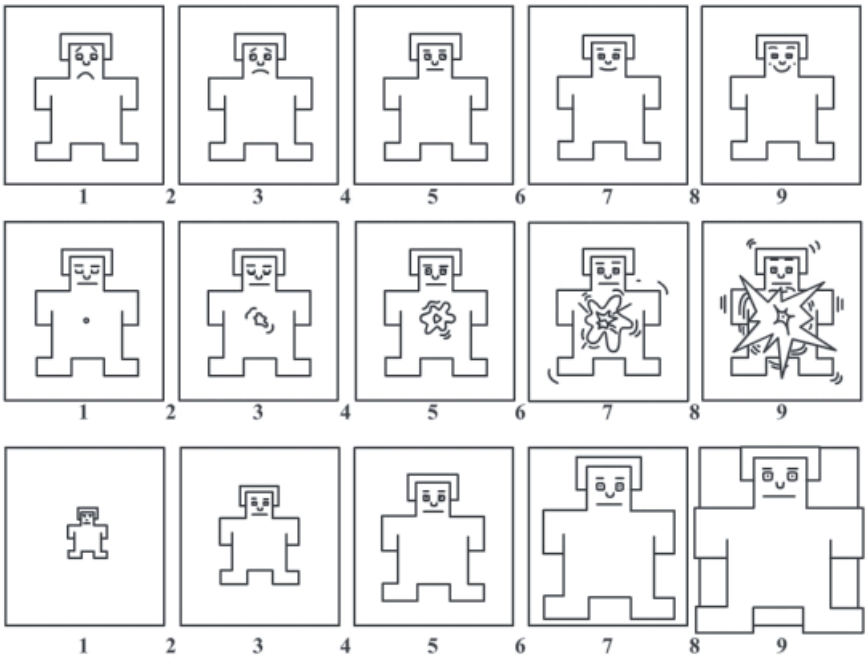


Figura 3.1: Escalas para asignar los valores de valencia, excitación y dominación (SAM)[26].

Un gran número de investigadores, entre los cuales se encuentran Mehrabian y Russell [27], Wundt [28] o Tellegen [29], han defendido una visión dimensional de las emociones. El modelo propuesto por Russell [30] afirma que todos los estados afectivos del individuo surgen a partir de dos sistemas neuropsicológicos denominados valencia y excitación.

Debido al carácter bidimensional de esta aproximación se pueden representar las distintas emociones con la ayuda de una circunferencia. A esta circunferencia se la conoce como modelo circumplejo de Russell. En la Figura 3.2 aparece representado este modelo junto con otros tres posteriores basados en él.

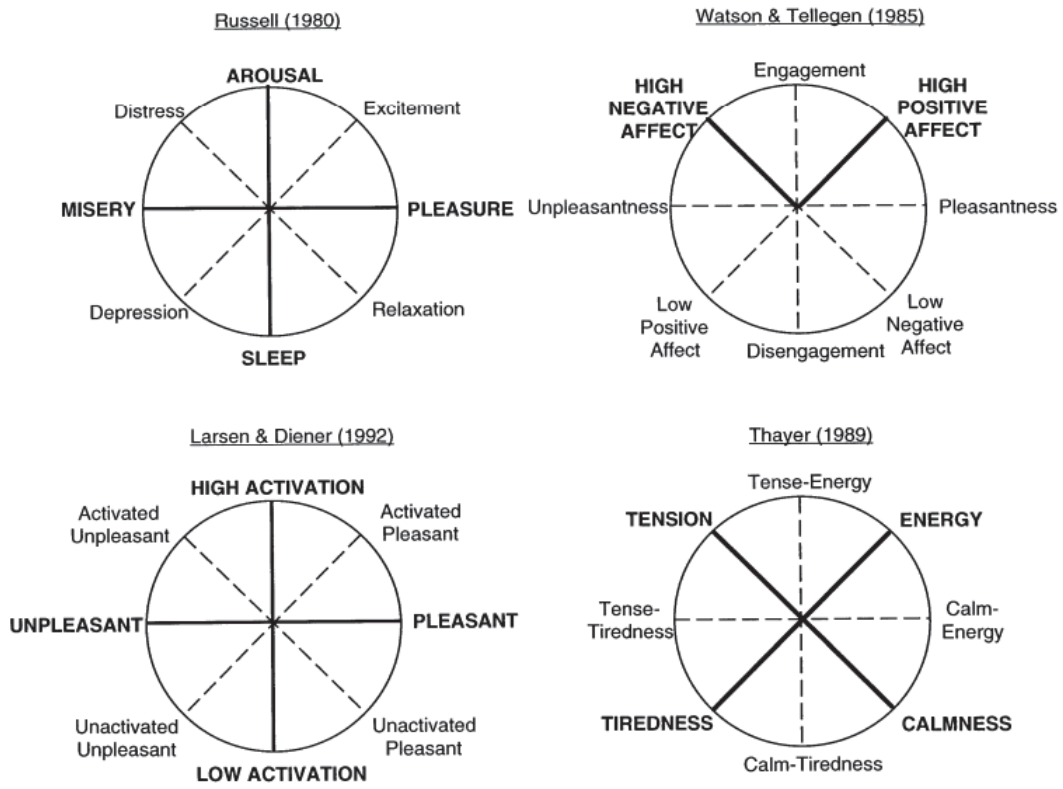


Figura 3.2: Cuatro modelos circumplejos de la afección [31].

En el diccionario usado, cada uno de los indicadores anteriores (valencia, excitación y dominación) presenta un valor comprendido entre 1 y 9. Este número se corresponde con la media de los valores obtenidos a partir de los sujetos a los que se les realizó el estudio. Se cuenta también con la desviación típica correspondiente a esa media.

3.1.3.1 Preproceso

Se ha separado el texto en palabras y se han eliminado los caracteres que no fuesen letras del alfabeto, incluyendo los signos de puntuación. Tampoco se han considerado las 313 *stopwords*[16] de la biblioteca nltk. Todos los caracteres se han convertido a minúsculas.

Para aumentar el número de coincidencias con el corpus se ha aplicado la técnica de *stemming*, ya explicada en la sección 3.1.2.1. Muchas palabras del corpus de Warriner se han unificado por tener significados e indicadores muy parecidos (por ejemplo, las palabras partícipe y participar comparten la raíz *particip*). Mediante el uso del *stemming* algunos fallos de ortografía en la terminación de las palabras han sido subsanados.

[16]Palabras más frecuentes de un determinado lenguaje.

El uso de emoticonos[17] puede ayudar en muchos casos a distinguir la polaridad de un texto. Sin embargo, no se han tenido en cuenta, ya que su presencia en los mensajes recopilados era bastante escasa (aparecían en menos de un 1 % del total de mensajes). Esto es comprensible, ya que el ámbito de la política encierra cierto grado de seriedad.

En cada uno de los *tweets* se han eliminado todos los elementos que no proporcionaban información relevante (como por ejemplo, los enlaces, los usuarios nombrados o el símbolo # de los *hashtags*). Tras realizar el *stemming* de las palabras que han pasado el filtro, se cotejan los *stems* obtenidos con los que se tienen en el corpus de afección de Warriner. Para cada mensaje se obtiene un cierto número de palabras con sus valores asociados de sentimiento.

Para calcular el valor total de cada indicador se ha supuesto que las medias del corpus de afección forman una distribución normal. Cuando presentan una desviación estándar mayor significa que se calcularon a partir de valores más dispersos. Se ha realizado una media ponderada de las palabras de las que se tienen valores de valencia, excitación y dominación, de forma que las palabras con una menor desviación típica tienen un peso mayor.

Para cada palabra se ha usado la función de distribución de una normal para estimar la probabilidad de que el valor que se tiene esté justo en la media, teniendo en cuenta su desviación típica. Estas probabilidades constituyen los pesos de cada palabra en la media ponderada citada anteriormente. Este método para el cálculo de los indicadores afectivos de cada *tweet* ha sido tomado de la tesis de Siddarth Ramaswamy [32].

La función de densidad de probabilidad de la normal usada toma como media y desviación típica los valores del diccionario de Warriner. Esta función es la siguiente:

$$f(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

donde μ es la media y σ la desviación típica.

Si consideramos que cada *tweet* tiene n palabras y que el valor resultante de aplicar la fórmula anterior es M_n, entonces tendremos que el valor total T para cada uno de los indicadores se obtiene de la siguiente forma:

$$T = \sum_{i=1}^{n} \sigma_i w_i$$

siendo w_i

$$w_i = \frac{M_i}{\sum_{i=1}^{n} M_n}$$

Se han normalizado los pesos para asegurar que el valor total obtenido se encuentra en el rango adecuado.

[17] Secuencia de caracteres que intentan representar una cara humana que expresa una emoción.

Por ejemplo, uno de los *tweets* recogidos de un usuario (que respondía a un mensaje del candidato de UPyD) tenía el siguiente texto: *@ramonmarcos @PedroChamonL @UPyD ejemplo a seguir*. Tras procesarlo y aplicarle el *stemming* se obtuvieron las siguientes raíces:*ejempl* y *segu*, ambas en el diccionario afectivo. A continuación, se muestra cómo se ha calculado la valencia total en este mensaje (en los casos de la excitación y la dominación se realiza de forma análoga).

La palabra *ejemplo* tiene los valores $\mu_1 = 2{,}0$ y $\sigma_1 = 4{,}9$, mientras que la palabra *seguir* tiene $\mu_2 = 1{,}05$ y $\sigma_2 = 4{,}82$. Para estos valores de la media y la desviación típica se calculan $M_1 = f(\mu_1, \mu_1, \sigma_1)$ y $M_2 = f_2(\mu_2, \mu_2, \sigma_2)$. Los valores obtenidos redondeados son: $M_1 = 0{,}2$ y $M_2 = 0{,}38$. Con ellos se obtienen los pesos para realizar la media ponderada: $w_1 = \frac{M_1}{M_1+M_2}$ y $w_2 = \frac{M_2}{M_1+M_2}$. El valor de estos pesos es $w_1 = 0{,}34$ y $w_2 = 0{,}66$. Aplicando la fórmula expuesta anteriormente para el cálculo de la valencia total se obtiene $T = 4{,}85$. Se observa que el valor se aproxima más a μ_2, ya que $\sigma_2 < \sigma_1$.

3.1.3.2 Clasificación

Los valores anteriores de valencia, excitación y dominación proporcionan información acerca del sentimiento de los diferentes *tweets*. Sin embargo, Pang et al. [33] desaconsejan el análisis de sentimientos basado únicamente en un diccionario léxico, ya que es menos efectivo que usar aprendizaje automático. Para poder aplicar aprendizaje automático se han usado los valores de valencia más significativos (los menores y los mayores). Este conjunto de entrenamiento es el que se emplea para obtener un clasificador. Sería preferible etiquetar los *tweets* de forma manual, pero no se disponían de recursos para hacerlo. Este método mixto que se basa en un etiquetado automático para confeccionar el conjunto de entrenamiento está inspirado en un artículo de Bing Liu et al. [34] en el que se propone un método semi-automático para etiquetar muestras para el entrenamiento de un clasificador.

Los resultados que se obtienen con las premisas anteriores son muy dependientes del corpus Warriner, ya que es el que ha determinado los *tweets* positivos y negativos que se usan como entrenamiento. Para poder obtener medidas de error fiables, que no dependiesen del corpus afectivo utilizado, se ha realizado un etiquetado manual de un conjunto muy reducido de los *tweets*. De esta forma, se pueden obtener medidas de error aproximadas.

Se ha empleado Naïve Bayes (explicado en detalle en la sección 2.6) para realizar la clasificación. La etiqueta de *tweet* neutro ha sido eliminada, ya que introducía mucho ruido.

El modelo que se obtiene funcionará bien con los *tweets* escogidos, ya que los textos del entrenamiento pertenecen al conjunto que se va a analizar. El problema de este método es que los *tweets* usados para el entrenamiento mantendrán su etiqueta, por lo que se están fijando ya muchos resultados. Como los resultados extremos se obtuvieron de la valencia, esto está justificado. De esta forma, se refinan los

resultados neutros de los que no se tenía información anteriormente y se puede etiquetar el resto de los *tweets*. Es importante escoger un número de *tweets* de entrenamiento óptimo, ya que en caso de ser un valor muy elevado no tendría sentido la clasificación, pues estaríamos fijando de antemano el sentimiento de casi todos los mensajes. En caso de ser un valor muy pequeño, el clasificador apenas aprendería, lo que tampoco sería positivo. Con esta técnica se produce sobreajuste, pero esto afecta a los resultados, ya que no se pretenden analizar textos que no sean contestaciones de los usuarios a los políticos.

3.2 Evaluación de los resultados

En esta sección se desarrollan las medidas de error tenidas en cuenta a la hora de verificar las herramientas desarrolladas.

3.2.1 Detección de temas

Para probar la eficacia de las categorías encontradas con la herramienta de *clustering* desarrollada era necesario contar con un corpus etiquetado, donde cada texto perteneciese a una categoría. La mayoría de los corpus incluidos en la biblioteca *nltk* están en inglés y aquellos que se encuentran en español suelen contener obras literarias que carecen de utilidad en el contexto de este trabajo, ya que los *tweets* que se analizan son textos cortos e informales.

Se descartó confeccionar un corpus de forma manual, ya que su tamaño debía ser considerable para conseguir resultados significativos. Como la mayoría de los artículos publicados sobre el tema usan unos corpus determinados, el uso de otros distintos impide la comparación de los resultados obtenidos. Por ello, se escogieron los corpus *Brown* y *Reuters* para realizar las pruebas, ya que son los más extendidos en el campo de la minería de textos.

El corpus Brown fue el primer corpus con más de un millón de palabras y lo confeccionó la Universidad de Brown en 1961. Está formado por 500 textos divididos en 15 géneros. El corpus Reuters consta de 10788 noticias sumando un total de 1.3 millones de palabras. Sus textos están divididos en 90 categorías. Este corpus es el más usado actualmente para la detección automática de temas.

Debido a la naturaleza informal y mucho más breve de los *tweets*, las pruebas que se hicieron con estos dos corpus no eran concluyentes. Sin embargo, sí resultaron muy útiles a la hora de depurar la herramienta y comparar los resultados con los de otros estudios. Cabe destacar que apoyarse demasiado en los resultados que se obtienen con un corpus no es recomendable, ya que entonces puede aparecer *overfitting*. Esto provocaría que la herramienta trabajase muy bien con un tipo determinado de textos, pero mucho peor con otros.

3.2.1.1 Medidas empleadas

Para comprobar la exactitud de las categorías encontradas se ha usado una medida que compara las categorías encontradas con las que venían previamente dadas en el corpus. Para poder obtener esta medida es necesario realizar una correspondencia entre los temas obtenidos y los que se encuentran inicialmente en el corpus. Para cada tema obtenido se comprueba el número de textos de cada uno de los temas originales que contiene. El mayor valor ha sido considerado como el indicador para definir las correspondencias con los temas originales.

Los resultados de la exactitud en corpus de prueba estándar son porcentajes bajos que a simple vista no parecen buenos, pero son similares a los que se especifican en algunos artículos consultados que usaban los mismos corpus [35].

También se ha calculado el error cometido en las matrices aplicando la norma de Frobenius a la matriz de los residuos. Al factorizar se obtenían las matrices W y F, cuya matriz producto $A = W \cdot F$ aproxima a la matriz M ($A \approx M$). Como ya se mencionó en la sección 3.1.2.2, la matriz del error de reconstrucción es $E = M - A$ y cuantifica el error cometido a la hora de aproximar M en la factorización. Este valor se puede relativizar con respecto a la norma de la matriz A, reduciendo de esta manera su dependencia de A. Esta medida se ha denominado error neto relativo a A:

$$E_n = \frac{||E||_F}{||M||_F}.$$

El problema que se observa con el error E_n es que, debido al gran tamaño de las matrices con las que se trabaja, unos pocos datos aislados pueden introducir mucho ruido.

Esta medida no es suficiente para determinar el error cometido en la factorización, ya que es muy susceptible de tomar valores elevados en casos en los que ciertos elementos poseen un error muy grande, aunque el resto de elementos tengan un error bajo. Para refinar el valor anterior se calculó el error promedio de todos los términos de la matriz. Este error se calcula a partir de la matriz de residuos relativos R.

$$R = r_{ij} \quad \text{donde } r_{ij} = |m_{ij} - a_{ij}| \quad \forall m_{ij} \in M, \quad \forall a_{ij} \in A$$

A partir de esta matriz se realiza un promedio por el número total de términos en la matriz (en nuestro caso $l = n \cdot k$).

$$E_p = \frac{\sum_{\forall i,j} r_{ij}}{l^2}$$

Se ha usado la norma matricial de Frobenius a la hora de calcular las normas matriciales especificadas anteriormente. Esta norma es una extensión del concepto de norma vectorial a las matrices. Dada una matriz A de orden $n \times k$, su norma

matricial de Frobenius se calcula de la siguiente forma:

$$||A||_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{k}|a_{ij}|}$$

3.2.1.2 Pruebas para determinar el valor de los parámetros

Uno de los mayores inconvenientes del método NMF es que hay que escoger previamente el número c de *clusters* que se quieren calcular. La elección óptima de c para un determinado corpus es un problema *NP-hard* [36]. Existen varios algoritmos para estimar un posible valor de c pero no se ha profundizado en ello. El valor de c se le ha asignado manualmente a la herramienta realizada. Se han realizado varias pruebas para estimar el valor óptimo para cada corpus.

Al realizar un proceso de *clustering*, el algoritmo desarrollado necesita que se le asignen tres parámetros: el número de *clusters* que se quieren obtener, el porcentaje máximo de frecuencia de aparición de una palabra entre los textos y el número mínimo de textos en que debe aparecer una palabra para ser considerada relevante.

El valor óptimo de estos tres parámetros depende del corpus empleado. Se realizaron varias pruebas para comparar los resultados obtenidos al variar estos parámetros.

Las pruebas se realizaron usando el corpus Brown. Por tanto, los resultados no son extrapolables al corpus de *tweets*, pero sí son bastante útiles para hacerse una idea general de cómo puede afectar al resultado la modificación de esos parámetros. Gracias a estas pruebas, ha sido más sencillo determinar los valores de los parámetros en el caso de los *tweets*. Además, se pretendía realizar una herramienta general que pudiese aplicarse a todo tipo de textos y no sólo a *tweets*. De esta forma, es mucho más general y puede reutilizarse en otros ámbitos.

La medida que se ha tenido en cuenta para determinar el error es la precisión, es decir, que los *clusters* resultantes coincidan con la clasificación inicial. Como el número de categorías del corpus Brown es de 15, parece lógico pensar que el número óptimo de *clusters* debe coincidir. Para comprobar la precisión en otros casos donde el número de *clusters* escogido no coincide, se ha considerado que cada categoría encontrada se corresponde con el *cluster* original más frecuente entre sus textos.

Un valor de precisión de 1 indica que el *cluster* es perfecto (idéntico a la clasificación inicial). En la Tabla 3.1 se muestran los valores de precisión obtenidos variando el número de *clusters* y tomado un porcentaje máximo de 80 % y el número de textos mínimo por palabra es de 5.

Tabla 3.1: Pruebas para determinar el número de *clusters* con los parámetros $Min = 5$ y $Max = 80\,\%$

Número de *clusters*	Precisión
5	0.262
10	0.304
14	0.374
15	0.402
16	0.358
20	0.356
25	0.324

Analizando la Tabla 3.1 se observa que el mejor resultado obtenido coincide con el número de categorías existentes, como era de esperar. Cabe destacar que esto no siempre es así, especialmente cuando se tienen muchas categorías en el corpus con el que se trabaja.

En la Tabla 3.2 se incluyen los resultados de precisión dejando fijo el número de categorías y variando los valores de Max y Min.

Tabla 3.2: Pruebas para determinar los valores Min y Max usando 15 *clusters*

Min	Max en %	Precisión
2	95	0.356
2	85	0.356
4	90	0.398
4	80	0.4
5	85	0.402
5	80	0.402
6	80	0.324
7	85	0.37
10	85	0.36

Se observa que al aumentar el porcentaje máximo por encima del 85 % no se obtiene una mayor precisión, sino que se reduce. Las diferencias entre el 80 % y el 85 % no parecen muy elevadas, por lo que se debería optar por el menor, ya que la cantidad de palabras que se computan es menor. En cuanto al número mínimo de palabras, se puede ver que en este caso un valor de 5 da buen resultado. En caso de trabajar con un mayor número de textos sería necesario que este valor aumentara lentamente. Sin embargo, con textos cortos esto no es recomendable, ya que se corre el riesgo de que muy pocas palabras consigan pasar el filtro. En el caso de

los *tweets* no sería recomendable asignar un valor mínimo superior a 5. Para textos con un número considerable de palabras se ha creado una heurística basada en las pruebas realizadas para estimar el valor del mínimo que se debe utilizar. Este valor es $log(n)^{37/21}$, donde n es el número de textos del corpus. Desafortunadamente, esta heurística no es válida cuando se tienen textos muy cortos.

