ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
GRADO EN INGENIERÍA INFORMÁTICA: MENCIÓN EN COMPUTACIÓN

VALORACIÓN DE SENTIMIENTO EN LOS SOCIAL MEDIA MEDIANTE UN SISTEMA BASADO EN APRENDIZAJE AUTOMÁTICO

RATING SENTIMENT IN SOCIAL MEDIA USING A MACHINE LEARNING-BASED SYSTEM

Realizado por
Pablo Fuster Caro
Tutorizado por
José Ignacio Peláez Sánchez
Departamento
Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, FEBRERO 2015

Fecha defensa:
El Secretario del Tribunal

Resumen

La expansión de la web 2.0 en los últimos años ha provocado un crecimiento exponencial de la información disponible en internet. Este fenómeno ha originado interés por el análisis de sentimientos, una tarea del procesamiento del lenguaje natural que identifica opiniones relacionadas con diferentes temáticas. Existen diferentes redes sociales donde los públicos expresan sus opiniones en tiempo real. De estas redes, Twitter y Facebook son las más destacadas, de manera que diariamente millones de personas interactúan entre sí, expresando mediante textos sus opiniones acerca de diferentes temas. Esto ha hecho que dichas redes se hayan convertido en focos de información que permiten conocer en tiempo real las opiniones que los usuarios expresan sobre una gran variedad de temáticas. Conocer el sentimientos que los públicos tienen acerca de algo, es muy importante tanto para empresas privadas como públicas, ya que permite conocer qué sienten los públicos respecto de sus productos, actuaciones, etc. Pero esto no es tarea sencilla, ya que hay que extraer, analizar y clasificar dicha información, haciendo uso de métodos que permitan determinar la positividad o negatividad de dichas opiniones. Todo este proceso es lo que se denomina análisis de sentimiento, el cual es el encargado de determinar la positividad o negatividad de la opinión expresada. Para ello, existen diferentes métodos basados, por ejemplo, en técnicas estadísticas como el clasificador Naïve Bayes, métodos basados en diccionarios, o métodos basados máquinas de vectores soporte (SVM). En este trabajo se propone un modelo de análisis de sentimiento basado en aprendizaje automático por medio de la técnica del clasificador Naïve Bayes y tomando como conjunto de datos de entrada al sistema textos de Twitter y Facebook del sector bancario.

Palabras clave: minería de opinión, análisis de sentimiento, aprendizaje automático, aprendizaje supervisado, Naïve Bayes, Twitter, Facebook, medios sociales.

Abstract

The expansion of Web 2.0 in recent years has led to an exponential growth of information available online. This phenomenon has led to interest in sentiment analysis; a task of natural language processing that identifies views regarding any subject. There are different social media where different publics express their opinions in real time. In these media, Twitter and Facebook are the most prominent, so every day millions of people interact, using texts to express their opinions about different topics. This has made these social media become hotbeds of information that allow real-time information of what users opine on a variety of topics. Knowing that the publics have feelings about something, it is very important to both private and public companies, allowing know how they feel the public about their products, performances, etc. But this is not easy, because you need to extract, analyze and classify the information, using methods to determine the positivity or negativity of their opinions. This whole process is what is called sentiment analysis, which is responsible for determining the positivity or negativity of the views expressed. For this, there are different methods, for example, statistical techniques like Naïve Bayes classifier, dictionary-based methods, or support vector machines (SVM) methods. This paper presents a machine learning-based system using the Naïve Bayes classifier and taking as input data set Twitter and Facebook texts of the banking sector.

Keywords: opinion mining, sentiment analysis, machine learning, supervised learning, Naïve Bayes, Twitter, Facebook, social media.