También se han realizado pruebas para estudiar cómo se comporta el algoritmo a la hora de trabajar con corpus de mayor tamaño donde se tienen muchas más categorías. Para ello, se ha utilizado el corpus Reuters, que consta de 10788 textos. Se ha tomado un porcentaje máximo del 85 % y un número de palabras mínimo de 54 que funciona bastante bien y es cercano al valor que se obtendría con la heurística anterior. Los resultados de esta prueba se muestran en la Tabla 3.3.

Tabla 3.3: Pruebas para determinar el número de *clusters* usando los parámetros $Min = 54$ y $Max = 85\,\%$ en el corpus Reuters

Número de *clusters*	Precisión
2	0.42
8	0.58
14	0.496
15	0.49
16	0.49
30	0.44
90	0.24

De los resultados mostrados en la Tabla 3.3 se deduce que el número de categorías no coincide con el número óptimo de *clusters*, ya que al aumentar el número de categorías el error que se comente al factorizar las matrices aumenta. Además, el tiempo que se emplea en la factorización también es mayor, por lo que es desaconsejable usar un número muy elevado de *clusters*, aún teniendo en el corpus etiquetado una gran cantidad de categorías. En este caso el mejor resultado se consigue sólo con 8 *clusters*, pero es obvio que hay una gran diferencia entre estas 8 clases y las 90 iniciales. Por tanto, se concluye que cuando se quiera detectar un gran número de categorías la herramienta desarrollada no arrojará resultados fiables.

3.2.1.3 Análisis de la técnica TF*IDF

Se han realizado pruebas para analizar la mejora en el resultado aplicando la técnica TF*IDF. Se ha visto que la mejora es sustancial en muchos casos. Como se ha explicado en la sección 3.1.2.2, se ha empleado el logaritmo de la TF[18].

[18]Frecuencia de aparición de cada palabra. En inglés, *term frecuency*

Los resultados que aparecen a continuación se han obtenido usando el corpus Reuters con los valores $Min = 22\,\%$, $Max = 80\,\%$ y 15 *clusters*.

- Con log(TF)*IDF (el usado en la herramienta desarrollada): 0.416
- Con TF*IDF: 0.495
- Con TF (sin usar IDF): 0.32
- Con TF*IDF (con TF relativa): 0.26
- Con log(TF)*IDF (con TF relativa): 0.31

El uso de una TF relativa produce peores resultados, ya que resta importancia a aquellas palabras que aparecen con frecuencia en un mismo texto. No usar la técnica IDF también produce resultados mucho peores. El único caso que se comporta mejor es cuando no se emplea el logaritmo de la TF. Esto implica que las palabras que aparecen mucho tienen un peso mucho mayor, con la contrapartida de que se pierde importancia en otras palabras significativas que aparecen menos.

Si realizamos una comparativa de ambas usando el corpus Brown se puede comprobar cómo el resultado es mucho mejor usando el logaritmo. Se han usado 15 *clusters*, $Min = 25$ y $Max = 80\,\%$:

- Con TF*IDF: 0.28
- Con log(TF)*IDF: 0.402

Se concluye que, de acuerdo a las pruebas realizadas, la técnica usada parece ser la que proporciona mejores resultados.

3.2.2 Análisis de sentimientos

Se ha desarrollado una herramienta que permite clasificar textos atendiendo a su polaridad. Las diferentes medidas que se han usado para evaluar el clasificador entrenado se definen a continuación. Estas medidas se aplican sobre los resultados obtenidos tras clasificar todas las instancias del conjunto de pruebas.

- *Exactitud*[19]*:* Proporción de resultados correctamente clasificados.
- *Precisión*[20]*:* Probabilidad de que un documento (escogido aleatoriamente) devuelto por el clasificador sea relevante.
- *Recall*: Probabilidad de que un documento relevante (escogido aleatoriamente) sea devuelto por el clasificador.

[19]En inglés, *accuracy*.
[20]En inglés, *precision*.

- *F-measure*: Media armónica de *precision* y *recall* $\left(\frac{2 \cdot recall \cdot precision}{recall + precision}\right)$.

Se han efectuado diversas comprobaciones para verificar que las técnicas de preproceso empleadas eran las adecuadas.

A la hora de realizar estas pruebas se ha optado por utilizar un corpus en inglés de la biblioteca *nltk* que estaba etiquetado específicamente para realizar análisis de sentimientos[21]. Se ha tomado esta decisión debido a la falta de textos cortos e informales etiquetados en castellano.

Se ha dividido el número de textos en dos, dedicando un 75 % al entrenamiento y un 25 % a las pruebas. Sin realizar ningún preprocesado se obtienen resultados bastante razonables. La exactitud obtenida es del 72.8 %. Las personas sólo están de acuerdo en la polaridad de un texto el 80 % de las veces [37], por lo que refinar las medidas por encima de ese valor no es una tarea primordial.

Las medidas obtenidas para la herramienta en el corpus Movie Reviews se muestran en la Tabla 3.4.

Tabla 3.4: Medidas de error para el análisis de sentimientos en el corpus Movie Reviews.

	Textos positivos	Textos negativos
Precision	0.98	0.476
Recall	0.65	0.96
F-measure	0.78	0.64

En la prueba ha obtenido una exactitud de 0.728. Los resultados obtenidos son aceptables, pero hay bastante margen de mejora, ya que se pueden tomar algunas consideraciones que mejorarían mucho los resultados. En este trabajo no se han desarrollado debido a que no se contaba con un corpus etiquetado y el modelo obtenido iba a depender mucho del corpus afectivo. Además, como el modelo obtenido sólo se aplica en nuestro corpus de *tweets*, no era necesario refinar más el preproceso.

Un posible mejora consistiría en el uso de bigramas, pues proporcionan resultados mejores, ya que en muchos casos palabras que tienen un sentido positivo (como por ejemplo *quiero*) pueden adquirir un matiz negativo (como en el caso del bigrama (*no*,*quiero*)). También se podrían usarse trigramas pero hay estudios que indican que los resultados obtenidos con bigramas son mejores [16].

[21] Corpus *Movie Reviews*, que consta de 2000 textos clasificados según su polaridad.

Capítulo 4

Resultados

En este capítulo se detalla el proceso de recolección de los tweets de los candidatos estudiados para conformar el corpus donde se aplicaron las herramientas desarrolladas en el Capítulo 3. Los resultados obtenidos se exponen en tablas y figuras para facilitar su comprensión.

4.1 Estudio de las cuentas de los candidatos

El primer paso consistió en recabar información sobre las cuentas de Twitter de los candidatos sujetos a estudio. Los seis candidatos escogidos pertenecían a los partidos políticos de mayor importancia. Al comienzo de este estudio, ni el candidato de Podemos ni el del PSOE contaban con una cuenta personal de Twitter. Afortunadamente, ambos se crearon una cuenta antes de empezar el proceso de extracción, lo que permitió obtener información de todos los partidos. La cuenta del PSOE no era una cuenta personal del candidato, sino de su equipo.

Una vez se habían empezado a recopilar los *tweets* se creó una cuenta del equipo de la candidata del PP, que ya contaba con cuenta personal en Twitter. Por ello, se decidió añadirla al estudio, puesto que aún no se había recopilado un gran número de mensajes. Por lo tanto, se han analizado 7 cuentas de Twitter cuya información básica se muestra a continuación[1]:

- **PP-1:**
  - **Nombre:** Cristina Cifuentes
  - **Usuario:** @ccifuentes

[1]Estos datos fueron tomados el 10-03-2015, exceptuando el caso de la cuenta de Cifuentes Candidata, cuya información se recogió el 24-03-2015.

  - **Descripción:** Delegada del Gobierno en Madrid, aunque este Twitter es personal. Mis opiniones aquí son sólo mías.
  - **Página web:** No figura.

- **PP-2:**
  - **Nombre:** Cifuentes candidata
  - **Usuario:** @CCifuentes2015
  - **Descripción:** Cuenta de la candidatura de @ccifuentes a la Comunidad de Madrid por el Partido Popular. Contamos con todos. #Nosgustamadrid
  - **Página web:** `cristinacifuentes.es`

- **Ciudadanos:**
  - **Nombre:** Ignacio Aguado
  - **Usuario:** @ignacioaguado
  - **Descripción:** Candidato de @CiudadanosCs a la Presidencia de la Comunidad de Madrid.
  - **Página web:** `ignacioaguado.es`

- **IU:**
  - **Nombre:** Luis García Montero
  - **Usuario:** @lgm_com
  - **Descripción:** Poeta y Catedrático de Literatura Española en la Universidad de Granada.
  - **Página web:** `http://luisgarciamontero.com/`

- **UPyD:**
  - **Nombre:** Ramón Marcos Allo
  - **Usuario:** @ramonmarcos
  - **Descripción:** Candidato de @UPyD a la presidencia de la Comunidad de #Madrid. Desde 2011, diputado autonómico. Antes, Letrado de la Seguridad Social.
  - **Página web:** `ramonmarcos.com`

- **Podemos:**
  - **Nombre:** Jose Manuel López
  - **Usuario:** @JoseManuel_Lop
  - **Descripción:** Ingeniero agrónomo, antiguo director general de Pluralismo y Convivencia. Consejero Ciudadano de @podemosmad. Es la hora de la gente.
  - **Página web:** `https://www.facebook.com/pages/Jos%C3%A9-Manuel-L%C3%B3pez/1558312534447897`
- **PSOE:**
  - **Nombre:** #EsGabilondo
  - **Usuario:** @equipoGabilondo
  - **Descripción:** Cuenta Oficial de la candidatura de Ángel Gabilondo @PSMadrid @PSOE a la Presidencia de la Comunidad de Madrid #EsGabilondo.
  - **Página web:** `https://www.facebook.com/gabilondo.pormadrid`

Algunas cuentas introducen el uso de *hashtags* en los nombres o las descripciones. Estos nombres pueden ser cambiados a lo largo de la campaña y este mecanismo fue utilizado por el PSOE, que fue modificando su nombre del usuario según el *hashtag* que quería promover en cada momento.

También se recogieron datos numéricos de las cuentas al principio y al final de la extracción de los *tweets* para así poder observar su evolución a lo largo de la campaña. Los datos fueron recabados el 12-03-2015, exceptuando los de la cuenta de Cifuentes Candidata (PP-2), que fueron tomados el 24-03-2015. Todos estos datos fueron extraídos de los perfiles de los distintos candidatos. En la Tabla 4.1 se muestran los datos antes de la extracción de los *tweets*, mientras que los de la Tabla 4.2 se obtuvieron tras terminar la recolección de los *tweets*, concretamente el 17/05/2015. Los datos que se han recogido son los siguientes:

- *Tweets:* Número de mensajes publicados que aparece en su perfil de Twitter. Este número contabiliza también los *retweets* y las contestaciones a otros usuarios.
- *Siguiendo:* Número de usuarios a los que sigue el candidato.
- *Seguidores:* Número de *followers* del candidato.
- *Favoritos:* Número de *tweets* marcados como favoritos por el candidato.

- *Registro:* Año en que se registró la cuenta en Twitter.

Tabla 4.1: Datos de las cuentas de los políticos antes de la extracción de los *tweets*

Cuenta	***Tweets***	**Siguiendo**	**Seguidores**	**Favoritos**	**Registro**
PP-1	28240	600	70345	9	2008
PP-2	286	1079	2895	247	2015
Ciudadanos	3267	1974	6791	1547	2010
IU	1296	616	21314	32	2009
UPyD	15630	1512	5249	1578	2009
Podemos	6	386	1009	0	2015
PSOE	509	998	4781	165	2015

Tabla 4.2: Datos de las cuentas de los políticos tras la extracción de los *tweets*

Cuenta	***Tweets***	**Siguiendo**	**Seguidores**	**Favoritos**	**Registro**
PP-1	29145	664	79794	22	2008
PP-2	2341	2968	9423	1308	2015
Ciudadanos	3475	1980	8808	1804	2010
IU	2062	744	24631	532	2009
UPyD	16757	1815	7069	2050	2009
Podemos	336	552	8101	22	2015
PSOE	1861	3050	13939	303	2015

Analizando las tablas 4.1 y 4.2 se observa que algunos candidatos ya contaban con cuentas de Twitter con bastantes seguidores y las han aprovechado para aumentar la difusión de sus mensajes. Otras cuentas, sin embargo, fueron creadas al inicio de la campaña, lo que limitaba bastante la difusión de sus *tweets*, ya que contaban con un menor número de seguidores.

Para visualizar la evolución de los candidatos a lo largo del tiempo que abarca este estudio, se ha calculado el incremento a partir de las tabas anteriores desde el 12-03-2015 (exceptuando la cuenta Cifuentes Candidata de la que se empezaron a obtener datos el 24-03-2015) al 17-05-2015. Estos cálculos aparecen reflejados en la Tabla 4.3.

Tabla 4.3: Incremento de los datos de las cuentas de los políticos

Cuenta	***Tweets***	**Siguiendo**	**Seguidores**	**Favoritos**
PP-1	905	4	9449	13
PP-2	2055	1889	6528	1061
Ciudadanos	148	6	2017	257
IU	766	128	3317	500
UPyD	1127	303	1820	472
Podemos	330	166	7092	22
PSOE	1352	2142	8158	138

Algunas cuentas de los candidatos tuvieron un gran incremento de usuarios a los que seguían. Esto puede deberse a que decidieron seguir a gente para que les correspondieran, un fenómeno bastante común en las redes sociales. También podría ser que cuando alguien les empezaba a seguir, los candidatos le seguían de vuelta a esa persona como agradecimiento, ya que esto transmite una mayor sensación de cercanía. Este mismo motivo también explicaría el incremento en los favoritos, que también fue elevado dado el corto período de tiempo que abarca este estudio.

4.2 Extracción de *tweets* de los candidatos

Para la extracción de los *tweets* se ha utilizado el programa realizado en Python para tal fin descrito en la sección 3.1.1. Para conseguir los mensajes de cada político y las respuestas del resto de los usuarios se empleó el lenguaje de consultas de la Search API de Twitter. Gracias a ellas se realizaban distintas descargas en función de los candidatos y el tipo de *tweet* que fuese. Muchas de estas consultas eran redundantes, ya que esa información estaba ya recogida en ciertos campos del *tweet*. Sin embargo, dado que sólo se podían extraer *tweets* con una semana de antigüedad (debido a las limitaciones de la API de Twitter), se prefirió tomar datos de más y después cotejar algunos de los campos para comprobar su consistencia. Las consultas realizadas vienen recogidas en la tabla 4.4.

Tabla 4.4: Consultas para la extracción de los datos

Consulta	**Datos extraídos**
from:POLÍTICO -filter:retweets -filer:replies	*Tweets* del político
from:POLÍTICO filter:retweets	*Retweets* del político
from:POLÍTICO filter:replies	Contestaciones del político
to:POLÍTICO filter:replies	Contestaciones de los usuarios
from:POLÍTICO filter:images filter:videos -filter:retweets -filer:replies	Imágenes y vídeos del político
from:POLÍTICO filter:links -filter:retweets -filer:replies	*Tweets* con enlaces del político

Las dos últimas consultas devuelven muchos *tweets* que se obtienen también en las anteriores. Además, algunos resultados de estas dos últimas consultas no eran del todo consistentes, por lo que al final esas características[2] no se utilizaron en el análisis, ya que no eran del todo fiables y tampoco aportaban demasiada información añadida.

Las consultas se realizaron para cada una de las siete cuentas analizadas. Por tanto, en cada extracción se obtenían 42 ficheros con los distintos *tweets*. Estas extracciones se planificaron a lo largo de la campaña electoral para abarcar un espacio de tiempo considerable y tener mensajes de todos los días bajo estudio. Una vez recopilados todos los ficheros (462 en total), se procedió a su integración, conformando de esta forma el corpus sobre el que se ha trabajado.

Como sólo se podían obtener los *tweets* de los últimos siete días, cada vez que se ejecutaba el programa encargado de recopilar los *tweets*, era necesario realizar extracciones periódicas para obtener el mayor número posible. La Search API de Twitter tampoco devuelve todos los *tweets* que están englobados dentro de cada una de las consultas. A pesar de estas limitaciones, al realizar la extracción de manera repetida a lo largo del tiempo de pre-campaña se ha logrado conformar un corpus de un tamaño considerable que es válido a la hora de ser analizado.

Las fechas en las que se ha ejecutado el programa extractor de *tweets* se encuentran recogidas en la tabla 4.5. Cada día de extracción se recopila información de hasta 7 días antes.

[2]Presencia de enlaces, imágenes y vídeos en los *tweets*.

Tabla 4.5: Fechas de extracción de los datos

Fecha de la extracción	**Fechas de los datos extraídos**
14-03-2015	Del 7-03-2015 al 14-03-2015
19-03-2015	Del 12-03-2015 al 19-03-2015
24-03-2015	Del 17-03-2015 al 24-03-2015
30-03-2015	Del 23-03-2015 al 30-03-2015
06-04-2015	Del 30-03-2015 al 06-04-2015
13-04-2015	Del 06-04-2015 al 13-04-2015
19-04-2015	Del 12-04-2015 al 19-04-2015
26-04-2015	Del 19-03-2015 al 26-04-2015
02-05-2015	Del 25-04-2015 al 02-05-2015
09-05-2015	Del 02-05-2015 al 09-05-2015
16-05-2015	Del 09-05-2015 al 16-05-2015

Algunos días de extracción se superponen, por lo que algunos de los *tweets* fueron recogidos más de una vez (esto también sucede porque algunas de las consultas no eran excluyentes entre sí). Gracias a que cada tweet consta de un número de identificación único, se detectaron cuáles eran los repetidos y se unificaron.

Para obtener una visión global de los datos, se ha representado en un gráfico de barras el número de *tweets* existente en cada uno de los días analizados. De esta forma, la Figura 4.1 permite visualizar cómo se distribuyen los *tweets* a lo largo del tiempo.

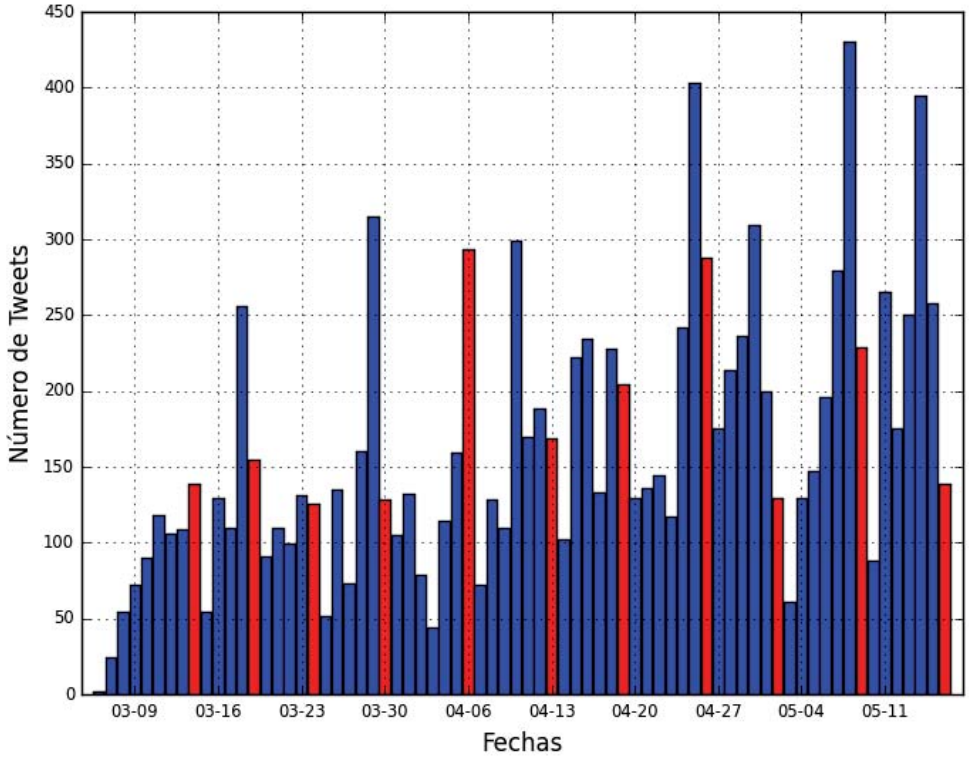


Figura 4.1: Distribución de todos *tweets* con las fechas de extracción marcadas en rojo

En la Figura 4.1 se observa que el número de *tweets* por día no es nada uniforme y que se suelen presentar picos los días anteriores a la extracción. Algunos días de extracción también presentan picos y esto es debido a que se realizaron por la noche, por lo que ya se había escrito la mayor parte de los *tweets* de ese día. Por tanto, la herramienta de extracción recoge menos *tweets* cuanto más lejanos en el tiempo se encuentren de la fecha de extracción, por lo que los mínimos suelen estar en los días siguientes a las extracciones. Los días en los que las extracciones se realizaron temprano también muestran un número bajo de *tweets*. Debe suponerse que existieron fluctuaciones en cuanto al número de *tweets* que había cada día, pero no es lógico que éstas sean tan pronunciadas como las que aparecen en la Figura 4.1. Analizando la Tabla 4.6 se comprueba que el número de *tweets* obtenidos se aproxima bastante al valor real, por lo que en ese caso la distribución no debería ser tan desigual. Por tanto, se puede afirmar que el desajuste en la distribución está causado por los *replies* de los usuarios a los políticos, que se toman en fechas cercanas a las extracciones y que, al ser más numerosos, desequilibran el gráfico.

Si no contabilizamos los replies en la Figura 4.1, se observa que este comportamiento anómalo alrededor de las fechas de extracción ya no es tan evidente (Figura 4.2). También se aprecia mejor el incremento del número de *tweets* a lo largo de la campaña electoral. Los máximos y mínimos también son más fiables, pues no son

tan dependientes de las fechas de extracción.

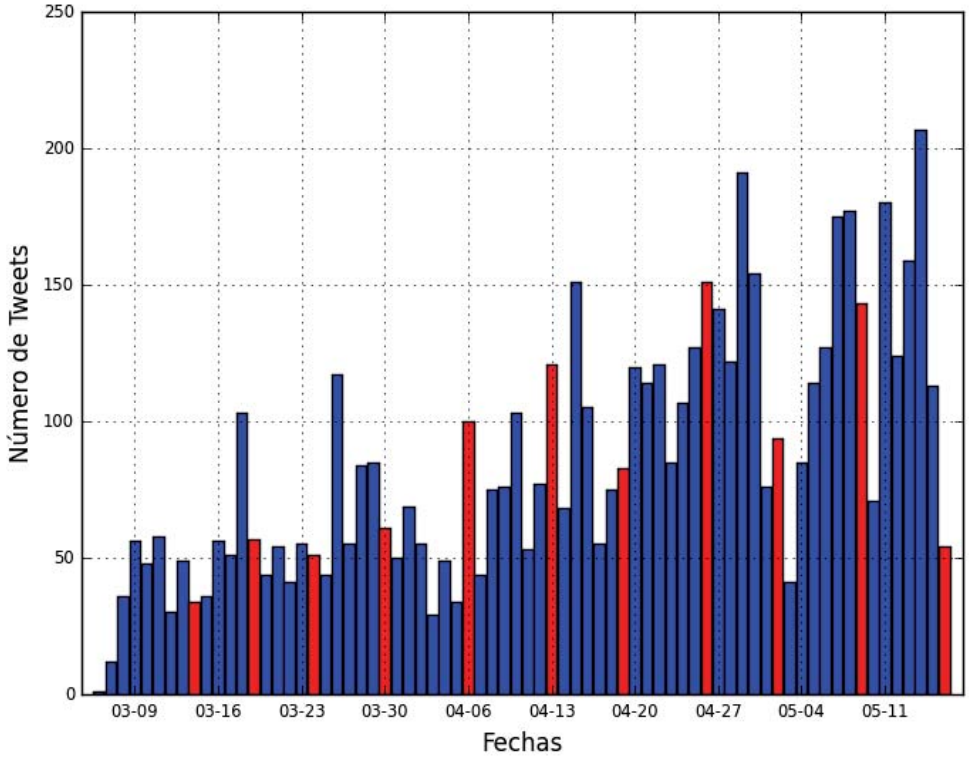


Figura 4.2: Distribución de los *tweets* de los políticos con las fechas de extracción marcadas en rojo

Para confirmar que el ruido en la Figura 4.1 se debe a los *replies*, se han representado sólo las contestaciones de los usuarios en la Figura 4.3. Se observa que la distribución de los *replies* depende bastante de las fechas de extracción. Aunque el número de contestaciones de los usuarios depende del número de los *tweets*, una variación tan marcada es anómala[3]. Por tanto, se puede afirmar que el número de contestaciones era mayor al recopilado en nuestro corpus y que los *tweets* que no se han almacenado estarían en su mayor parte situados en los días siguientes a las extracciones.

[3]Por ejemplo, en el día 5-20 (posterior a un día de extracción) en las Figuras 4.2 y 4.3, se comprueba que, aunque el número de *tweets* de los políticos era normal, el de los *replies* era inusitadamente bajo.

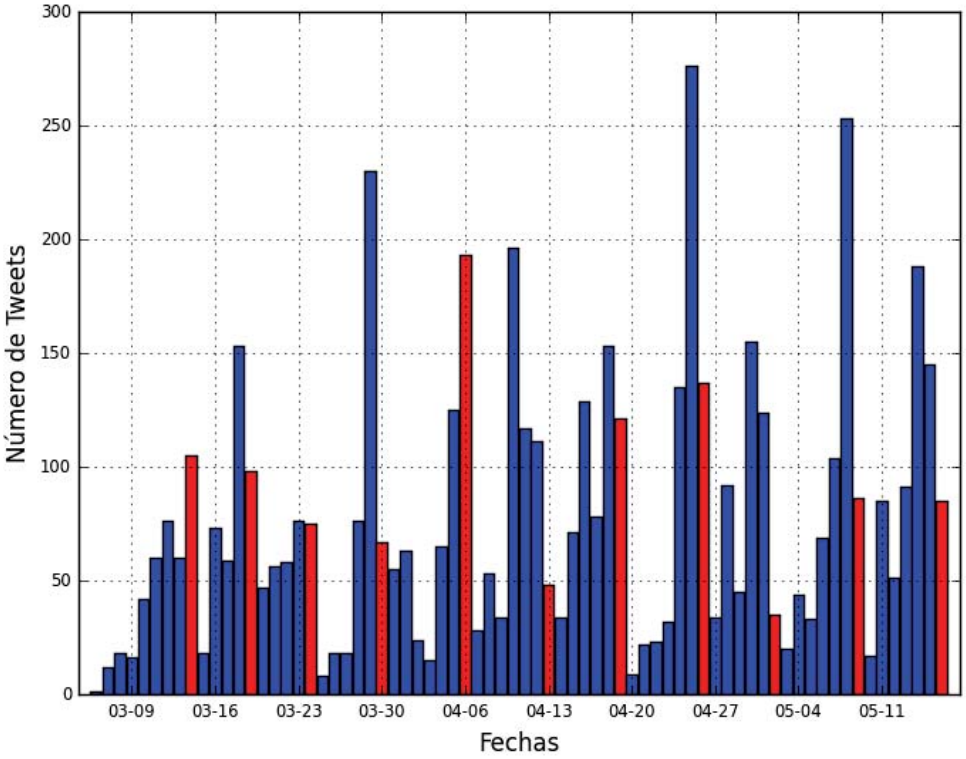


Figura 4.3: Distribución de los *replies* de los usuarios a los políticos con las fechas de extracción marcadas en rojo

4.3 Análisis del corpus

Tras estudiar la distribución en el tiempo de los datos y confirmar que se contaba con un corpus razonablemente consistente en ese aspecto, se pasó a analizar las distribuciones de *tweets* en función de los candidatos políticos.

En la Tabla 4.3 se tiene el número de *tweets* totales de cada candidato que se corresponde con el valor observado de la Tabla 4.6. Este valor se ha comparado con el número de datos que se tienen en el corpus confeccionado, que se ha denominado valor del corpus. Como se observa en la Tabla 4.6, ambos valores son similares, pero no idénticos, ya que los primeros *tweets* que se tienen son del 7-03-2015[4], mientras que los datos de la Tabla 4.3 se tomaron más tarde. Esto explica que, en algunos casos, el valor real mayor sea mayor. Sin embargo, en otras cuentas el valor es menor. Esto se debe a que la extracción con la API Search de Twitter no es uniforme ni exhaustiva, por lo que algunos *tweets* pueden no haberse recopilado.

[4]En realidad, se cuenta con algunos *tweets* del 6-03-2015, pero su número es muy pequeño. Esto se debe a que las horas de extracción no eran fijas y a veces incluían algunas horas de días que no están contemplados en la Tabla 4.5

Tabla 4.6: Comparativa del número de *tweets* del corpus y del observado

	PP-1	**PP-2**	**Ciudadanos**	**IU**	**UPyD**	**Podemos**	**PSOE**
Valor observado	905	2055	148	766	1127	330	1352
Valor del corpus	941	1608	246	712	1248	325	1083

La estructura del corpus final conformado, que es el que se ha usado a la hora de obtener todos los datos y sobre el que se han aplicado las herramientas desarrolladas, se resume en la Tabla 4.7, que se muestra a continuación.

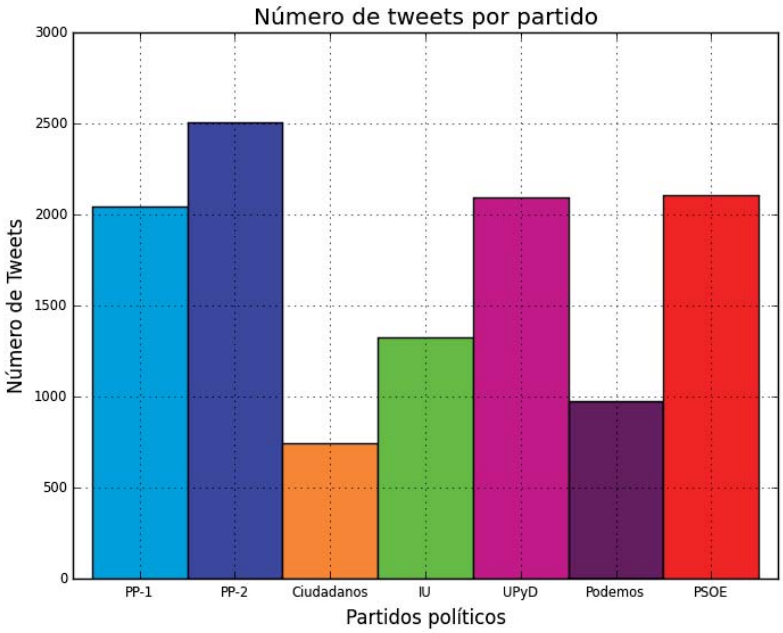
Tabla 4.7: Composición del corpus recopilado.

	Tweets	***Replies***	***Retweets***	***Replies* de usuarios**	**Total**
PP-1	266	256	419	1102	2043
PP-2	891	191	526	897	2505
Ciudadanos	77	63	106	499	745
IU	395	40	277	611	1323
UPyD	424	146	678	845	2093
Podemos	215	7	103	649	974
PSOE	904	2	177	1022	2105
Total	3172	705	2286	5625	11786

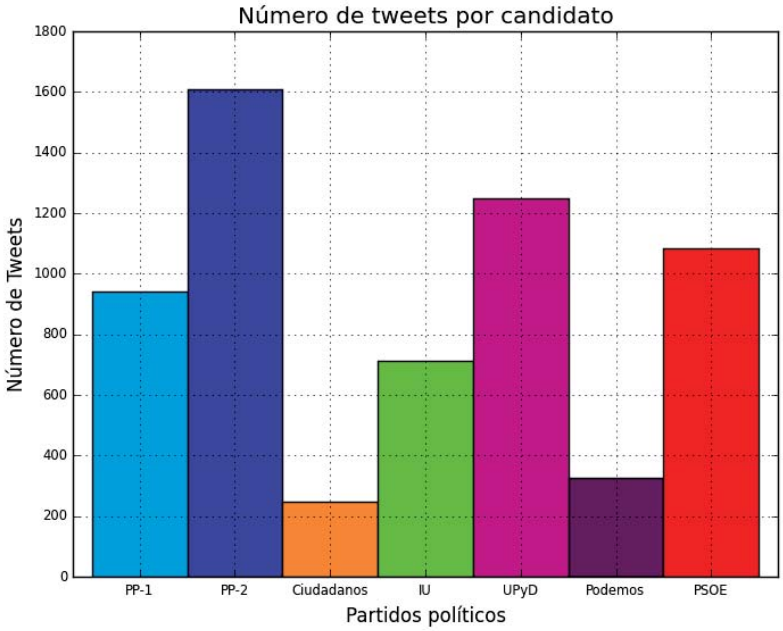
La suma del número de los usuarios de los distintos partidos (11788) no se corresponde con el número total de *tweets* que aparecen en el cuadro (11786). Esto se debe a que en algunos casos los políticos interactúan entre ellos, por lo que un mismo tweet puede encontrarse en la columna de *replies* de un partido y también en la columna de *replies* de los usuarios. Esto tan sólo sucede en dos ocasiones (un *reply* de Ignacio Aguado a Cristina Cifuentes y otro de José Manuel López también a Cifuentes). Cuando se analizan los *tweets* específicos de un partido, se consideran estos *tweets* como *replies* del candidato y también como *replies* de los usuarios, por lo que se encuentran duplicados. Sin embargo, cuando se trabaja con los *tweets* totales sólo se considera que es un *reply* de un candidato y no contabiliza como *reply* de un usuario para el candidato que estaba siendo repondido.[5] Como los identificadores de los *tweets* son únicos, sólo pueden ser contabilizarlos una vez para calcular el número total de *tweets*, lo que explica que el número total de mensajes del corpus sea 11786.

[5]En realidad hay 19 casos en los que el autor de la respuesta y el destinatario se encuentran entre los políticos analizados. Sin embargo, en 17 de estos casos el político se contesta a sí mismo, por lo que no cuentan como respuestas de otros usuarios.

A continuación, se muestran los datos anteriores en gráficos de barras. Se ha diferenciado entre *tweets* totales, incluyendo los *replies* de los usuarios (veáse la Figura 4.4a) y *tweets* sólo de los políticos (veáse la Figura 4.4b). Su distribución es muy similar en ambos casos.



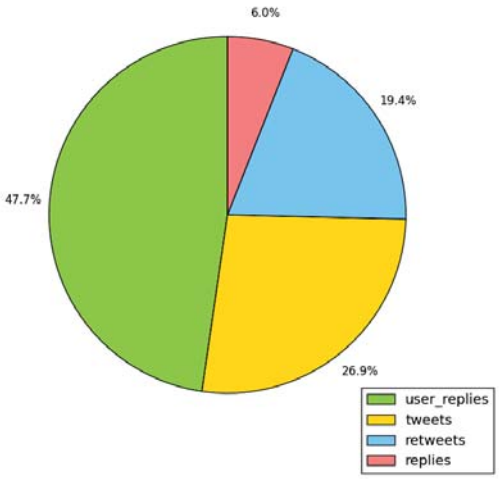
(a) Número de *tweets* totales por partido



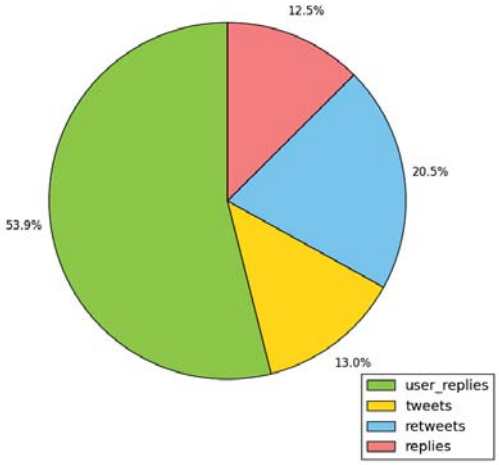
(b) Número de *tweets* de los políticos

Figura 4.4: Distribución de *tweets* por partidos

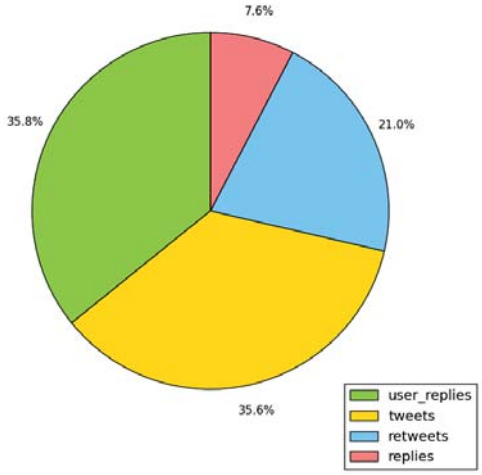
Se han realizado también gráficos circulares para visualizar la distribución de *tweets* por tipo en total (veáse la Figura 4.5a) y para cada uno de los candidatos (veáse Figura 4.5b-h). Se observa que en algunos casos (Podemos y PSOE) el número de *replies* de los candidatos es muy bajo y casi no se aprecian en los gráficos.



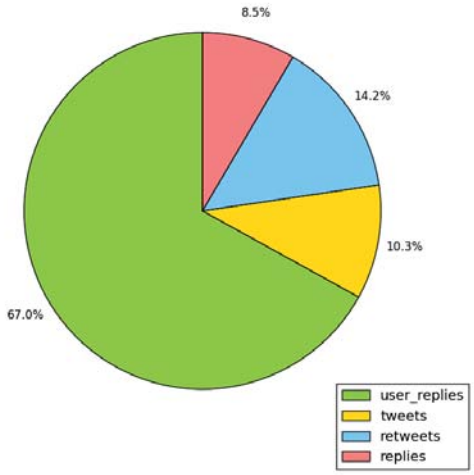
(a) Total



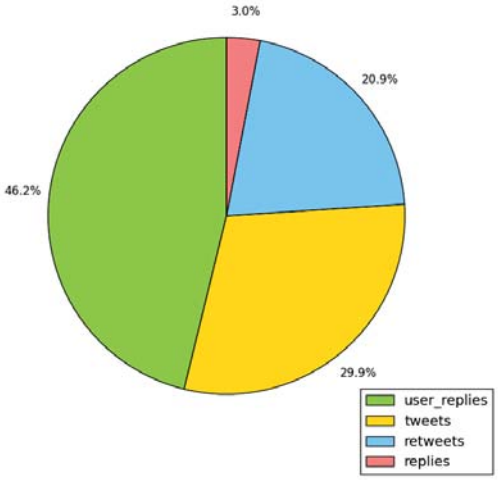
(b) PP-1



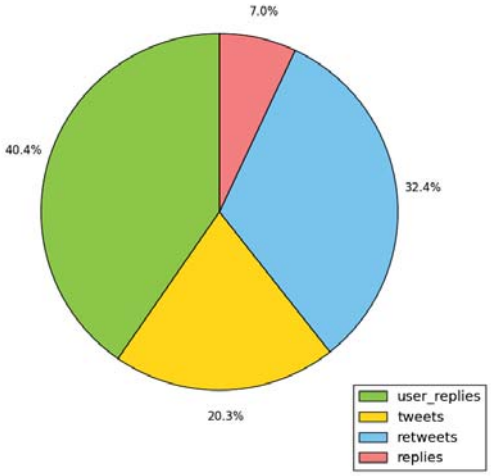
(c) PP-2



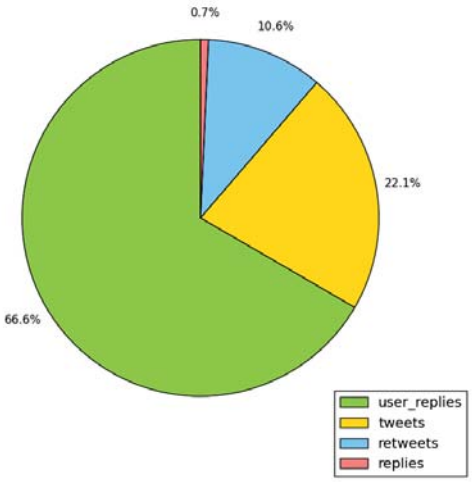
(d) Ciudadanos



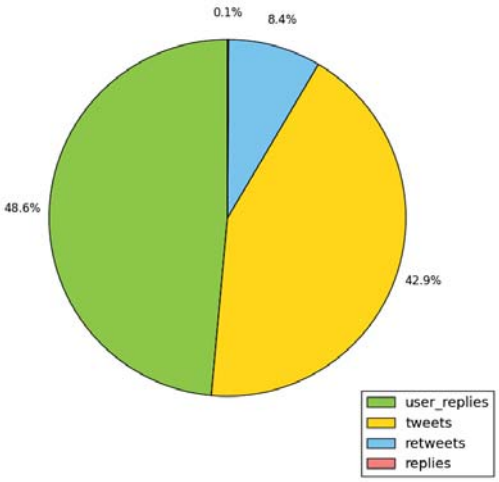
(e) IU



(f) UPyD



(g) Podemos



(h) PSOE

Figura 4.5: Distribución de *tweets* por tipo

Como en el corpus se tienen las fechas y las horas de cada *tweet*, se ha realizado un gráfico donde se muestran las horas en las que los políticos escribían sus mensajes. Para ello se ha divido el día en cuatro franjas horarias[6]:

- **Mañana:** 8:00-14:00.
- **Tarde:** 14:00-20:00.
- **Noche:** 20:00-00:00.
- **Madrugada:** 00:00-7:00.