Índice

1. Introducción

Desde hace algunos años venimos viendo la expansión de la web 2.0 y cómo esta va englobando a todos los ámbitos. La Web 2.0 o Web Social es una denominación de origen que engloba un gran número de espacios web basados en el principio de una comunidad de usuarios. Se puede entender como 2.0 todas aquellas utilidades y servicios de Internet que se sustentan en una base de datos, la cual puede ser modificada por los usuarios del servicio, ya sea en su contenido (añadiendo, cambiando o borrando información o asociando datos a la información existente), en la forma de presentarlos, o en contenido y forma simultáneamente. Algunos de estos servicios, utilidades y términos relacionados con la Web 2.0 pueden verse en Fig. 1.

Fig. 1: Términos relacionados con la Web 2.0

El concepto Web 2.0 surgió para referirse a nuevos sitios web que se diferenciaban de los sitios web más tradicionales englobados bajo la denominación Web 1.0. La característica diferencial es la participación colaborativa de los usuarios. Un ejemplo de sitio web 1.0 sería la Enciclopedia Británica donde los usuarios pueden consultar en línea los contenidos elaborados por un equipo de expertos. Como alternativa web 2.0 se encuentra la Wikipedia en la cual los usuarios que lo deseen pueden participar en la construcción de sus artículos.

La Web 2.0 abarca una amplia variedad de redes sociales, blogs, wikis y servicios multimedia interconectados cuyo propósito es el intercambio ágil de información entre los usuarios y la colaboración en la producción de contenidos. Todos estos sitios utilizan la inteligencia colectiva para proporcionar servicios interactivos en la red donde el usuario tiene control para publicar sus datos y compartirlos con los demás.

El porcentaje mayor del tráfico actual de internet está destinado a las redes sociales, esto se debe en gran parte a la necesidad innata del ser humano a relacionarse y comunicarse con sus semejantes, lo que limita en gran medida la vida moderna. Estas redes suplen de cierta forma el aislamiento social y la falta de relaciones interpersonales que trae consigo el desarrollo y la vida moderna. Estimula también la participación en estos sitios la facilidad que ofrecen de conocer personas y de establecer relaciones de amistad con gente que poseen gustos e intereses similares a los nuestros.

La utilización de las redes sociales es vital para que un buen plan de marketing digital funcione, captando nuevos clientes y aumentando las ventas. El uso de plataformas como Facebook y Twitter, dos de las redes sociales más usadas a nivel mundial en la actualidad, puede significar para tu empresa el ser conocida masivamente, o pasar desapercibida para todo el mundo.

Facebook es la red social más exitosa, conocida y popular de internet. Es una herramienta social para conectar personas, descubrir y crear nuevas amistades, subir fotos, compartir vínculos de páginas externas y videos. La popularidad de Facebook se debe a diversas causas:

- La facilidad de compartir contenido, ya sean enlaces, fotografías o videos.
- Una interfaz sencilla aún para los usuarios no experimentados en la navegación web.
- La facilidad de convertirse en miembro y crear una cuenta de manera gratuita.
- La facilidad que agrega el chat, que permite comunicarse en tiempo real sin necesidad de terceros.
- La integración de mensajes y correos electrónicos.

- Las recomendaciones de nuevos amigos, en muchas ocasiones acertadas.
- Las páginas de fans beneficiosas para negocios, empresas y marcas.
- La posibilidad de ganar dinero creando aplicaciones e integrándolas en el sistema.

Estos y muchos otros factores hacen que Facebook sea la red preferida, tanto para personas, negocios y empresas, hasta para el spam. Para registrarse en Facebook solo se necesita una dirección de email válida.

Twitter es una red social de microblogging, o sea una red para publicar, compartir, intercambiar, información, mediante breves comentarios en formato de texto, con un máximo de 140 caracteres, llamados tweets, que se muestran en la página principal del usuario. Es la plataforma de comunicación en tiempo real, más importante que existe en la actualidad. Los usuarios pueden suscribirse a los Tweets de otros, a esto se le llama "seguir" y a los suscriptores se les llaman "seguidores". Posee un especial atractivo para actualizar el estado rápidamente desde dispositivos móviles y para compartir noticias en tiempo real.