En la Figura 4.6 se observa que todos los candidatos siguen un patrón similar a la hora de *twittear*.Las horas más activas son la mañana y la tarde. Por la noche se reduce considerablemente el número de *tweets* y de madrugada son muy escasos.

[6]La hora final de cada franja horaria no se encuentra incluida.

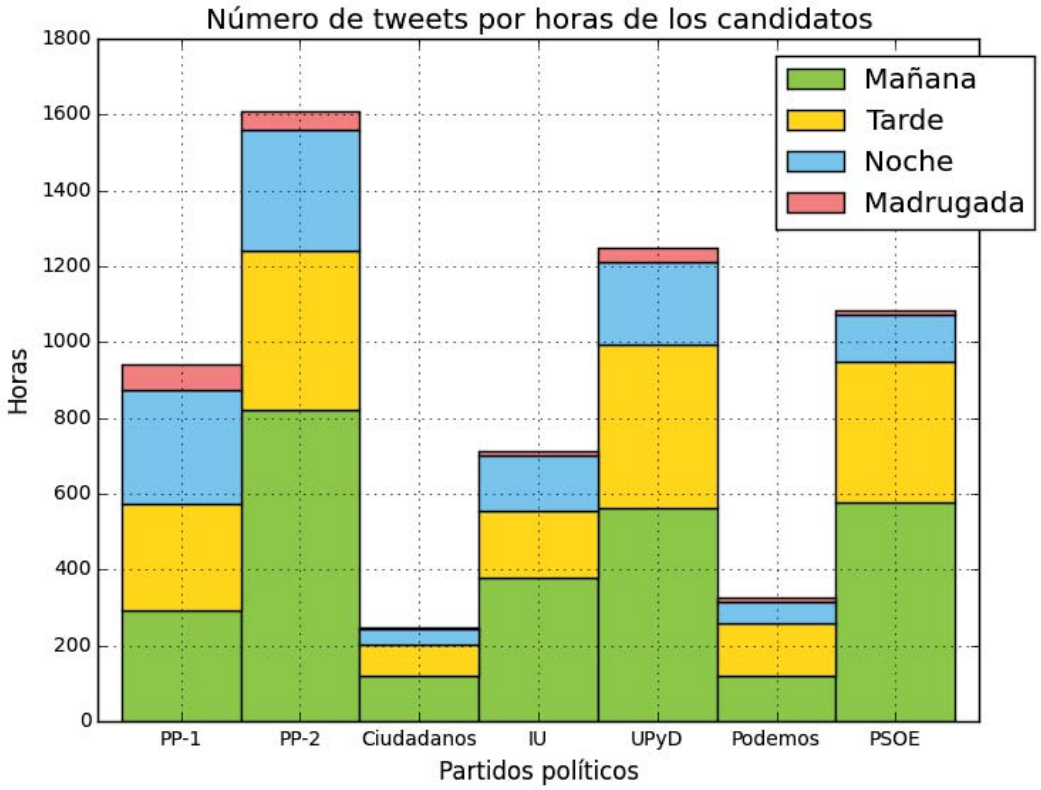


Figura 4.6: *Tweets* de los políticos según la hora de publicación

Otro elemento importante para caracterizar a cada político son los *hashtags*. Estas palabras son clave a la hora de difundir un mensaje y promover debates en la red social. Cuando un *hashtag* se hace popular, un gran número de usuarios empieza a *twittear* ese *hashtag*, por lo que tienen un gran potencial.

El número de *hashtags* empleado varía entre candidatos, ya que algunos los consideran un eje central en su campaña en la red (como por ejemplo el PSOE, cuyo nombre de usuario cambiaba en función del hashtag que se quisiera difundir) y otros apenas hacen uso de ellos. En la Tabla 4.8 se indica el número de *hashtags* empleado en los *tweets* obtenidos según el partido al que pertenezcan.

Tabla 4.8: Número de *hashtags* por partido

	PP-1	**PP-2**	**Ciudadanos**	**IU**	**UPyD**	**Podemos**	**PSOE**
Hashtags totales	541	1168	163	471	700	105	756
Hashtags distintos	237	283	94	111	229	59	95

Los datos de la Tabla 4.8 se han recogido en el gráfico de barras de la Figura 4.7. Se ha señalado en amarillo la cantidad de *hashtags* no repetidos para así po-

der visualizar la repetición de los *hashtags* por parte de los candidatos. Se observa que PP-2 y PSOE son los que mejor uso hacen de los *hashtags*, ya que los repiten bastante. El uso de *hashtags* sin repetición (los casos de Podemos y Ciudadanos) no favorece tanto su difusión.

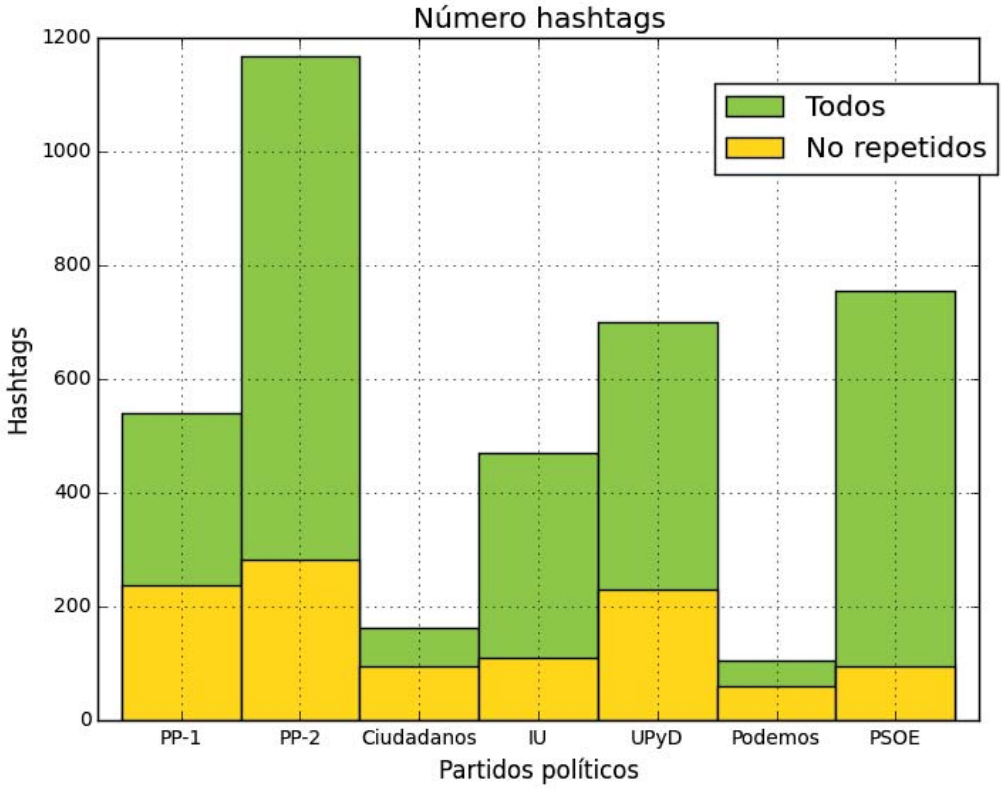


Figura 4.7: Número de *hashtags* por partido

Una forma sencilla de visualizar los *hashtags* más frecuentes es utilizando nubes de *tags* o de palabras[7]. Las nubes de tags son representaciones visuales de las palabras más frecuentes de un texto, donde las que tienen una mayor frecuencia de aparición se representan con un tamaño mayor y un color más llamativo. En la Figura 4.8 se han construido las nubes de *tags* a partir de la consecución de los *hashtags* empleados por cada una de las cuentas estudiadas. Estas nubes de palabras se realizaron usando la aplicación web WordItOut[8].

[7] En inglés, *tag cloud* o *word cloud*.
[8] `www.worditout.com/`

(a) PP-1

(b) PP-2

(c) Ciudadanos

(d) IU

(e) UPyD

(f) Podemos

(g) PSOE

Figura 4.8: Nubes de tags por partido

En las figuras anteriores se distinguen con facilidad los *hashtags* que más difundieron las distintas cuentas. Se aprecia que algunos políticos sólo poseen un hashtag que usan mucho y en torno al cual centran el debate. Aunque algunas cuentas (como las dos del PP y la del PSOE) sí usan frecuentemente más de un hashtag, estos son muy parecidos entre sí, por lo que tampoco pueden considerarse que tengan más variedad que el resto. Se concluye que ninguno de los partidos realiza un uso adecuado de los hahtags como medio para generar debates sobre temas, sino que los emplean como *slogan*, siendo bastante escaso el número de *hashtags* que pretenden abrir un debate entre la ciudadanía.

4.4 Detección de temas en los *tweets* de los candidatos

Para encontrar temas en los *tweets* de los candidatos se empleó la herramienta desarrollada en la sección 3.1.2. Se realizaron ciertos ajustes a la herramienta, ya que debido a que los nombres de los candidatos y de los partidos aparecen en multitud de *tweets*, esas palabras tenían mucho peso. Esto daba lugar a que los temas no aportasen ninguna información de utilidad. Por ello, se filtraron todas las palabras que tuvieran relación con los nombres o los partidos. Además, también se eliminaron los enlaces y las menciones a otros usuarios, puesto que era información irrelevante que generaba ruido en la búsqueda de temas.

En un primer lugar, se ejecutó la herramienta usando todos los *tweets* del corpus para determinar si era posible obtener algunos temas generales. Se buscaron 15 categorías. El valor máximo de aparición de cada palabra en los textos fue del 85 % y el valor mínimo de textos en los que debía aparecer fue de 5.

Para cada caso se han representado dos gráficas dependiendo de la aproximación usada a la hora de determinar la pertenencia de un texto a una categoría. En la Figura 4.9 se tiene en cuenta la primera aproximación y en la Figura 4.10 se tiene en cuenta la segunda.

Para cada categoría se muestra el número de textos que pertenecen a ella, divididos según su relevancia. Siguiendo la primera aproximación, la suma de todos los textos equivale al total de textos, mientras que en el otro caso suele ser superior, debido a que un texto puede estar incluido en varias categorías.

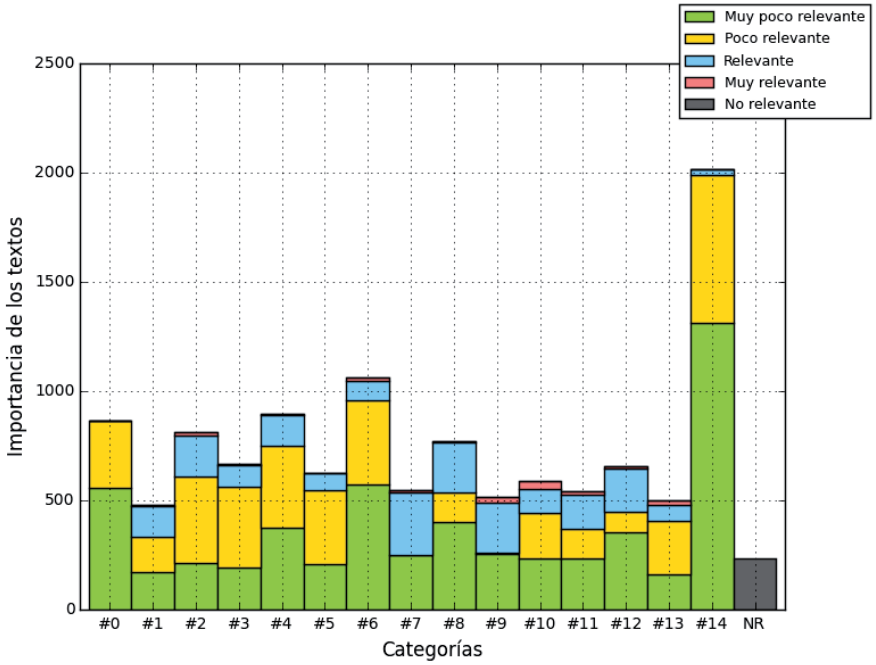


Figura 4.9: Detección de temas en todos los *tweets* usando la primera aproximación

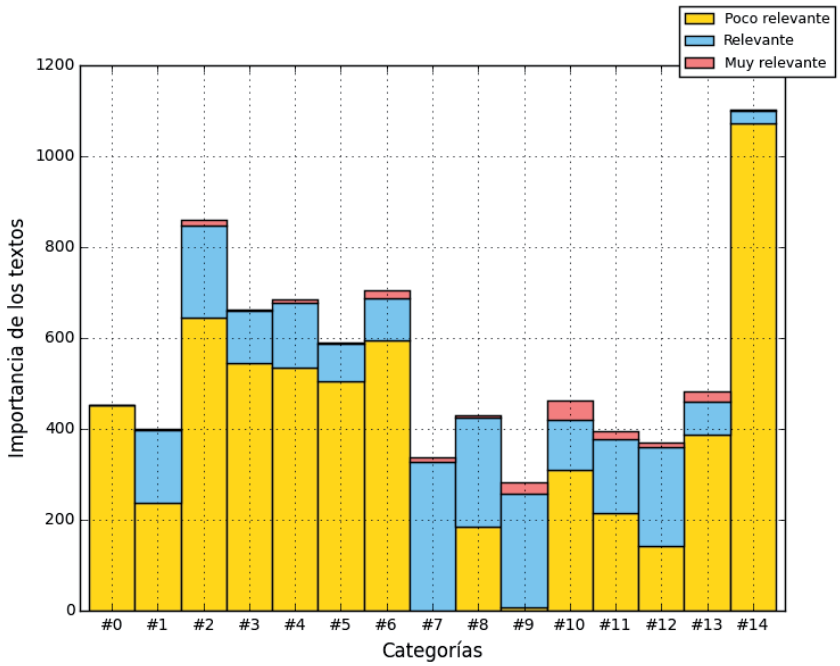


Figura 4.10: Detección de temas en todos los *tweets* usando la segunda aproximación

Las categorías encontradas tienen asociados distintos *stems* que permiten obtener una visión general del tema. Los *stems* más significativos de cada categoría, junto con sus valores de importancia, son los siguientes:

- **Categoría 0:** madr: 10.9, comun: 8.1 y candidat: 7.1
- **Categoría 1:** trabaj: 7.5, dign: 0.6 y par: 0.6
- **Categoría 2:** public: 5.9, sanid: 3.2 y educacion: 2.7
- **Categoría 3:** polit: 6.9, social: 1.9 y deb: 1.0
- **Categoría 4:** graci: 6.0, much: 3.3 y suert: 0.8
- **Categoría 5:** hac: 7.3, falt: 1.2 y campan: 0.9
- **Categoría 6:** program: 4.5, propuest: 4.2 y electoral: 2.9
- **Categoría 7:** mejor: 7.0, madrilen: 0.85 y bien: 0.8
- **Categoría 8:** part: 6.8, corrupcion: 1.2 y espanol: 1.2
- **Categoría 9:** vot: 7.4, gan: 0.7 y lueg: 0.7
- **Categoría 10:** person: 6.0, contig: 2.7 y primer: 1.6
- **Categoría 11:** impuest: 6.0, baj: 3.2 y sucesion: 3.0

- **Categoría 12:** cambi: 6.7, gobiern: 1.6 y quer: 1.3
- **Categoría 13:** cre: 6.1, emple: 3.5 y priorid: 0.7
- **Categoría 14:** pued: 4.6, quier: 3.2 y sol: 2.5

La categoría 14 es la que tiene un mayor número de textos asociados y coincide con la categoría cuya palabra más relevante tiene un valor menor. Sin embargo, la mayoría de estos textos son de poca relevancia. Por otro lado, la categoría 0 tiene valores muy elevados para las tres palabras más relevantes, por lo que es una categoría muy restrictiva y no cuenta con tantos textos relevantes.

Se pueden relacionar los distintos *stems* con los temas tratados. Sin embargo, en algunos casos no es fácil inferir un tema generalizado para los distintos *stems*, debido a que las palabras que definen el tema son demasiado generales. En estos casos, esos temas se han etiquetado como no relevantes. En la Tabla 4.9 se han especificados los distintos temas encontrados.

Tabla 4.9: Temas encontrados en todo el corpus de *tweets*

Categoría 0	Candidatos de Madrid
Categoría 1	Trabajo
Categoría 2	Sanidad y educación pública
Categoría 3	Política social
Categoría 4	Agradecimientos
Categoría 5	No relevante
Categoría 6	Programas electorales
Categoría 7	No relevante
Categoría 8	Corrupción
Categoría 9	No relevante
Categoría 10	No relevante
Categoría 11	Impuestos
Categoría 12	Cambio
Categoría 13	Empleo
Categoría 14	No relevante

Como se puede comprobar en la figuras anteriores, se han obtenido muy pocos *tweets* de gran relevancia para cada categoría. Por ello, no se profundizó más en este análisis, ya que la cantidad de *tweets* es demasiado elevada como para obtener un número satisfactorio de muestras relevantes en cada tema.

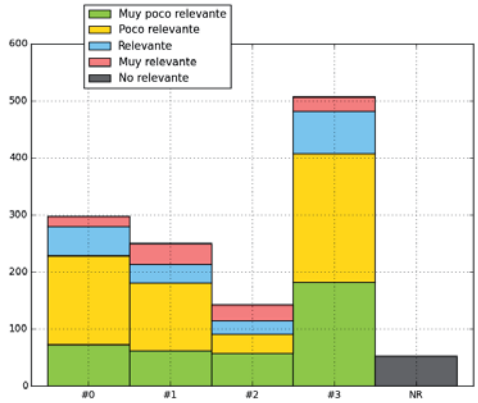
A continuación, se realizó una búsqueda automática de temas en el subconjunto de los *tweets* de los políticos. Se redujo el valor del número mínimo de textos en los que debía aparecer una palabra de 5 a 3, ya que de esta forma se obtenían mejores

resultados. Para determinar el número c de categorías, se realizaron varias pruebas con distintos valores de c. El número de temas es menor que en el caso anterior, ya que el número de *tweets* de cada político es significativamente menor. Estas pruebas para determinar el valor de c se realizaron con los mensajes del candidato de UPyD, ya que tiene 845 *tweets*, que es valor más cercano a la media del número de *tweets* de los distintos candidatos.

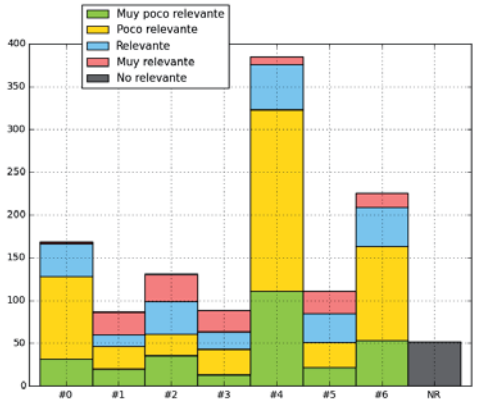
En la Figura 4.11 se ha empleado la primera aproximación para determinar la pertenencia de los *tweets* a las categorías, mientras que en la Figura 4.12 se ha utilizado la segunda aproximación. Se aprecia que los resultados son bastante mejores que en las gráficas anteriores, por lo que se puede realizar un análisis mucho más exacto de los temas que están tratando los distintos candidatos.

Los resultados para los distintos valores de c probados han sido muy parecidos. Se realizaron las búsquedas de las categorías para los distintos candidatos con $c = 10$, ya que es un valor que permite encontrar un número suficiente de categorías y con el que se obtiene un número razonable de textos para cada categoría.

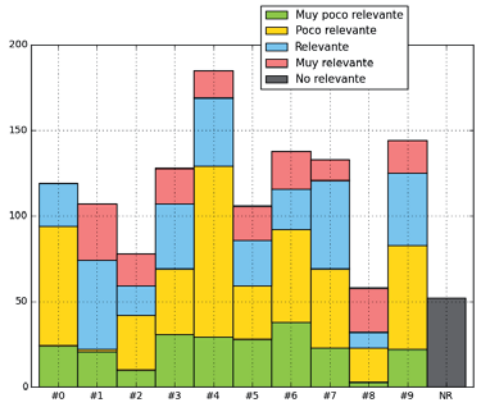
Para valores de menores de c se aprecia que algunos temas tienen una importancia desmesurada comparados con otros. En los casos $c = 10$ y $c = 13$ se observa una mayor homogeneidad. Ambos valores son perfectamente válidos, pero se eligió el valor $c = 10$ para facilitar la presentación de los resultados y porque cada categoría tiene alrededor de 100 mensajes, un número bastante razonable. Además, con $c = 13$ los *tweets* relevantes y muy relevantes por tema se reducen de forma significativa, por lo que se obtendrían temas menos significativos. Si se sigue aumentando c llegará un momento en que las categorías se diluyan demasiado y tengan un número de mensajes relevantes muy pequeño. En las figuras 4.11 y 4.12 se ha usando $c = 4$, $c = 7$, $c = 10$ y $c = 13$.