La principal característica de Twitter es su sencillez, también la facilidad y diversidad de formas existentes para conectarse a dicha red y poder comunicarse con otros. Puede ser desde una PC, un portátil, una tableta, un iPhone u otro Smartphone o hasta desde un teléfono cualquiera, aunque no tenga acceso a internet, solo que sea capaz de enviar mensajes SMS. Algunos de sus usuarios se han vuelto tan adictos a esta red, que no conciben internet sin Twitter. Muchas de las páginas de internet incluyen un botón que permite compartir con los seguidores en esta red cualquier página que se considere puede resultar interesante y útil, de esa forma adicionalmente se mantiene nuestro perfil constantemente actualizado.

El fenómeno de las redes sociales pasó de ser un medio de interacción para jóvenes a ser un canal de comunicación que las empresas deben considerar seriamente. El estudio de J.D. Power y asociados (2013) nos dice que el 67% de los consumidores utilizan los perfiles en redes sociales de una empresa para obtener

información sobre los servicios, y que el 43% de los jóvenes de entre 18 y 29 años prefiere interactuar con las marcas a través de las redes sociales que utilizar otro tipo de comunicación.

La disponibilidad de toda esta información permite que mediante las técnicas adecuadas de monitorización y análisis se pueda extraer el sentimiento de las opiniones sobre una gran variedad de ámbitos (sociales, económicos, laborales, etc.). A estas técnicas es lo que se conoce como minería de opinión.

A pesar de tratarse de una disciplina relativamente nueva, muchos han sido ya los trabajos realizados relacionados con la minería de opiniones y, más concretamente, con la clasificación de la polaridad. Se pueden distinguir dos metodologías principales a la hora de abordar el problema. Por una parte, la aproximación basada en aprendizaje automático utiliza una colección de datos representativos, denominado *corpus*, para entrenar los clasificadores y clasificar los datos. Por otra parte, la aproximación basada en orientación semántica, que no necesita un entrenamiento previo, sino que se tiene en cuenta la orientación positiva o negativa de las palabras.

En una clasificación basada en aprendizaje automático o supervisado, se requieren dos tipos de documentos: el conjunto de entrenamiento y el de prueba. El conjunto de entrenamiento se usa para enseñar al clasificador y el de prueba para ver la eficacia del clasificador automático. Hay disponibles un alto número de técnicas de aprendizaje automático para el análisis de sentimiento, entre las que destacan por sus éxitos Naïve Bayes, Máxima Entropía y Máquinas de Vectores Soporte (SVM), como muestra el estudio realizado por Pang, Lee y Vaithyanathan (2002).

El propósito de este trabajo es el desarrollo e implementación de un sistema para análisis de sentimiento basado en aprendizaje automático usando el clasificador Naïve Bayes, y las redes sociales Twitter y Facebook. Para ello el trabajo ha sido organizado como sigue: en la segunda sección se hace un breve estudio de los diferentes enfoques de la minería de opinión que existen en la actualidad; en la sección tercera, se presenta nuestra propuesta de modelo para el análisis de sentimiento en las redes social twitter y Facebook; en la cuarta sección se muestra la implementación realizada de los sistemas de lectura, base de datos e interfaz; y finalizaremos con las conclusiones.

2. Estado del arte

La minería de opinión, también conocido como análisis de sentimiento es el área de estudio que analiza las emociones, sentimientos, opiniones, actitudes y evaluaciones de las personas hacia productos, servicios, empresas, individuos y temas concretos. En la actualidad existe una gran diversidad de nombres para hacer alusión al mismo ámbito de estudio: valoración de sentimiento, análisis de sentimiento, minería de opinión, extracción de opinión, minería de sentimiento, análisis de subjetividad, análisis de emociones, etc.

Si bien las primeras apariciones de la terminología de *minería de opinión* y *análisis de sentimiento* aparecieron en los estudios realizados por Kushal, Lawrence y Pennock en 2003 y Nasukawa y Yi, también en el mismo año, las primeras investigaciones relativas a opiniones y sentimientos aparecieron a partir del año 1999, como es el caso del estudio elaborado por Wiebe. A raíz de esta investigación surgieron otras con la misma temática como las de Das y Chen en 2001, Morinaga y colaboradores en 2002, Pang, Lee y Vaithyanathan en 2002, Tong en 2001 o Turney en 2002.

Para poder realizar minería de opinión a un texto es necesario que el mismo tenga dos componentes elementales:

- El **tópico**: entidad, objeto o tema del que habla el texto.
- El **sentimiento**: polaridad referida al tópico.

Para solucionar la problemática del análisis de sentimiento, una solución fácil y común es asumir que, en base a un diccionario de palabras de sentimiento (*sentiment words*) positivas y negativas, si un texto tiene más palabras positivas que negativas, el texto será catalogado como positivo y análogamente, si tiene más negativas que positivas será catalogado como negativo. En caso de igualdad numérica el texto se toma como neutro.

Si bien esta solución no es completamente incorrecta, hay muchos más aspectos que deben ser considerados, debido a la complejidad que este campo de estudio, el procesamiento del lenguaje natural, requiere.

Calcular la polaridad de un texto en función de las palabras de sentimiento que este contenga resulta insuficiente, dado que la **semántica** influye a la hora de determinar la polaridad del mismo.

Se entiende por semántica, se refiere a los aspectos del significado, sentido o interpretación de signos lingüísticos como símbolos, palabras, expresiones o representaciones formales. Esta relación existente entre palabras da lugar a que no se pueda asumir que una frase que contiene palabras de sentimiento positivo sea también positiva; del mismo modo que si contuviese palabras de sentimiento negativo fuera negativa. De hecho, existen palabras que pueden invertir o potenciar el sentimiento de otra palabra.

El análisis de sentimiento puede enfocarse desde diferentes perspectivas dependiendo del tipo de análisis que se quiera aplicar:

- **Nivel de documento:** analiza el sentimiento general expresado en un texto. Tal y como expresan en sus estudios Pang, Lee y Vaithyanathan (2002) y Turney (2002), este tipo de análisis suele funcionar mejor bajo la precondición de que todo el documento sólo habla de un tópico, por lo que no es aplicable a documentos que comparan o contienen varias entidades.
- **Nivel de frase:** analiza el sentimiento expresado en cada frase. Dado un texto, lo divide en frases y analiza cada una de ellas de forma independiente. Como se dice en el estudio de Wiebe (1999), esta clasificación está altamente relacionada con clasificación de subjetividad, que distingue entre frases denominadas *frases objetivas* (que representan hechos) de *frases subjetivas* (representan opiniones). Para obtener el sentimiento final del texto completo se suele usar medias o mayorías de los sentimientos de las frases.
- **Nivel Entidad-Aspecto:** es un análisis de sentimiento con mayor granulado. Partiendo de un tópico, selecciona los aspectos que se desean analizar y evalúa el sentimiento para cada uno de ellos. En un principio Minqing y Liu (2004) lo denominaron *feature level*.

En la literatura, además, Jindal y Liu (2006b) hacen distinción entre dos tipos de opiniones, haciendo el problema del análisis aún más complejo:

- *Regular opinion*
- *Comparative opinion*

Una opinión estándar expresa el sentimiento de una sola entidad o una única característica de una entidad, mientras que una opinión

comparativa compara distintas entidades basadas en características comunes.

Teniendo en cuenta todo lo anterior, se exponen los problemas más comunes referentes a la minería de opinión:

- **Contexto:** ya que dependiendo del contexto una misma palabra puede presentar polaridades diferentes.
- **Ambigüedad del sentimiento:** del mismo modo que una frase que contenga *sentiment words* no implica que exprese una opinión positiva o negativa, una que tenga ausencia de ellas puede conllevar sentimiento.
- **Sarcasmo**, las palabras pueden perder todo su significado en presencia de ironía, burla, sarcasmo, etc.