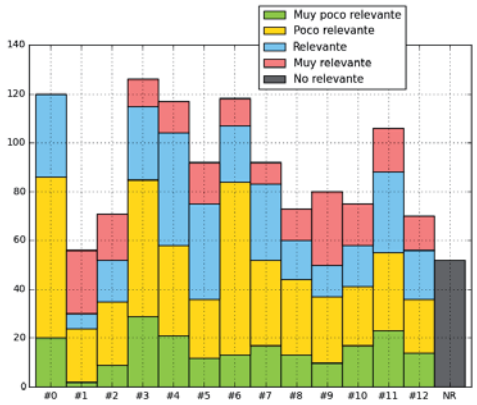
(a) $c = 4$



(b) $c = 7$

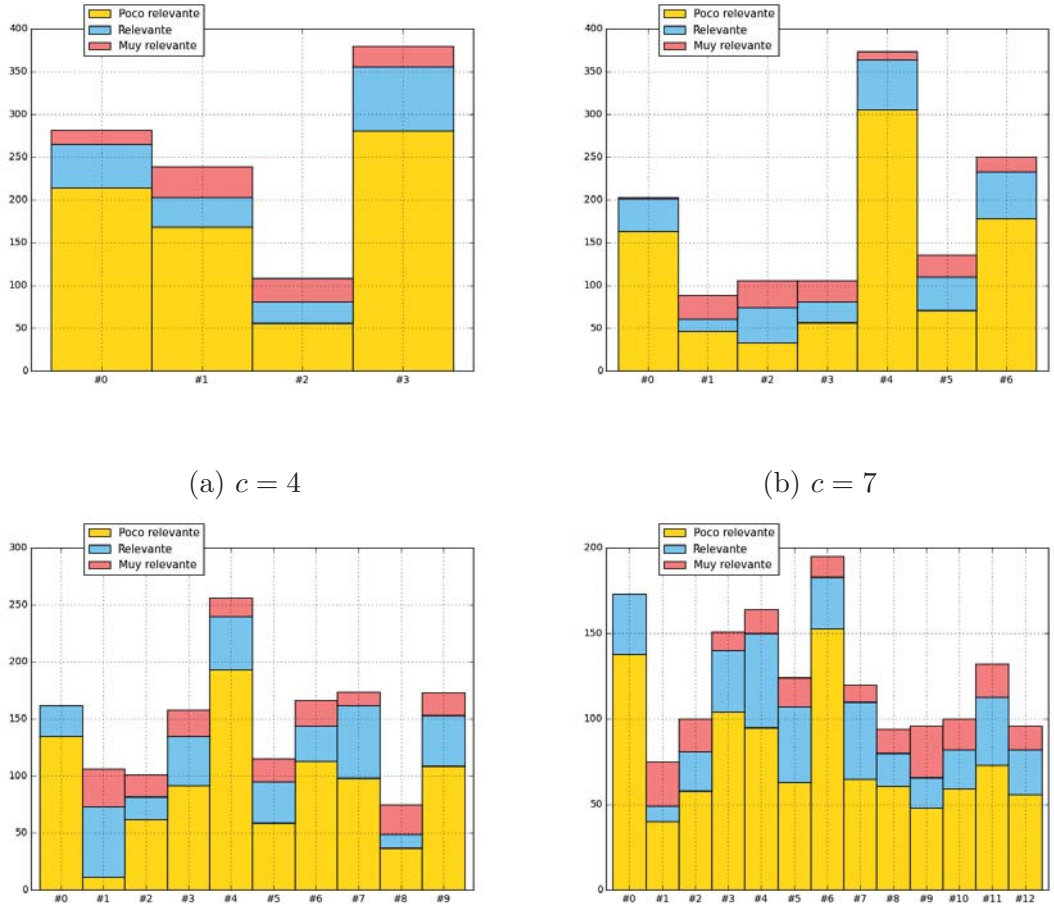


(c) $c = 10$



(d) $c = 13$

Figura 4.11: Comparativa en los resultados en la detección de temas al variar el número de *clusters* con la primera aproximación

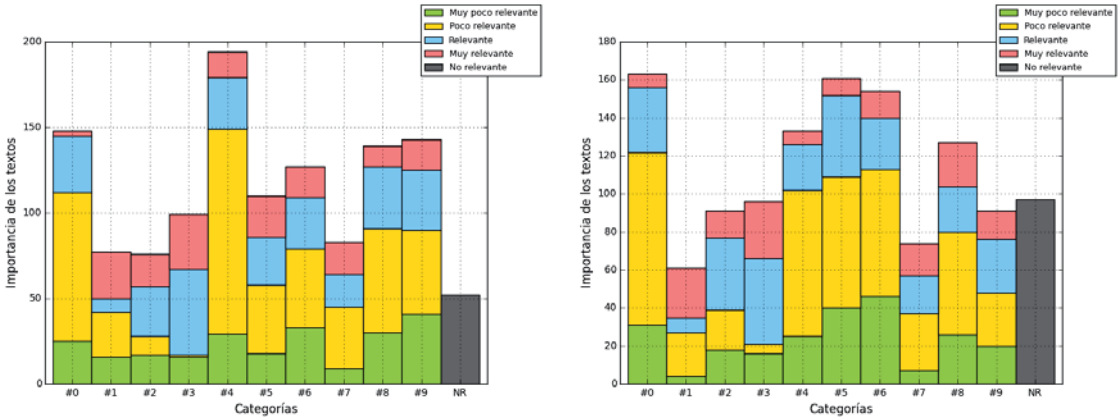


(a) $c = 4$ (b) $c = 7$

(c) $c = 10$ (d) $c = 13$

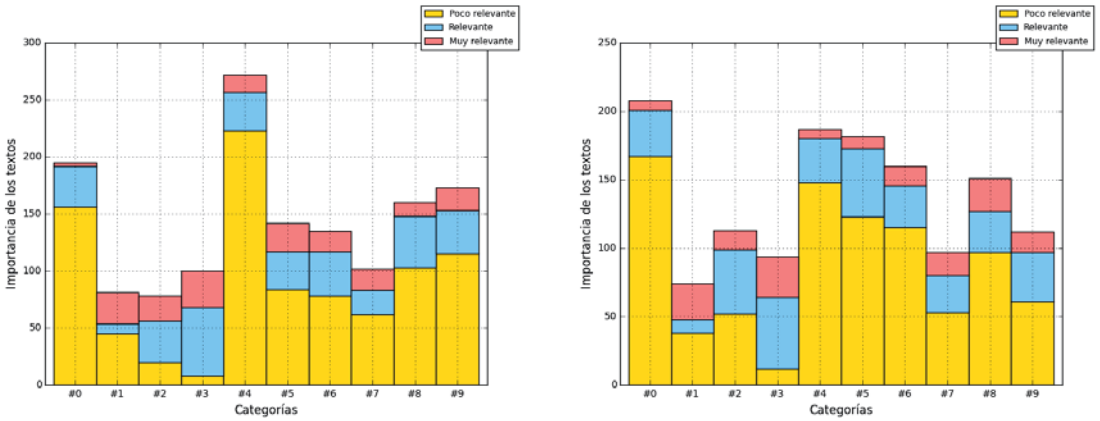
Figura 4.12: Comparativa en los resultados en la detección de temas al variar el número de *clusters* con la segunda aproximación

También se ha representado la comparativa de los resultados obtenidos asignando 3 y 5 al mínimo número de textos de aparición de las palabras significativas. Se ha utilizado $c = 10$ y el conjunto de los mensajes usados para la prueba han sido los del candidato de UPyD. En la Figura 4.13 se usa la primera aproximación, mientras que en la Figura 4.14 se usa la segunda.



(a) $min = 3$ (b) $min = 5$

Figura 4.13: Comparativa en los valores del número mínimo de textos en los que debe aparecer una palabra para ser tenida en cuenta con la primera aproximación



(a) $min = 3$ (b) $min = 5$

Figura 4.14: Comparativa en los valores del número mínimo de textos en los que debe aparecer una palabra para ser tenida en cuenta con la segunda aproximación

En la figura 4.13 se aprecia un ligero empeoramiento de los resultados al aumentar el valor mínimo, ya que el número de *tweets* no relevantes aumenta y se tiene un menor de *tweets* relevantes para cada categoría. Además, al consultar las palabras más significativas de cada categoría se observa que al usar un valor mínimo de 5 empeoran visiblemente los resultados.

Se utilizó $c = 10$ en los *tweets* de los distintos candidatos con la segunda aproximación, ya que proporciona más información y no incluye los *tweets* menos relevantes. También se generaron tablas con los *stems* más relevantes obtenidos en cada categoría. Acompañando a las figuras se incluyen tablas con el número de textos y

su relevancia en cada categoría. Se consideraron textos no relevantes aquellos cuyo valor es cero para todas las categorías y también los considerados muy poco relevantes según la primera aproximación (aquellos textos cuya categoría más significativa tenga un valor inferior a 0.1).

4.4.1 Temas en los *tweets* de PP-1

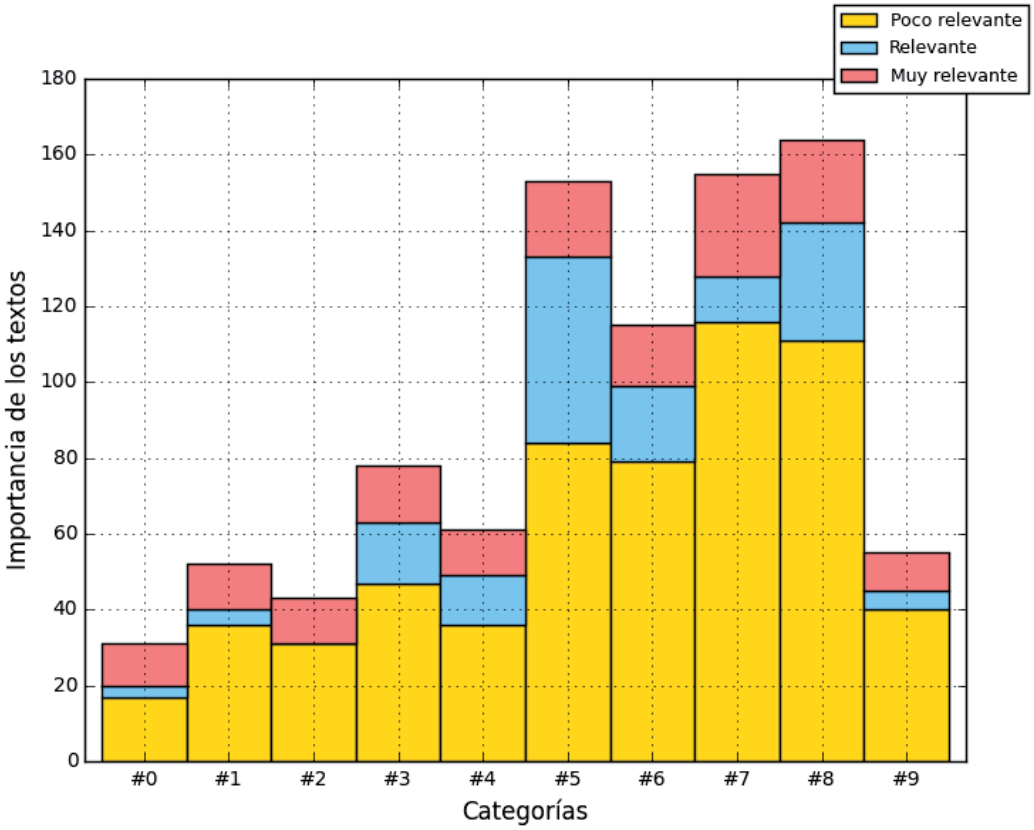


Figura 4.15: Detección de temas en los *tweets* de Cristina Cifuentes

Tabla 4.10: Número de *tweets* de Cristina Cifuentes por categoría.

	Poco relevantes	Relevantes	Muy relevantes	Total
Categoría 0	17	3	11	31
Categoría 1	36	4	12	52
Categoría 2	31	0	12	43
Categoría 3	47	16	15	78
Categoría 4	36	13	12	61
Categoría 5	84	49	20	153
Categoría 6	79	20	16	115
Categoría 7	116	12	27	155
Categoría 8	111	31	22	164
Categoría 9	40	5	10	55

Tabla 4.11: Stems más significativos de los *tweets* del equipo de Cristina Cifuentes por categoría.

Categoría 0	sucesion: 2.9	baj: 2.9	patrimoni: 2.7	irpf: 2.4	vec: 2.4
Categoría 1	serm: 2.0	hospital: 1.8	public: 1.7	integr: 1.6	empres: 1.6
Categoría 2	privatiz: 2.0	externaliz: 2.0	habr: 1.9	nuev: 1.9	sanitari: 1.4
Categoría 3	univers: 2.0	apoy: 1.9	public: 1.9	calid: 1.7	matic: 1.6
Categoría 4	segur: 2.0	junt: 1.8	local: 1.5	gobiern: 1.2	deleg: 1.2
Categoría 5	graci: 3.2	much: 1.5	mejor: 1.0	buen: 0.9	abraz: 0.9
Categoría 6	vecin: 2.7	escuch: 1.6	call: 1.6	ilusionpormadr: 1.4	villaverd: 1.4
Categoría 7	program: 2.5	electoral: 2.2	escrib: 1.7	salud: 1.5	pued: 1.1
Categoría 8	madr: 2.7	comun: 2.4	candidat: 2.2	president: 2.0	futur: 1.1
Categoría 9	tarif: 1.9	plan: 1.9	eur: 1.8	abon: 1.4	transport: 1.3

4.4.2 Temas en los *tweets* de PP-2

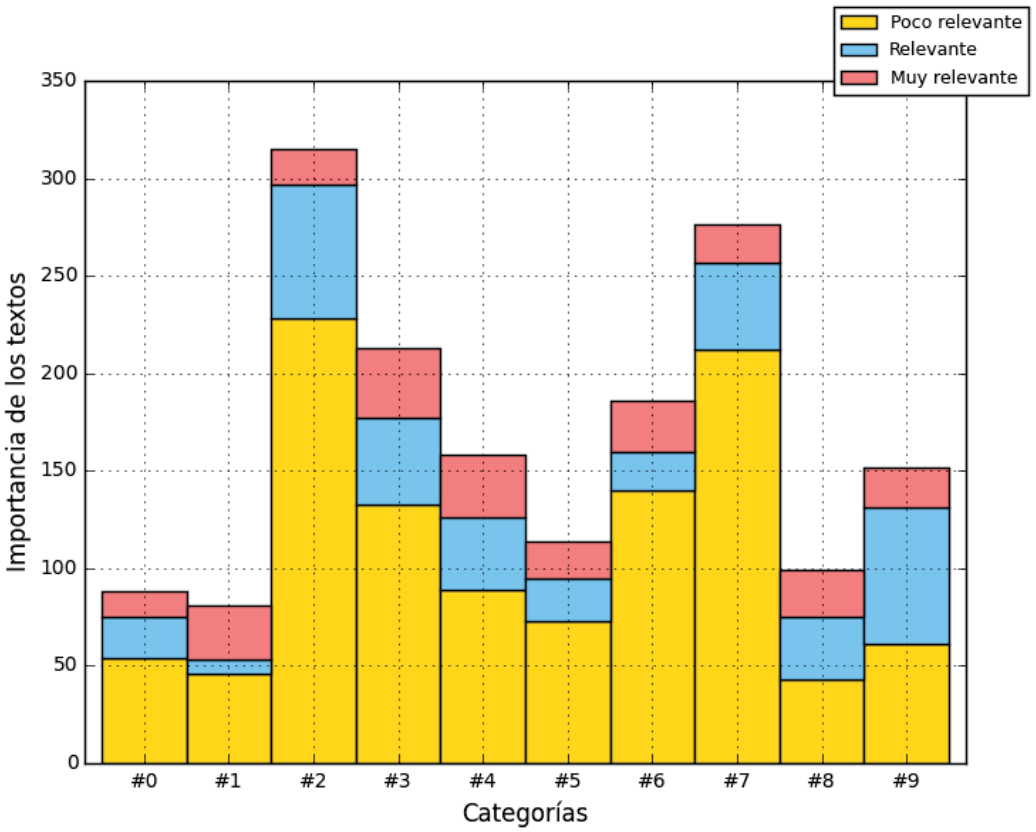


Figura 4.16: Detección de temas en los *tweets* del equipo de Cristina Cifuentes

Tabla 4.12: Número de *tweets* de Cristina Cifuentes por categoría.

	Poco relevantes	Relevantes	Muy relevantes	Total
Categoría 0	54	21	13	88
Categoría 1	46	7	28	81
Categoría 2	228	69	18	315
Categoría 3	133	44	36	213
Categoría 4	89	37	32	158
Categoría 5	73	22	19	114
Categoría 6	140	20	26	186
Categoría 7	212	45	19	276
Categoría 8	43	32	24	99
Categoría 9	61	70	21	152

Tabla 4.13: Stems más significativos de los *tweets* del equipo de Cristina Cifuentes por categoría.

Categoría 0	educacion: 3.4	padr: 3.1	libert: 3.0	eleccion: 2.2	concert: 2.2
Categoría 1	abon: 1.9	eur: 1.9	tarif: 1.7	plan: 1.7	jov: 1.6
Categoría 2	madr: 3.3	comun: 1.8	proyect: 1.4	candidat: 1.3	ilusionpormadr: 1.1
Categoría 3	program: 3.1	electoral: 2.0	propuest: 1.4	polit: 1.0	cumpl: 0.8
Categoría 4	public: 3.1	sanid: 2.5	mejor: 1.3	gratuit: 1.1	calid: 1.0
Categoría 5	famili: 3.3	numer: 2.0	hij: 1.2	discapac: 1.0	especial: 1.0
Categoría 6	periscop: 2.5	liv: 2.1	direct: 1.5	presentacion: 1.1	vecin: 1.0
Categoría 7	part: 2.0	hac: 1.9	acuerd: 1.2	campan: 1.1	lleg: 1.1
Categoría 8	list: 2.4	corrupcion: 2.0	imput: 1.4	etic: 1.3	codig: 1.3
Categoría 9	impuest: 2.7	emple: 2.4	baj: 1.7	creacion: 0.9	autonom: 0.8

4.4.3 Temas en los *tweets* de Ciudadanos

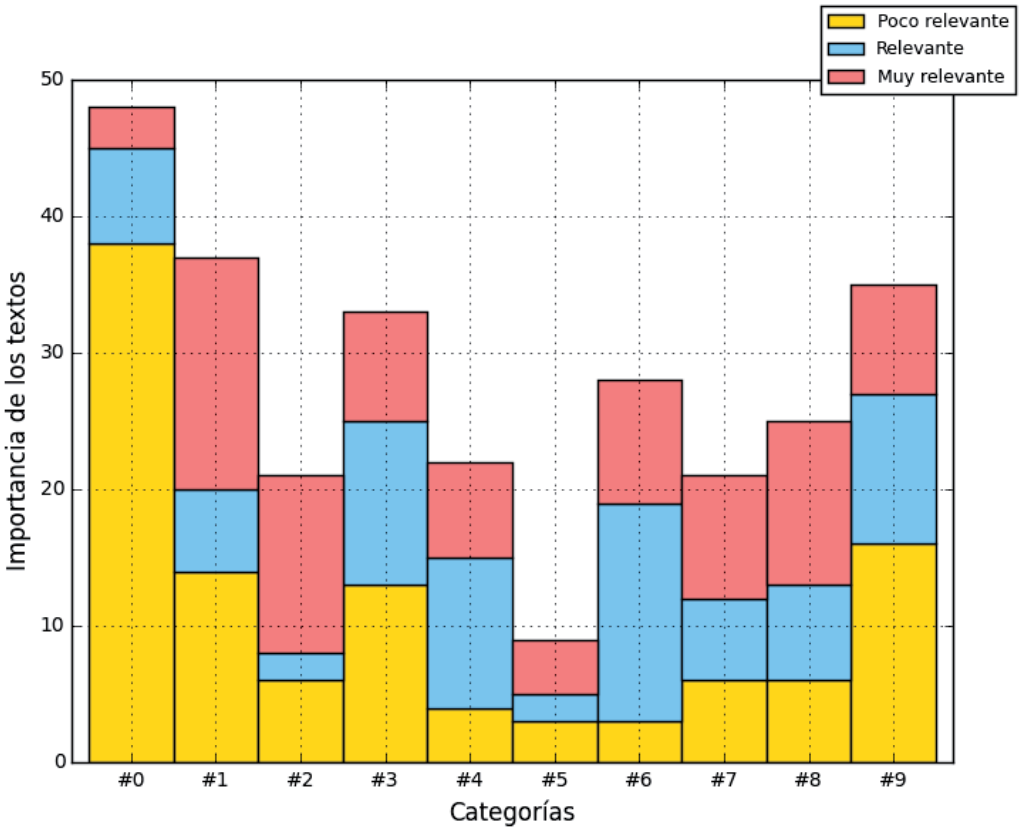


Figura 4.17: Detección de temas en los *tweets* de Ignacio Aguado

Tabla 4.14: Número de *tweets* de Ignacio Aguado por categoría.