Una solución necesaria pero no suficiente a para resolver estas problemáticas radica en el uso de un diccionario. En la literatura se documentan dos procedimientos para elaborar los diccionarios. Taboada (2011) y Tong (2001) explican que estos pueden ser creados de forma manual. Por el contrario, Kanayama (2006), Kayiy Kitsuregawa (2007), Turney (2002) y Turney y Littman (2003) defienden la posibilidad de crear el diccionario de forma automática; expandiéndolo a través de unas palabras que actúan como semillas.

3. Nuestra propuesta

El modelo que se propone en este trabajo es un modelo basado en aprendizaje automático supervisado, de manera que mediante una serie de ejemplos o conjunto de entrenamiento el sistema será capaz de aprender a determinar la **polaridad** (positiva, neutra o negativa) e **intensidad** del sentimiento de nuevos ejemplos. Estos ejemplos serán textos extraídos de las redes sociales Twitter y Facebook y realizaremos la clasificación en ellos a nivel de documento.

La polaridad del sentimiento es la subjetividad positiva, neutra o negativa que expresa el escritor en el texto. Estas tres polaridades serán nuestras tres clases básicas en las que vamos a realizar la clasificación.

Para trabajar la intensidad del sentimiento le daremos cierta granularidad a la polaridad distinguiendo en las clases positiva y negativa dos tipos de polaridad: normal y fuerte. De esta manera

tendremos que el sistema clasificará las informaciones en base a cinco clases, que ordenadas de menor a mayor polaridad son: muy negativo, negativo, neutro, positivo y muy positivo.

El sistema además trabajará con ciertas secciones de *Apertium*, que es un traductor gratuito y de código abierto. Las funcionalidades que usaremos de esta herramienta se detallarán en posteriores apartados.

3.1. Modelo de clasificación

El clasificador que usaremos está basado en métodos bayesianos. Los métodos bayesianos y la teoría de la probabilidad son dos de las técnicas que más se han utilizado en problemas de inteligencia artificial, de aprendizaje automático y de minería de datos. Dos son las razones principales por las que los métodos bayesianos son relevantes para el aprendizaje automático:

1. Son un método práctico para realizar inferencias a partir de los datos, induciendo modelos probabilísticos que después serán usados para razonar sobre nuevos valores observados. Además, permiten calcular de forma explícita la probabilidad asociada a cada una de las hipótesis posibles, lo que constituye una gran ventaja sobre otras técnicas.
2. Facilitan un marco de trabajo útil para la comprensión y análisis de diversas técnicas de aprendizaje y minería de datos que no trabajan explícitamente con probabilidades.

En este trabajo sólo trataremos con la primera faceta que se ha comentado, dado que construiremos y trabajaremos con un clasificador puramente probabilístico.

En teoría de la probabilidad, el teorema de Bayes es la regla básica para realizar inferencias. De esta forma, el teorema nos permite actualizar la creencia que tenemos en un suceso o conjunto de sucesos a la luz de nuevos datos. Es decir, nos permite pasar de la probabilidad a priori *P(suceso)* a la probabilidad a posteriori *P(suceso|observaciones)*. La probabilidad *a priori* puede verse como la probabilidad inicial que fijamos sin saber nada más, mientras que la probabilidad *a posteriori* es la que obtendríamos tras conocer cierta información y por tanto, puede verse como un refinamiento del conocimiento.

Teniendo en cuenta estos conceptos, el teorema de Bayes viene representado por la siguiente expresión:

$$P(h\,|O) = \frac{P(O \mid h) \cdot P(h)}{P(O)}$$

Donde lo que aparecen son la probabilidad a priori de las hipótesis *P(h)* y de las observaciones *P(O)* y las probabilidades condicionadas *P(h|O)* y *P(O|h)*.

Si nos centramos en el problema de la clasificación, con una variable clase *(C)* y un conjunto de variables predictoras o atributos *{A_1, …, A_n}* el teorema de Bayes queda como sigue:

$$P(C \mid A_1 \dots A_n) = \frac{P(\,A_1 \dots A_n|\,C) \cdot P(C)}{P(\,A_1 \dots A_n)}$$

De manera evidente, si *C* tiene *k* posibles valores *{c_1, …, c_k}*, lo que nos interesa es identificar la clase que tenga una probabilidad máxima a posteriori (*MAP*) dados los atributos y devolverla como resultado de la clasificación. Así, la clase o valor a devolver será:

$$c_{MAP} = \arg\max_{c \in \theta_c} p(c|A_1, \dots, A_n) = \arg\max_{c \in \theta_c} \frac{p(A_1, \dots, A_n\,|c)\,p(c)}{p(A_1, \dots, A_n)}$$
$$= \arg\max_{c \in \theta_c} p(A_1, \dots, A_n\,|c)\,p(c)$$

Donde Θ_c representa el conjunto de clases que puede tomar la variable *C*. En el último paso se ha eliminado la división debido a que el divisor es el mismo para todas las clases.

Naïve Bayes

Es una de las técnicas bayesianas más sencillas, aunque esto no merma su potencial, ya que es competitivo con otras técnicas más complejas como pueden ser Máxima Entropía o SVM, tal y como demuestran Pang, Lee y Vaithyanathan (2002).

El fundamento principal del clasificador Naïve Bayes es la suposición de que todos los atributos son independientes conocido el valor de la clase. Esta hipótesis de independencia hace que la expresión para obtener la hipótesis MAP quede como sigue:

$$c_{MAP} = \arg\max_{c \in \theta_c} p(A_1, \dots, A_n \,|c)\, p(c) = \arg\max_{c \in \theta_c} p(c) \prod_{i=1}^{n} p(A_i|c)$$

Es decir, la tabla de probabilidad *P(A₁, …, Aₙ|c)* ha sido factorizada como el producto de n tablas que solo involucran a dos variables. Por tanto, los parámetros que tendremos que estimar son *P(Aᵢ|c)* para cada atributo y la probabilidad a priori de cada clase *P(c).*

La estimación de la probabilidad condicional se basa, en caso de tener atributos discretos como es el caso del análisis de textos, en las frecuencias de aparición que obtenemos de los textos. Así, si llamamos *n(aᵢ, cₖ)* al número de veces que el atributo a_i toma el valor c_k y a *n(cₖ)* al número de veces que aparece cualquier atributo en la clase c_k, entonces una forma simple de estimar *P(aᵢ, cₖ)* es:

$$P(a_i|c_k) = \frac{n(a_i, c_k)}{n(c_k)}$$

Es decir, el número de casos favorables dividido por el número de casos totales. Sin embargo, dado que cualquier atributo que no se encuentre en una clase determinada tendrá probabilidad cero, tendremos que usar el estimador basado en la ley de la sucesión de Laplace:

$$P(a_i|c_k) = \frac{n(a_i, c_k) + 1}{n(c_k) + |A|}$$

Es decir, el número de casos favorables más uno, dividido por el número de casos totales más el número total de atributos.

Si enfocamos todo lo visto sobre el clasificador Naïve Bayes hasta este punto y lo enfocamos hacia el análisis de textos, tenemos que los atributos serán un conjunto de palabras distintas formadas a partir de los ejemplos de entrenamiento y que la probabilidad a priori de una clase será la frecuencia relativa del número de documentos de entrenamiento de una clase respecto del total de documentos de entrenamiento.

$$P(c_k) = \frac{|d_k|}{|D|}$$

Así pues, distinguiremos dos fases: entrenamiento y clasificación. En la fase de entrenamiento el sistema obtendrá el conjunto de palabras sobre los que basará la posterior clasificación, inferirá las probabilidades a priori de cada clase *P(cₖ)* y las probabilidades condicionadas de las palabras *P(aᵢ|cₖ)* a partir de los documentos de

entrenamiento. En la fase de clasificación, para cada nueva información que entre al sistema se obtendrán las palabras que coincidan con las entrenadas y en base a la probabilidad de estas, obtener la clase con mayor probabilidad a posteriori.