	Poco relevantes	Relevantes	Muy relevantes	Total
Categoría 0	38	7	3	48
Categoría 1	14	6	17	37
Categoría 2	6	2	13	21
Categoría 3	13	12	8	33
Categoría 4	4	11	7	22
Categoría 5	3	2	4	9
Categoría 6	3	16	9	28
Categoría 7	6	6	9	21
Categoría 8	6	7	12	25
Categoría 9	16	11	8	35

Tabla 4.15: Stems más significativos de los *tweets* de Ignacio Aguado por categoría.

Categoría 0	comun: 3.2	candidat: 3.0	madr: 2.4	manan: 1.4	carp: 0.9
Categoría 1	graci: 1.8	much: 1.3	abraz: 1.2	pront: 1.0	vosotr: 0.9
Categoría 2	medi: 2.1	clas: 1.8	impuest: 0.9	fiscal: 0.6	ambient: 0.6
Categoría 3	propuest: 2.1	present: 1.0	econom: 1.0	materi: 0.9	fiscal: 0.7
Categoría 4	espan: 2.0	tod: 1.4	grand: 1.2	sol: 1.2	bloqu: 0.8
Categoría 5	compart: 1.8	custodi: 1.8	vot: 1.2	graci: 0.5	madr: 0.2
Categoría 6	hol: 1.8	administracion: 1.1	cre: 1.1	salud: 1.1	justici: 1.0
Categoría 7	corrupcion: 1.8	anos: 1.3	ningun: 1.1	quer: 0.9	ser: 0.5
Categoría 8	cambi: 2.3	cos: 1.5	sensat: 0.8	vot: 0.7	may: 0.4
Categoría 9	gran: 1.6	equip: 1.3	polit: 1.2	futur: 1.2	mejor: 1.0

4.4.4 Temas en los *tweets* de IU

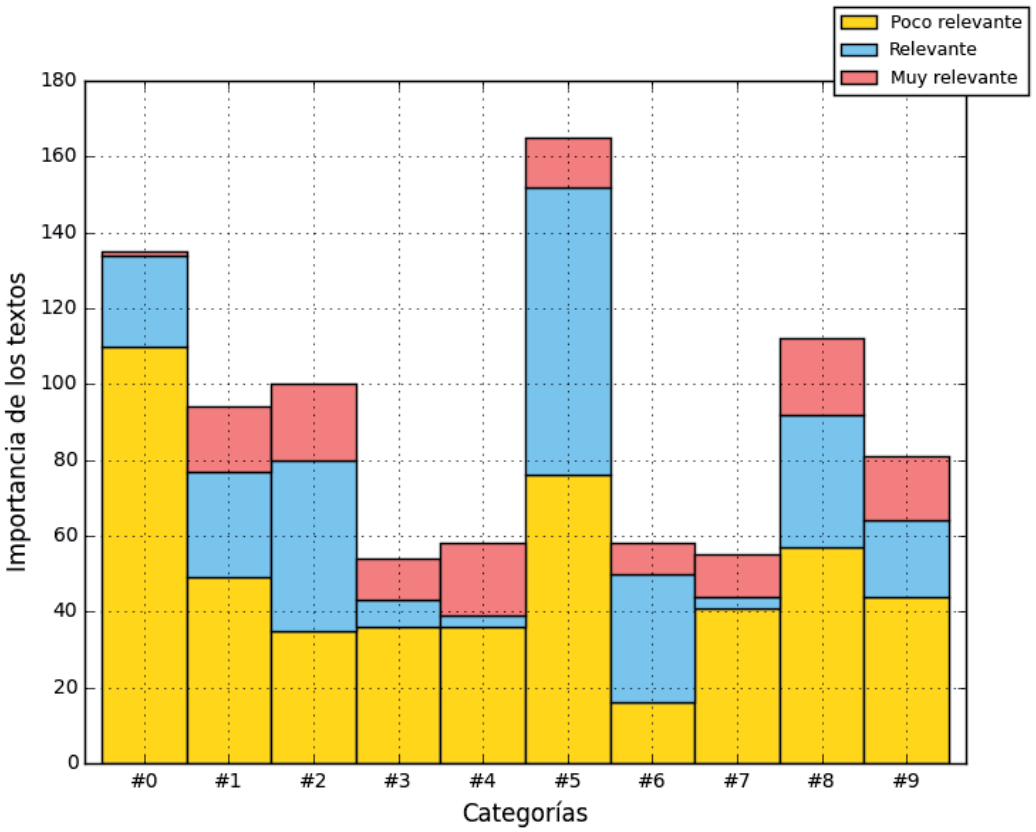


Figura 4.18: Detección de temas en los *tweets* de Luis García Montero

Tabla 4.16: Número de *tweets* de Luis García Montero por categoría.

	Poco relevantes	Relevantes	Muy relevantes	Total
Categoría 0	110	24	1	135
Categoría 1	49	28	17	94
Categoría 2	35	45	20	100
Categoría 3	36	7	11	54
Categoría 4	36	3	19	58
Categoría 5	76	76	13	165
Categoría 6	16	34	8	58
Categoría 7	41	3	11	55
Categoría 8	57	35	20	112
Categoría 9	44	20	17	81

Tabla 4.17: Stems más significativos de los *tweets* de Luis García Montero por categoría.

Categoría 0	madr: 4.2	comun: 3.8	candidat: 2.4	echar: 0.7	rest: 0.6
Categoría 1	polit: 3.3	deb: 1.3	gent: 1.2	hac: 1.0	quier: 0.6
Categoría 2	izquierd: 3.1	unid: 1.1	mayori: 1.0	derech: 0.9	valor: 0.7
Categoría 3	contrat: 1.9	consej: 1.6	civic: 1.5	social: 1.3	manan: 0.9
Categoría 4	miguel: 2.2	rios: 2.2	sabin: 1.2	joaquin: 1.1	conciert: 1.0
Categoría 5	elcorazondelaizquierd: 2.8	graci: 1.0	vam: 1.0	luch: 0.4	futur: 0.4
Categoría 6	acto: 2.4	candidatur: 2.4	presentacion: 1.7	empiez: 0.9	madridsemuev: 0.7
Categoría 7	vid: 2.4	buen: 1.9	institu: 0.7	echar: 0.7	esper: 0.6
Categoría 8	public: 2.4	servici: 1.5	trabaj: 1.2	decent: 0.9	debatetm: 0.9
Categoría 9	corrupcion: 2.3	part: 2.1	bien: 1.0	debatetm: 0.8	form: 0.7

4.4.5 Temas en los *tweets* de UPyD

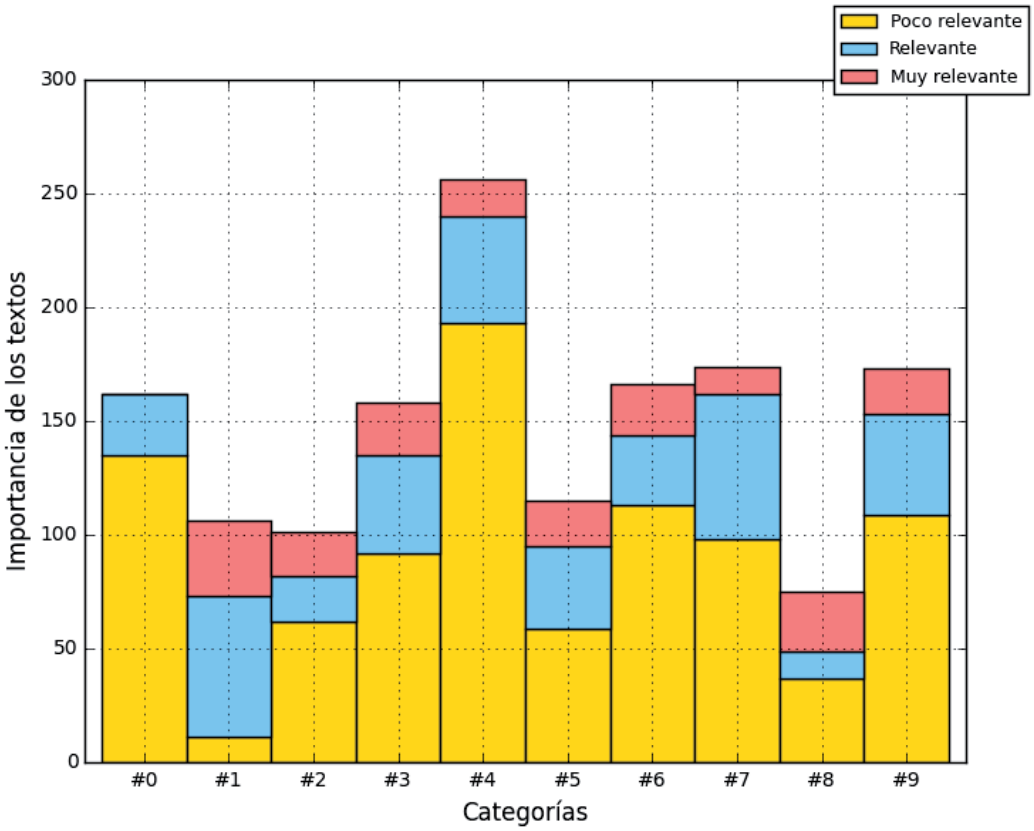


Figura 4.19: Detección de temas en los *tweets* de Ramón Marcos Allo

Tabla 4.18: Número de *tweets* de Ramón Marcos Allo por categoría.

	Poco relevantes	Relevantes	Muy relevantes	Total
Categoría 0	135	27	0	162
Categoría 1	11	62	33	106
Categoría 2	62	20	19	101
Categoría 3	92	43	23	158
Categoría 4	193	47	16	256
Categoría 5	59	36	20	115
Categoría 6	113	31	22	166
Categoría 7	98	64	12	174
Categoría 8	37	12	26	75
Categoría 9	109	44	20	173

Tabla 4.19: Stems más significativos de los *tweets* de Ramón Marcos Allo por categoría.

Categoría 0	candidat: 4.7	madr: 4.3	comun: 4.1	president: 2.6	ayto: 0.9
Categoría 1	graci: 3.0	much: 2.4	explic: 0.5	vosotr: 0.5	gran: 0.4
Categoría 2	sanid: 2.7	public: 2.5	universal: 1.6	mareablanc: 1.1	propon: 1.0
Categoría 3	trabaj: 2.5	hech: 2.0	segu: 1.1	entrev: 1.0	manan: 0.8
Categoría 4	libr: 3.2	polit: 1.6	emple: 1.2	social: 1.0	cre: 0.9
Categoría 5	levantaandaluci: 2.0	andaluci: 2.0	corrupcion: 1.6	vot: 1.6	canalsurm: 1.0
Categoría 6	espan: 1.9	tod: 1.5	fiscal: 1.3	pued: 1.0	gast: 1.0
Categoría 7	gent: 2.4	hac: 2.3	propuest: 0.8	cos: 0.7	servici: 0.7
Categoría 8	espanol: 2.6	queri: 2.3	part: 2.2	hiz: 1.1	sol: 0.5
Categoría 9	habl: 2.4	junt: 1.7	vecin: 1.3	visit: 0.9	parl: 0.7

4.4.6 Temas en los *tweets* de Podemos

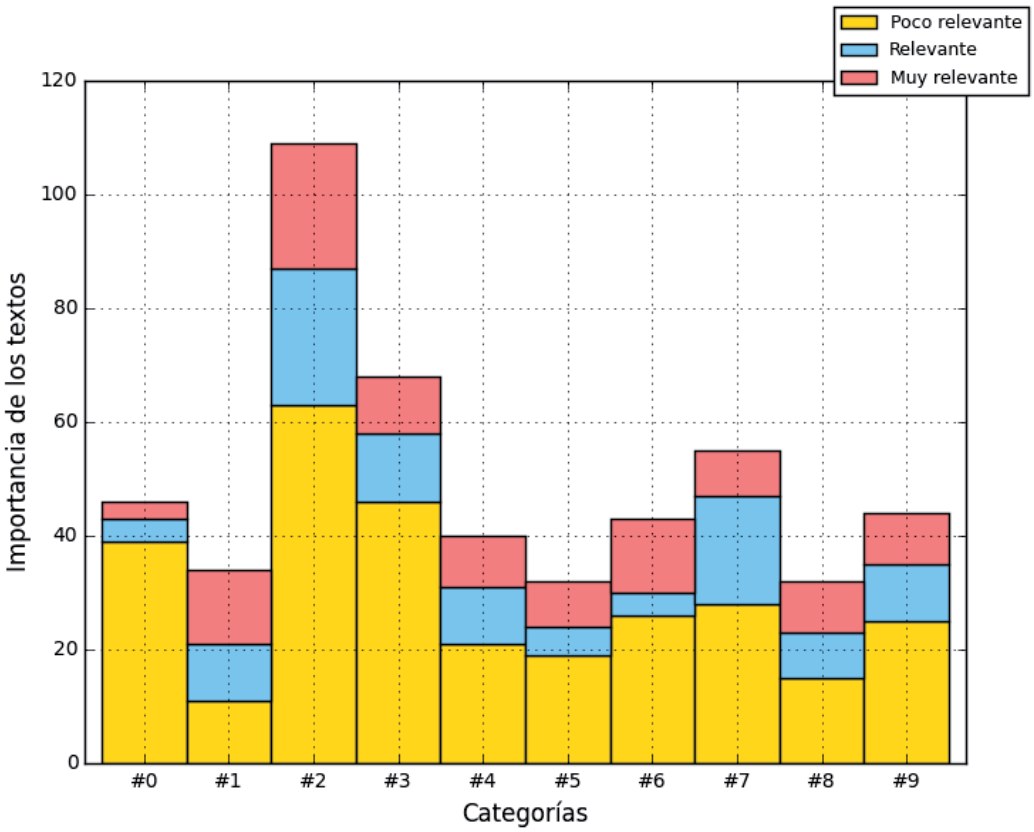


Figura 4.20: Detección de temas en los *tweets* de José Manuel López

Tabla 4.20: Número de *tweets* de José Manuel López por categoría.

	Poco relevantes	Relevantes	Muy relevantes	Total
Categoría 0	39	4	3	46
Categoría 1	11	10	13	34
Categoría 2	63	24	22	109
Categoría 3	46	12	10	68
Categoría 4	21	10	9	40
Categoría 5	19	5	8	32
Categoría 6	26	4	13	43
Categoría 7	28	19	8	55
Categoría 8	15	8	9	32
Categoría 9	25	10	9	44

Tabla 4.21: Stems más significativos de los *tweets* de José Manuel López por categoría.

Categoría 0	polit: 2.8	cambi: 2.6	proyect: 2.4	vam: 2.4	pais: 1.9
Categoría 1	vecin: 2.0	graci: 1.4	much: 1.2	tard: 0.8	henar: 0.5
Categoría 2	madr: 1.9	comun: 1.8	candidat: 1.3	entrev: 0.6	primari: 0.6
Categoría 3	campan: 2.2	hortalez: 1.0	empez: 1.0	gent: 1.0	barri: 1.0
Categoría 4	banc: 2.0	agu: 1.4	model: 0.9	microcreditospod: 0.9	signif: 0.8
Categoría 5	aguirr: 1.8	esper: 1.8	auster: 1.3	corrupcion: 0.9	sol: 0.8
Categoría 6	program: 2.0	gobern: 1.2	present: 1.1	mir: 0.8	gent: 0.8
Categoría 7	acto: 2.0	ahor: 1.3	cambiamadr: 1.2	punt: 1.0	comenz: 1.0
Categoría 8	cas: 1.5	camp: 1.5	vem: 1.3	encuentr: 1.0	buen: 0.9
Categoría 9	public: 1.4	propuest: 1.3	sanid: 1.2	debatetm: 1.1	gestion: 1.0

4.4.7 Temas en los *tweets* del PSOE

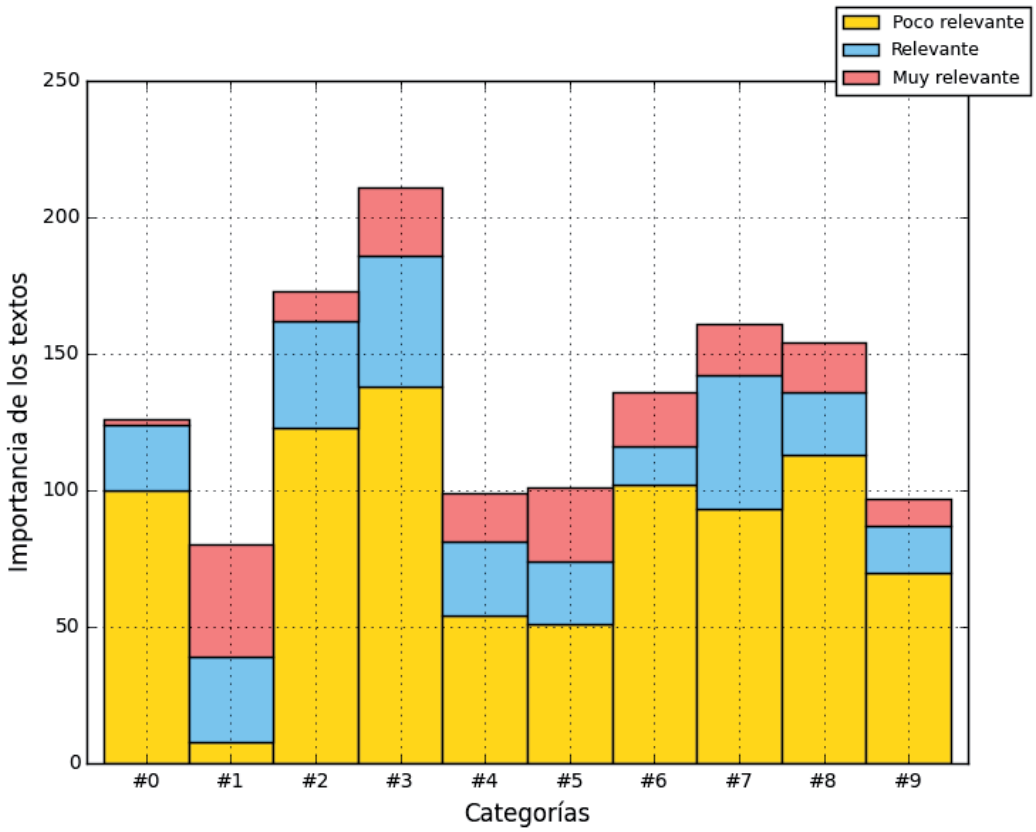


Figura 4.21: Detección de temas en los *tweets* del equipo de Ángel Gabilondo

Tabla 4.22: Número de *tweets* del equipo de Ángel Gabilondo por categoría.

	Poco relevantes	Relevantes	Muy relevantes	Total
Categoría 0	100	24	2	126
Categoría 1	8	31	41	80
Categoría 2	123	39	11	173
Categoría 3	138	48	25	211
Categoría 4	54	27	18	99
Categoría 5	51	23	27	101
Categoría 6	102	14	20	136
Categoría 7	93	49	19	161
Categoría 8	113	23	18	154
Categoría 9	70	17	10	97

Tabla 4.23: Stems más significativos de los *tweets* del equipo de Ángel Gabilondo por categoría.

Categoría 0	educacion: 4.5	cre: 2.4	sanid: 2.4	cultur: 1.7	emple: 1.3
Categoría 1	just: 2.9	solucion: 2.8	recuperacion: 0.5	necesit: 0.5	expert: 0.4
Categoría 2	solucionesjust: 2.9	derech: 1.7	votapso: 1.5	diferent: 1.4	votadivers: 0.6
Categoría 3	polit: 2.9	social: 1.9	public: 1.9	deb: 1.2	servici: 0.8
Categoría 4	acto: 2.8	pued: 1.9	segu: 1.5	streaming: 0.8	public: 0.8
Categoría 5	cambi: 2.9	quer: 1.5	gobiern: 1.4	gobern: 1.4	madr: 1.3
Categoría 6	pobrez: 2.9	viv: 1.5	person: 1.5	emple: 1.0	madr: 1.0
Categoría 7	hac: 2.7	mejor: 1.3	maner: 1.1	sol: 1.0	sin: 0.9
Categoría 8	candidat: 2.5	comun: 2.3	madr: 1.7	manan: 1.5	president: 1.1
Categoría 9	trabaj: 1.9	esfuerz: 1.9	estan: 1.5	medi: 1.5	clas: 1.5

4.4.8 Evaluación de los temas obtenidos

En la Tabla 4.24 se indica el número de textos no relevantes en cada una de las búsquedas automáticas de temas realizadas anteriormente. Los porcentajes se han redondeado a las décimas.

Tabla 4.24: Textos irrelevantes al detectar temas en los *tweets* de los candidatos

	Tweets irrelevantes	Número total de *tweets*	Porcentaje de irrelevantes
PP-1	288	941	30.61 %
PP-2	432	1608	26.87 %
Ciudadanos	61	246	24.80 %
IU	132	712	18.54 %
UPyD	281	1248	22.52 %
Podemos	47	325	14.46 %
PSOE	223	1083	20.59 %

También se tomaron medidas de los distintos errores cometidos al aplicar el algoritmo NMF. Estas medidas de error están explicadas en detalle en la sección 3.2.1.1, de donde se ha tomado la notación empleada: M es la matriz construida a partir de las frecuencias y E es la matriz de reconstrucción. Se han utilizado las normas de Frobenius de la matrices anteriores:

$$N_m = ||M||_F \qquad N_e = ||R||_F$$

Las medidas de error que se calcularon son el error neto E_n y el error promedio por término E_t.

Al realizar la norma de la matriz de reconstrucción los resultados son muy elevados. Estudiando los elementos de las matrices de reconstrucción en los distintos casos, se aprecia que algunos son muy grandes, aunque la mayoría de ellos presentan valores muy bajos. Al calcular el error de esta forma, esos pocos datos con gran error tienen un gran peso, por lo que no proporcionan una idea clara de cuánto error se ha cometido por término. Sin embargo, con el error promedio por término se cuantifica mejor el error cometido. Para valores similares de error promedio por término, un valor más elevado en la norma de la matriz de reconstrucción indica que los errores máximos son de mayor valor.

Los valores obtenidos (redondeados a las décimas) están recogidos en la Tabla 4.25.

Tabla 4.25: Errores cometidos al aplicar la técnica NMF

	N_m	N_e	E_n	E_t
PP-1	158.79	144.29	0.91	0.05
PP-2	246.53	233.51	0.95	0.04
Ciudadanos	52.32	41.82	0.80	0.10
IU	132.61	122.16	0.92	0.064
UPyD	204.03	192.94	0.95	0.05
Podemos	75.32	65.56	0.87	0.10
PSOE	190.42	178.78	0.94	0.05

Las normas N_m y N_e están directamente relacionadas con el tamaño de la muestra de los *tweets*, recogido en la Tabla 4.24. Los valores para el error neto E_n también dependen del número de textos. Se obtienen valores bastante elevados, ya que se trabaja con matrices de gran tamaño y, a pesar de que el número de errores no sea muy elevado, si el error es significativo produce una gran variación en la norma. Por ello, se ha tomado la medida E_t como la más representativa de la muestra, ya que proporciona información acerca del error cometido en cada campo de la matriz de frecuencias aproximada que se obtuvo tras la factorización.

Si se analiza la categorización de los textos según su importancia, un texto muy poco relevante podrá llegar a ser poco relevante, pero nunca relevante o muy relevante (podría darse, pero sólo en casos atípicos).

Si se analizan los *stems* más significativos, se le pueden asignar etiquetas a cada tema para así identificarlos mejor. Estas etiquetas se reúnen en la Tabla 4.26. Las categorías de esta tabla se han ordenado de mayor a menor importancia (entendiendo la importancia de una categoría como el número de *tweets* relevantes en ella). Los casos en en los que no es posible determinar la etiqueta de un tema se han etiquetado como *desconocido*. En algunos casos varios *clusters* determinan el mismo tema, ya que los *stems* que los identifican son muy similares (por ejemplo en PP-1, donde el tema sanidad aparece representado por dos *clusters* distintos).

Tabla 4.26: Temas por orden de importancia tratados en los *tweets* de los distintos candidatos

PP-1	PP-2	Ciudadanos	IU
Madrid(8)	Madrid(2)	Madrid(0)	Izquierda(5)
Programa electoral(7)	Campaña electoral(7)	Agradecimientos(1)	Madrid(0)
Agradecimientos(5)	Programa electoral(3)	Equipo político(9)	Servicios públicos(8)
Barrios(6)	Ind.(6)	Economía(3)	Izquierda(2)
Educación(3)	Sanidad(4)	Competencias públicas(6)	Política(1)
Ind.(4)	Impuestos y empleo(9)	Cambio(8)	Corrupción(9)
Transporte(9)	Familias(5)	España(4)	Concierto(4)
Sanidad(1)	Corrupción(8)	Impuestos(2)	Campaña electoral(6)
Sanidad(2)	Educación(0)	Corrupción(7)	Ind.(3)
Impuestos(0)	Transporte(1)	Ind.(5)	Ind.(7)

UPyD	Podemos	PSOE
Empleo(4)	Madrid(2)	Política social(3)
Ind.(7)	Barrios(3)	Soluciones(2)
Barrios(9)	Actos(7)	Ind.(7)
Impuestos(6)	Cambio(0)	Madrid(8)
Madrid(0)	Sanidad(9)	Pobreza(6)
Trabajo(3)	Programa(6)	Educación y sanidad(0)
Corrupción(5)	Economía(4)	Cambio(5)
Agradecimientos(1)	Barrios(1)	Actos de campaña(4)
Sanidad(2)	Corrupción(5)	Trabajo(9)
España(8)	Campaña electoral(8)	Soluciones(1)

4.5 Análisis de sentimientos en las respuestas de los usuarios a los candidatos

Se ha empleado la herramienta desarrollada en la sección 3.1.3 para detectar sentimiento en los *tweets* de los usuarios. Al igual que en el caso de la búsqueda de temas se han filtrado las palabras que tuviesen que ver con los partidos políticos y con los candidatos. También se han eliminado los enlaces y las menciones a otros usuarios.

El estudio se ha realizado en el conjunto de *tweets* de los usuarios que respondían a los candidatos. El número de *tweets* y su distribución se muestra en la Tabla 4.27. Se puede observar que el número no coincide con la tabla 4.7, ya que no hemos tenido en cuenta las contestaciones de los políticos entre ellos.

Tabla 4.27: Número y distribución de los *replies* de los usuarios

PP-1	**PP-2**	**Ciudadanos**	**IU**	**UPyD**	**Podemos**	**PSOE**	**Total**
1100	897	499	611	845	649	1022	5623

En primer lugar, se detectaron aquellos *tweets* que eran preguntas a los candidatos y los que eran muy enérgicos. Para ello se tuvieron en cuenta los signos de interrogación y exclamación antes de eliminarlos durante la fase de preproceso. Estos resultados se recogen en la Tabla 4.28.

Tabla 4.28: Número y distribución de las preguntas y mensajes enérgicos.

	PP-1	**PP-2**	**Ciudadanos**	**IU**	**UPyD**	**Podemos**	**PSOE**	**Total**
Preguntas	207	168	118	111	148	113	129	994
Enérgicos	150	116	75	58	101	100	96	696

En la Tabla 4.29 se ha calculado el ratio de preguntas y mensajes enérgicos entre políticos y usuarios, para así comparar su uso.

Tabla 4.29: Comparación del número de preguntas y mensajes enérgicos entre usuarios y políticos.

	Usuarios	**Políticos**	**Total**
Preguntas	994 (84.8 %)	178 (15.2)	1172
Enérgicos	696 (70.7 %)	288 (29.3 %)	984

Se observa una menor presencia de estos recursos entre los políticos debido a su posición. Los usuarios están mucho más predispuestos a hacer preguntas y a salirse de tono en las respuestas, ya que no son personajes públicos y pueden expresarse con mayor libertad.

Tras obtener los valores para la valencia, la excitación y la dominación, se cuenta con un total de 356 nulos. Estos valores nulos se dan en aquellos términos que no fueron encontrados en el diccionario de afectividad. Este resultado se obtiene a través de los indicadores para todo el corpus, pero como sólo se estudió la polaridad de las contestaciones de los usuarios, el número de mensajes nulos sólo es de 230. Este valor es bastante alto debido a que algunos mensajes son muy breves y casi no contienen información. Los políticos, al tener que dirigirse a un mayor número de gente, cuidan más sus mensajes, que suelen tener mayor información que los del resto de usuarios. Esto explica que no tengan tantos *tweets* nulos.

En la Tabla 4.30 aparecen los *tweets* de los que se tiene información. Estos valores se obtienen tras eliminar las respuestas de los usuarios a los políticos. Este es el subconjunto del corpus inicial en donde se realizó el análisis de sentimiento.

Tabla 4.30: Número y distribución de los *replies* de los usuarios

PP-1	**PP-2**	**Ciudadanos**	**IU**	**UPyD**	**Podemos**	**PSOE**	**Total**
1057	857	488	584	811	618	978	5393

La disminución tras eliminar los *tweets* es similar entre los candidatos respondidos, lo que indica que se tiene una muestra uniforme de mensajes, independientemente del candidato al que se dirigen.

Los valores de valencia, excitación y dominación para los distintos *tweets* se han representado los gráficos de barras de las Figuras 4.22, 4.23 y 4.24.

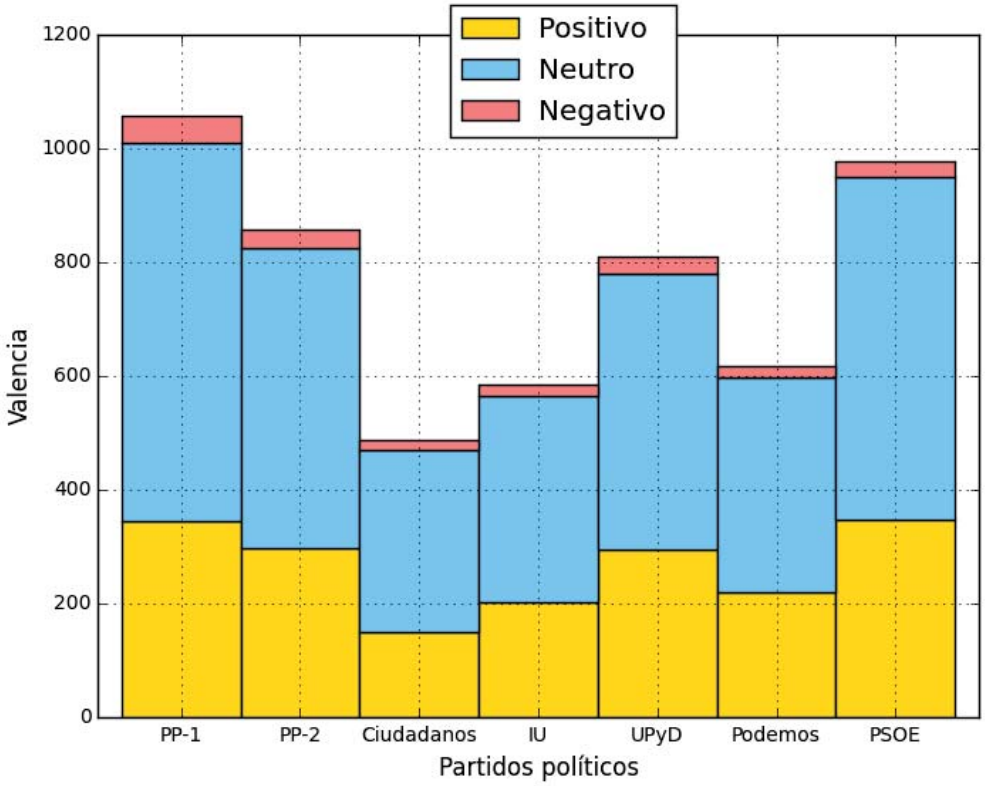


Figura 4.22: Valores de la valencia en las contestaciones de los usuarios

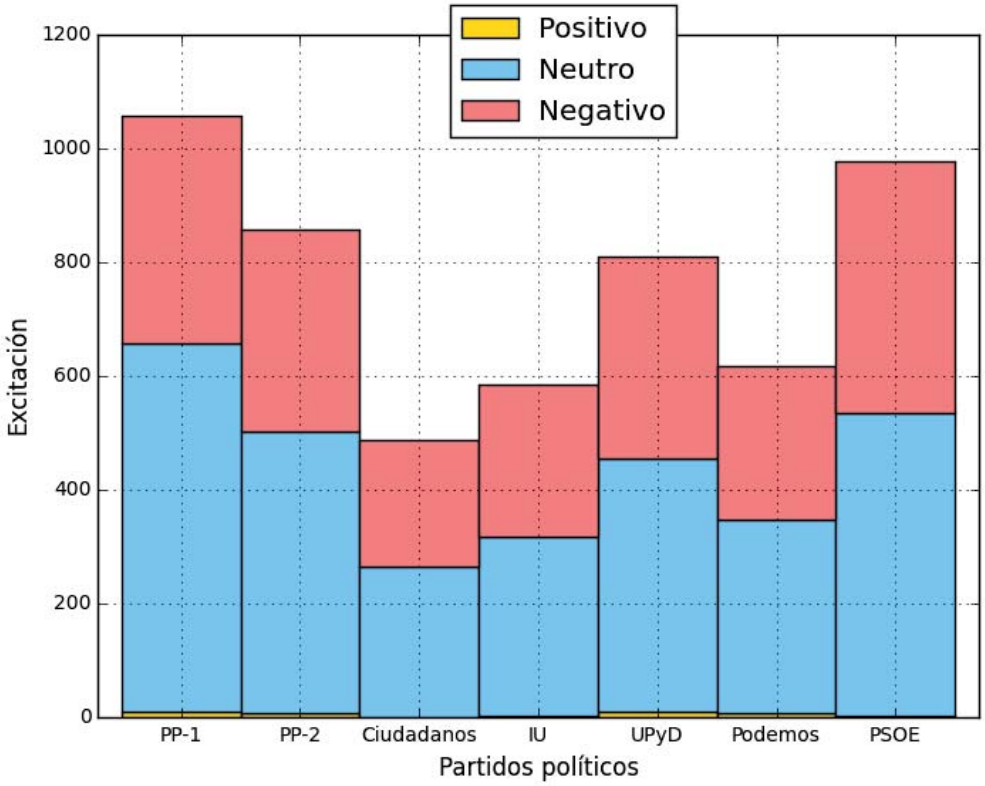


Figura 4.23: Valores de la excitación en las contestaciones de los usuarios

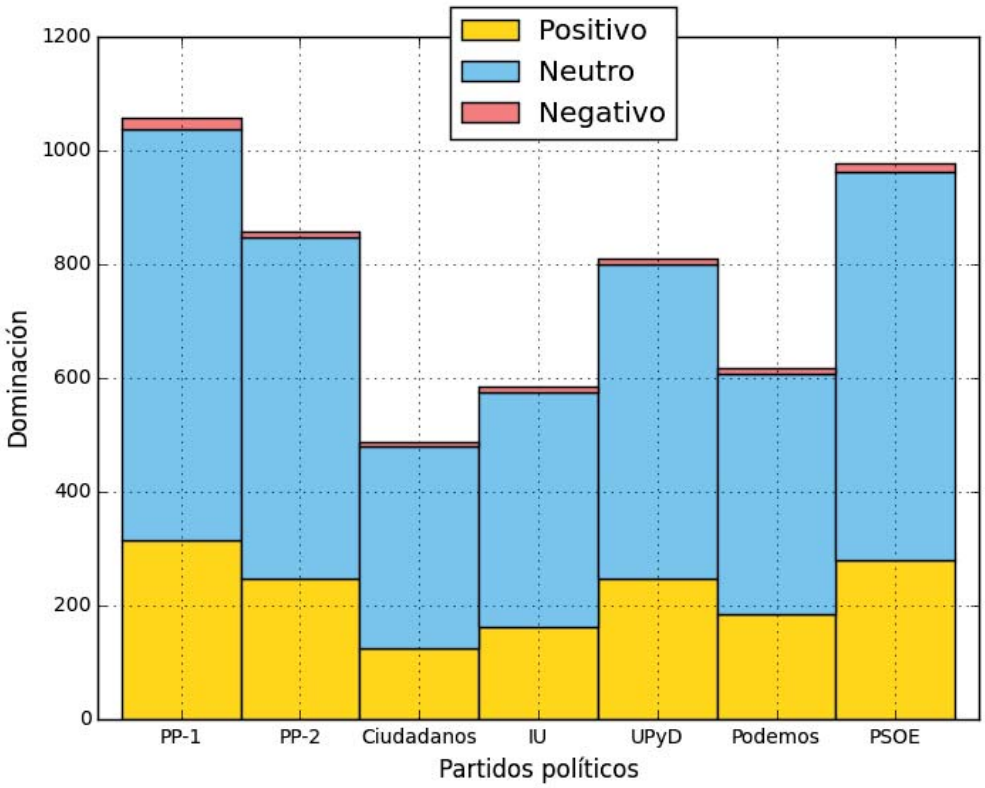


Figura 4.24: Valores de la dominación en las contestaciones de los usuarios

Los resultados obtenidos en las figuras anteriores dependen mucho del corpus usado y reflejan un mal comportamiento a la hora de detectar la polaridad de los *tweets* usando diccionarios léxicos con valores afectivos. Los valores de valencia se inclinan más hacia el lado positivo, mientras que en los de excitación estos son prácticamente inexistentes. En la dominación también predominan los positivos frente a los negativos. Sin embargo, en los tres casos el número de neutros es bastante elevado. Esto es comprensible, ya que la mayoría de las palabras suelen tener valores neutros para los tres indicadores. Se podrían haber refinado estos resultados si no se hubiesen tenido en cuenta algunas de las palabras neutras, pero los resultados tampoco hubiesen sido determinantes, ya que el uso de diccionarios afectivos no proporcionan buenos resultados.

A partir de los datos de valencia y excitación de las Figuras 4.22 y 4.23 se han representado los *tweets* en un plano de acuerdo al modelo circumplejo de Russel explicado en la sección 3.1.3. En función del partido político se ha usado un color distinto:

- PP-1: Azul claro.
- PP-2: Azul oscuro.

- Ciudadanos: Naranja.
- IU: Verde.
- Podemos: Violeta.
- PSOE: Rojo.

Los puntos se han representado por partido de forma superpuesta, por lo que en las zonas de mayor concentración parece que predominan los del PSOE (el último partido en ser representado), pero el número de los demás partidos es muy similar.

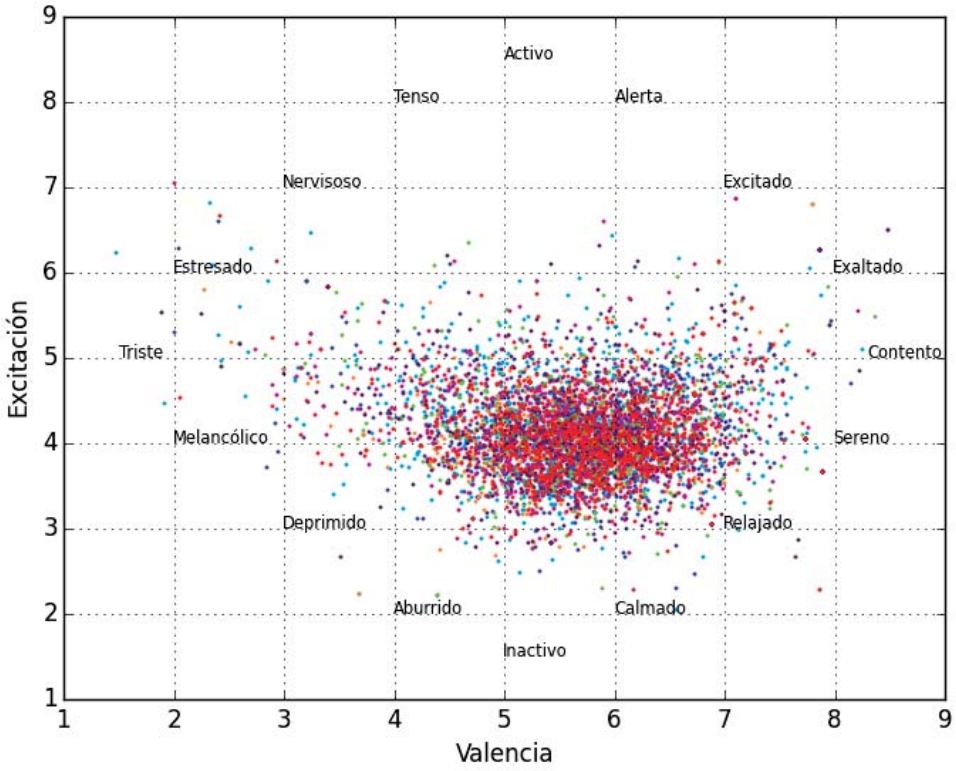


Figura 4.25: Visualización del sentimiento de los *tweets* usando el modelo circumplejo de Russel.

La mayoría de los mensajes representados en la Figura 4.25 están en la zona neutra, por lo que aportan poca información. Los puntos que se encuentran en los extremos se corresponden con mensajes en los que había palabras con valencias muy altas o muy bajas. En el caso de la excitación, hay una menor variación y menos casos atípicos.

A partir de los resultados anteriores de valencia, que es el indicador más relacionado con positivo y negativo, se realizó una clasificación usando Naïve Bayes. Se tomó un conjunto de entrenamiento de 1500 *tweets*. Este conjunto estaba etiquetado

en función de los resultados obtenidos anteriormente, de forma que 750 muestras se correspondía con los mensajes de máxima valencia (*tweets* positivos) y las otras 750 muestras contenían mensajes con un valor mínimo (*tweets* negativos). De esta forma, el conjunto de entrenamiento estaba balanceado.

Los resultados obtenidos para los diferentes partidos políticos se muestran en la Figura 4.26.

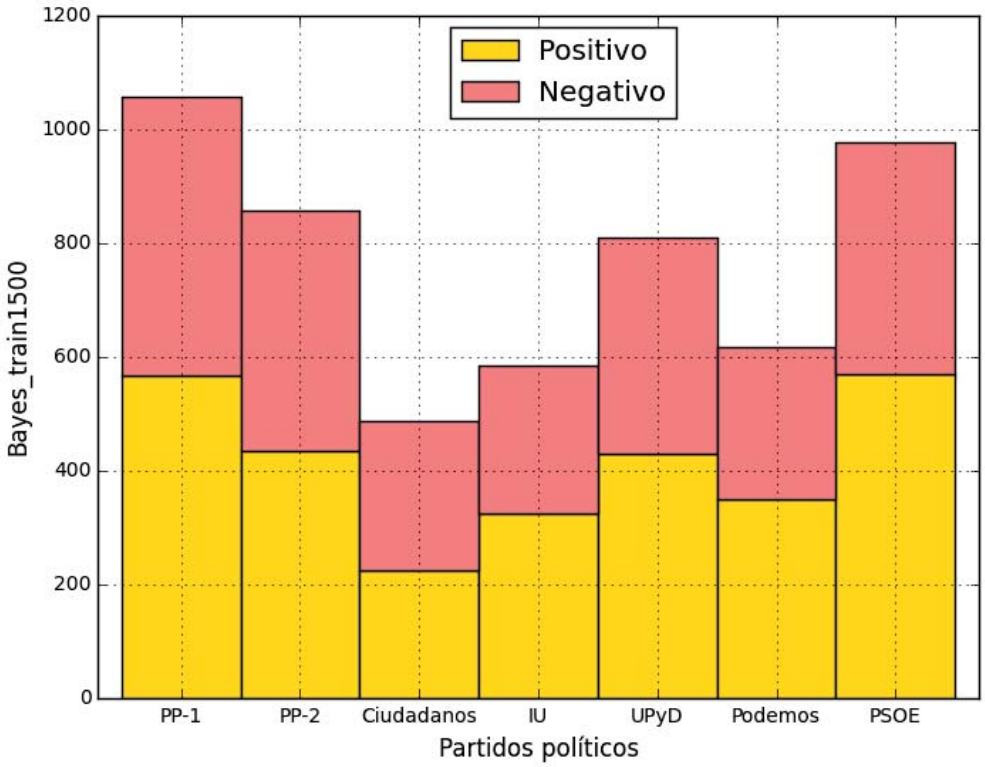


Figura 4.26: Clasificación de los *tweets* según el sentimiento en las respuestas a los distintos candidatos.

Los porcentajes de mensajes positivos por cada partido político extraídos de la Figura 4.26 se muestran en la Tabla 4.31.

Tabla 4.31: Porcentaje de mensajes positivos en las respuestas a los políticos

PP-1	**PP-2**	**Ciudadanos**	**IU**	**UPyD**	**Podemos**	**PSOE**
51.64 %	48.50 %	44.89 %	53.19 %	50.89 %	53.93 %	55.77 %

El menor valor máximo de valencia tomado en el entrenamiento ha sido 6.56 y el mayor valor mínimo 4.82. Estos valores se salen ligeramente de la escala de valencia

presentada en la sección 3.1.3, pero es preferible tomarlos, ya que en caso de usar una muestra de entrenamiento menor los resultados serían peores.

Las medidas de error no se pudieron realizar usando el etiquetado que se tenía, por lo que fue necesario etiquetar una pequeña muestra para calcular las medidas de error. En caso de contar con un mayor porcentaje de textos etiquetados, se mejorarían los resultados.

Los resultados siguen sin ser óptimos, ya que la muestra de entrenamiento es muy pequeña, pero se obtienen mejorías con respecto al caso anterior. Se ha decidido no tener en cuenta la etiqueta de neutro porque introducía mucho ruido. Gracias a la clasificación usando Naïve Bayes, muchos *tweets* que antes estaban considerados como neutros se han detectado como positivos o negativos. El único inconveniente es que entre las etiquetas de positivos y negativos se encuentran mensajes neutros que no aportan información. Se puede suponer que el porcentaje de neutros en las etiquetas de positivos y negativos es parecido, por lo que los valores de la Tabla 4.31 tienen cierta validez.

Se ha detectado que las contestaciones al candidato del PSOE son ligeramente más positivas que las del resto de candidatos. Las más negativas son las de Ciudadanos. En general, las cuentas reciben un número muy parecido de *tweets* positivos y negativos, por lo que parece que el número de usuarios afines y de detractores que contestan es similar. Además, normalmente, cuando un usuario critica a un candidato, otro afín a éste le defiende, por lo que acaban contrarrestándose.

Para obtener las medidas de error se ha utilizado un conjunto de entrenamiento etiquetado manualmente de 100 textos. Para que este conjunto de prueba estuviese balanceado se han tomado 50 textos con valencia alta y otros 50 con valencia baja. Esto no implica que tras el etiquetado manual haya 50 textos positivos y 50 negativos, pero ayuda a que el conjunto etiquetado esté balanceado. Las medidas de error obtenidas se muestran en la Tabla 4.32.

Tabla 4.32: Medidas de error del clasificador.

	Textos positivos	Textos negativos
Precision	0.88	0.93
Recall	0.92	0.83
F-measure	0.90	0.88

Se ha obtenido una exactitud de 0.78. Los valores obtenidos son bastante buenos, pero no deben interpretarse en sentido literal. Como se han elegido textos con valores de valencia altos y bajos, se explica que las medidas obtenidas sean superiores a lo esperado. Si se hubiera etiquetado un mayor número de textos y sus valores de valencia hubiesen sido neutros los resultados habrían resultado ser visiblemente inferiores. Aunque estas medidas tienen sus limitaciones, permiten corroborar el buen funcionamiento del clasificador.

En la Tabla 4.33 aparecen los *stems* que más tiene en cuenta el clasificador a la hora de determinar si un texto es positivo o negativo. Estos *stems* se encuentran ordenados de mayor a menor en función de su importancia.

Tabla 4.33: *Stems* más representativos del clasificador obtenido.

Positivos	Negativos
buen	*corrupt*
cre	*mentir*
gan	*necesit*
salud	*pag*
mejor	*baj*
graci	*los*

Uno de los mayores problemas a la hora de interpretar los resultados obtenidos está relacionado con el concepto que se tome de positivo y negativo. En muchas ocasiones los *tweets* que escriben los usuarios a los políticos son de críticas a otros partidos. Estas críticas se etiquetarían como negativas, pero son mensajes positivos para el político que las recibe, ya que critican las mismas cosas que ellos. Estas críticas cruzadas no se han tenido en cuenta y es uno de los principales motivos por los que no se puede aplicar la clasificación obtenida para determinar el número de apoyos a los distintos candidatos.

Capítulo 5

Conclusiones

En este trabajo se ha realizado una revisión del estado del arte y se han desarrollado varias técnicas y herramientas para analizar el comportamiento en Twitter de los candidatos a las elecciones autonómicas madrileñas del 2015. También se ha estudiado la respuesta que provocaban los tweets de los políticos en el resto de usuarios.

Se comenzó realizando un análisis pormenorizado del corpus, pues es una parte muy importante para realizar un correcto preproceso de los datos. También fue necesario realizar una limpieza de los datos antes de aplicar los algoritmos de Minería de Datos integrados en las herramientas desarrolladas. Los resultados obtenidos se consideran aceptables, pero podrían ser mejorados si se tienen en cuenta ciertas consideraciones que se comentan al final de este Capítulo.

Un hecho que ha limitado bastante este estudio fue la difusión dispar de mensajes de los distintos candidatos. Como algunos de los candidatos eran poco conocidos, su interacción con otros usuarios ha sido menor, lo que ha dificultado las comparaciones entre ellos.

Se aprecian grandes diferencias entre los candidatos que ya usaban Twitter con frecuencia y los que no. Este hecho ha influido bastante en la difusión de los *tweets* de los candidatos. Cuando los equipos de los políticos eran los que escribían los mensajes, se aprecia un número mayor de *tweets*. Sorprende las pocas respuestas a otros usuarios por parte de Podemos y el PSOE, lo que ha podido repercutir en su acogida y reducirla. En general, los *hashtags* no han sido aprovechados todo lo que deberían. El PSOE es el que más los ha usado, pero aún así no promovió debates en la ciudadanía, sino simplemente reforzaba mensajes propagandísticos. Se distingue muy poca variedad y originalidad entre los *hashtags*, lo que los hace poco memorables. Además, la poca repetición que se hace de ellos tampoco ayuda a integrase mejor en el ecosistema de Twitter. La comunicación entre los políticos y los ciudadanos es bastante vertical, con intercambios reducidos. En muy pocas ocasiones son los ciudadanos los que marcan los debates. Este enfoque puramente propagandístico no

es el más adecuado, por lo que seguramente ninguno de los candidatos ha logrado un gran calado entre el resto de usuarios.

Las interacciones son claramente mejorables y seguramente se tienda a invertir más en este tipo de campañas en el futuro. Se ha apreciado cierta participación por parte del resto de los usuarios, pero inferior a la esperada. Los políticos cuidan más el lenguaje que los que les siguen y les contestan. Tras realizar una búsqueda automática de los temas tratados, se ha observado que los temas tratados son muy similares entre los distintos partidos. La mayoría dedica bastantes *tweets* a agradecer a otros usuarios o a hablar de Madrid. La corrupción y los servicios públicos también son una gran preocupación. En los partidos de reciente creación (Podemos y Ciudadanos) el tema del cambio está también presente. Esta cuestión también se encuentra en el PSOE, que de esta forma intenta incorporarse a las tendencias que marcan estos nuevos partidos. UPyD centra su discurso en el empleo, Podemos en los los barrios, el PSOE en política social y Ciudadanos en la economía. En el PP se anuncian muchos actos de campaña electoral y se repasan los asuntos más importantes de su programa electoral.

Estos temas sirven para poder obtener una idea general acerca de en qué parte del espectro político se sitúa cada uno de los candidatos y qué estrategia sigue. Muchos de los temas son compartidos, ya que están marcados por la agenda, pero las pequeñas diferencias que se aprecian son suficientemente significativas.

En el análisis de sentimientos realizado, los resultados obtenidos no son muy exactos pero, en general, los niveles de aceptación son similares en todos los casos, por lo que no se pueden sacar conclusiones relevantes. Si los candidatos fuesen más conocidos (como sería el caso si el estudio se hubiese realizado en unas elecciones generales), el volumen de *tweets* recogido habría sido mayor y se podría haber contado con más contestaciones para analizar el sentimiento. Esto podría haber proporcionado mejores datos.

En general, los *tweets* de los políticos no causan gran impacto entre los usuarios. La mayoría o les apoya o les critica, pero los debates son poco frecuentes, por lo que en ese sentido se están desaprovechando las oportunidades que ofrece Twitter, ya sea por culpa de los candidatos, del resto de usuarios o de ambas partes.

El objetivo principal de este trabajo era estudiar el modo en que se comunicaban los políticos, detectar los temas que trataban y estudiar la acogida por parte del resto de usuarios. Además de esta información, resulta bastante interesante ver si es posible extraer de los datos obtenidos conclusiones relevantes que nos permitan averiguar el resultado de las elecciones. Obtener este tipo de conclusiones no es sencillo, debido a las dificultades intrínsecas a este tipo de análisis.

Al intentar utilizar los resultados para predecir el resultado de las elecciones surgen varios problemas. En primer lugar, no todo el mundo usa Twitter, por lo que sus usuarios no constituyen una muestra representativa e imparcial de todos los votantes. El número de usuarios jóvenes en Twitter es muy elevado, mientras que

el de personas mayores es escaso. Además, que una persona tenga una cuenta de Twitter no implica que escriba *tweets* sobre política ni tampoco que la información que escriba sea fiable. Por otro lado, el análisis de sentimientos realizado es bastante simplista, lo que da lugar a que sea complicado detectar automáticamente la preferencia política en un *tweet*. Por todas estas razones, este trabajo no permite predecir los resultados de las elecciones, ya que los resultados no serían fiables. A día de hoy, las predicciones políticas analizando Twitter son poco consistentes y aún queda mucho por mejorar en este aspecto. En [38], Daniel Gayo-Avello advierte acerca de la gran dificultad de estas predicciones e insta a no exponer resultados muy vistosos, pero insconsistentes sobre el tema. Recomienda centrarse más en los métodos para que sean reproducibles y, de esta forma, poder irse afinando cada vez más.

Este trabajo deja abiertas futuras líneas de investigación para profundizar en el tema tratado, pudiendo realizar este mismo estudio a una mayor escala, como podrían ser unas elecciones generales. Con el fin de que aquellos que se encuentren interesados en el tema puedan profundizar y mejorar el trabajo desarrollado, se proponen las siguientes mejoras:

- *Ampliación de las técnicas de preproceso:* Se podría usar un diccionario con palabras en castellano para así corregir las palabras con faltas de ortografía. El análisis de bigramas y trigramas, así como la incorporación de técnicas de procesamiento del lenguaje natural, también podrían suponer una mejora.

- *Refinamiento del número de clusters*: En este trabajo no se ha profundizado en el número óptimo de *clusters*. Una mejor determinación de este número podría mejorar considerablemente los resultados.

- *Ampliación o mejora del diccionario de afección de Warriner:* El diccionario empleado ha sido útil para obtener valores afectivos de las palabras, pero su traducción no está completamente validada. Lo ideal sería complementarlo con un corpus etiquetado de *tweets*, ya que en este trabajo se depende en exceso de los valores de dicho diccionario.

Bibliografía

[1] Salomé Berrocal. *Comunicación política en televisión y nuevos medios*. Ariel, 2003.

[2] Roberto Rodríguez-Andrés. Los efectos de la "americanización"de las campañas electorales del mundo. *Revista del Instituto Universitario de Investigación en Estudios Norteamericanos "Benjamin Franklin"de la Universidad de Alcalá*, 8:28–38, Marzo 2012.

[3] Miguel Túñez and José Sixto. Redes sociales, política y compromiso 2.0: La comunicación de los diputados españoles en facebook. *Revista Latina de comunicación social*, 66:1–25, 2011.

[4] Manuel Alejandro Martínez Martín. Redes sociales y política 2.0: Presencia en twitter de los candidatos. Master's thesis, Universidad de Sevilla, 2012.

[5] Andranik Tumasjan, Timm Oliver Sprenger, Philipp G Sandner, and Isabell M Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. *ICWSM*, 10:178–185, 2010.

[6] Seth Grimes. Unstructured data and the 80 percent rule. `http://breakthroughanalysis.com/2008/08/01/unstructured-data-and-the-80-percent-rule/`, 2008. Fecha de acceso: 2015-05-03.

[7] Ronen Feldman and Ido Dagan. Knowledge discovery in textual databases (kdt). In *KDD*, volume 95, pages 112–117, 1995.

[8] Andreas Hotho, Andreas Nürnberger, and Gerhard Paaß. A brief survey of text mining. In *Ldv Forum*, volume 20, pages 19–62, 2005.

[9] Pete Chapman, Julian Clinton, Randy Kerber, Thomas Khabaza, Thomas Reinartz, Colin Shearer, and Rudiger Wirth. Crisp-dm 1.0 step-by-step data mining guide. 2000.

[10] Melba M Crawford, Devis Tuia, and Hsiuhan Lexie Yang. Active learning: Any value for classification of remotely sensed data? *Proceedings of the IEEE*, 101(EPFL-ARTICLE-196283):593–608, 2013.

[11] Chuan-bi Lin. Projected gradient methods for nonnegative matrix factorization. *Neural computation*, 19(10):2756–2779, 2007.

[12] Christos Boutsidis and Efstratios Gallopoulos. Svd based initialization: A head start for nonnegative matrix factorization. *Pattern Recognition*, 41(4):1350–1362, 2008.

[13] Raúl Benítez, Gerard Escudero, Samir Kanaan, and David Masip Rodó. *Inteligencia artificial avanzada*. Editorial UOC, 2014.

[14] Jin Huang, Jingjing Lu, and Charles X Ling. Comparing naive bayes, decision trees, and svm with auc and accuracy. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 553–556. IEEE, 2003.

[15] Grigori Sidorov, Sabino Miranda-Jiménez, Francisco Viveros-Jiménez, Alexander Gelbukh, Noé Castro-Sánchez, Francisco Velásquez, Ismael Díaz-Rangel, Sergio Suárez-Guerra, Alejandro Treviño, and Juan Gordon. Empirical study of machine learning based approach for opinion mining in tweets. In *Advances in Artificial Intelligence*, pages 1–14. Springer, 2013.

[16] Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 1320–1326, 2010.

[17] SL Ting, WH Ip, and Albert HC Tsang. Is naive bayes a good classifier for document classification?

[18] Reza Entezari-Maleki, Arash Rezaei, and Behrouz Minaei-Bidgoli. Comparison of classification methods based on the type of attributes and sample size. *Journal of Convergence Information Technology*, 4(3):94–102, 2009.

[19] Sotiris B Kotsiantis, I Zaharakis, and P Pintelas. Supervised machine learning: A review of classification techniques, 2007.

[20] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis, volume 2 (1–2) of foundations and trends in information retrieval. *Now Publ*, 2008.

[21] Eugenio Martínez-Cámara, M Teresa Martín-Valdivia, L Alfonso Urena-López, and A Rturo Montejo-Ráez. Sentiment analysis in twitter. *Natural Language Engineering*, 20(01):1–28, 2014.

[22] Anjali Ganesh Jivani et al. A comparative study of stemming algorithms. *Int. J. Comp. Tech. Appl*, 2(6):1930–1938, 2011.

[23] Margaret M Bradley and Peter J Lang. Affective norms for english words (anew): Instruction manual and affective ratings. Technical report, Technical Report C-1, The Center for Research in Psychophysiology, University of Florida, 1999.

[24] Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. Norms of valence, arousal, and dominance for 13,915 english lemmas. *Behavior research methods*, 45(4):1191–1207, 2013.

[25] Daniel Gayo-Avello. Affective lexicons in spanish. `http://danigayo.info/PFCblog/index.php?entry=entry130117-183114`, 2015. Fecha de acceso: 2015-06-09.

[26] Jaime Redondo, Isabel Fraga, Isabel Padrón, and Montserrat Comesaña. The spanish adaptation of anew (affective norms for english words). *Behavior research methods*, 39(3):600–605, 2007.

[27] Albert Mehrabian and James A Russell. *An approach to environmental psychology.* the MIT Press, 1974.

[28] Wilhelm Wundt. *Outlines of psychology.* Springer, 1980.

[29] Auke Tellegen. Structures of mood and personality and their relevance to assessing anxiety, with an emphasis on self-report. 1985.

[30] James A Russell. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161, 1980.

[31] Michelle SM Yik, James A Russell, and Lisa Feldman Barrett. Structure of self-reported current affect: Integration and beyond. *Journal of personality and social psychology*, 77(3):600, 1999.

[32] Siddarth Ramaswamy. Visualization of the sentiment of the tweets. 2011.

[33] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.

[34] Bing Liu, Xiaoli Li, Wee Sun Lee, and Philip S Yu. Text classification by labeling words. In *AAAI*, volume 4, pages 425–430, 2004.

[35] Tao Li and Chris Ding. The relationships among various nonnegative matrix factorization methods for clustering. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 362–371. IEEE, 2006.

[36] Stephen A Vavasis. On the complexity of nonnegative matrix factorization. *SIAM Journal on Optimization*, 20(3):1364–1377, 2009.

[37] Maria Ogneva. How companies can use sentiment analysis to improve their business. *Retrieved August*, 30, 2010.

[38] Daniel Gayo-Avello. Don't turn social media into another'literary digest'poll. *Communications of the ACM*, 54(10):121–128, 2011.

Este documento esta firmado por

Firmante	CN=tfgm.fi.upm.es, OU=CCFI, O=Facultad de Informatica - UPM, C=ES
Fecha/Hora	Thu Jun 25 22:09:48 CEST 2015
Emisor del Certificado	EMAILADDRESS=camanager@fi.upm.es, CN=CA Facultad de Informatica, O=Facultad de Informatica - UPM, C=ES
Numero de Serie	630
Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)