Para cualquier texto que entre al sistema se eliminarán todas las *stop words* que contenga, es decir, aquellas palabras que no aportan nada de cara al análisis del sentimiento tales como determinantes, preposiciones, conjunciones, etc.

3.2. Arquitectura del sistema

El sistema deberá ser capaz de comunicarse con la base de datos ya sea para obtener la información necesaria para su funcionamiento o para almacenar los resultados obtenidos.

A continuación, en la Fig. 2 se muestran los distintos módulos del sistema y las comunicaciones que existen entre ellos.

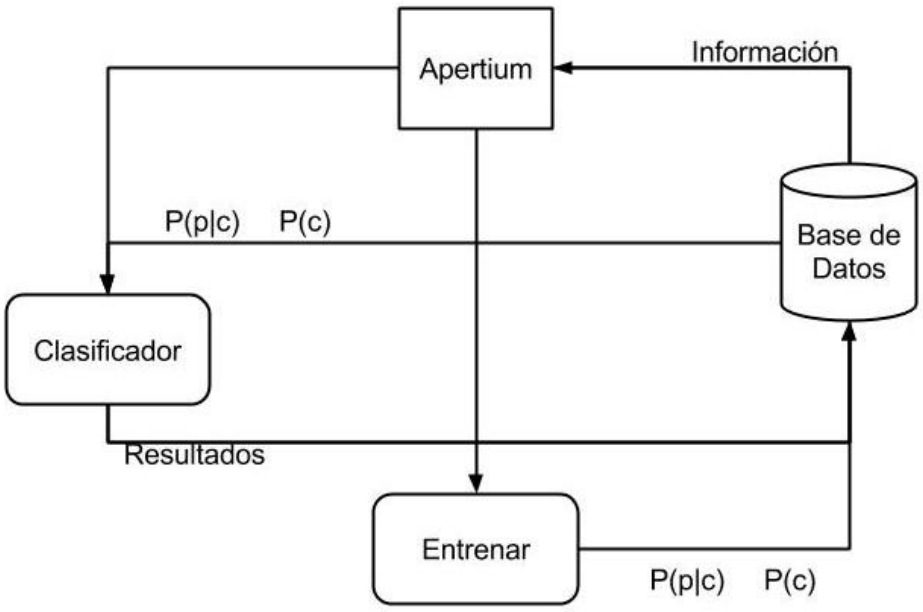


Fig. 2: Diagrama de flujo del sistema

- **Apertium**: este módulo realiza, teniendo como entrada textos de Twitter y/o Facebook una serie de procesos de normalización

del texto para posteriormente realizar el *etiquetado* y suprimir las *stop words*.

- **Entrenar**: realiza el entrenamiento del clasificador obteniendo el conjunto de palabras clave, las probabilidades a priori y las probabilidades condicionadas de las palabras mediante un conjunto de informaciones de entrenamiento después de realizar el módulo de *Apertium*.
- **Clasificador**: recibe como entrada informaciones ya depuradas con *Apertium* y las probabilidades a priori de las clases y las condicionadas de las palabras, obteniendo a partir de esto la clase con mayor probabilidad máxima a posteriori.

3.2.1. Apertium

Este módulo tiene a su vez una serie de módulos internos que están agrupados en tres categorías en función del momento en el que se utilizan. En la Fig. 3 se pueden ver estas categorías, así como los módulos que la forman y el orden en el que se van a ir ejecutando.

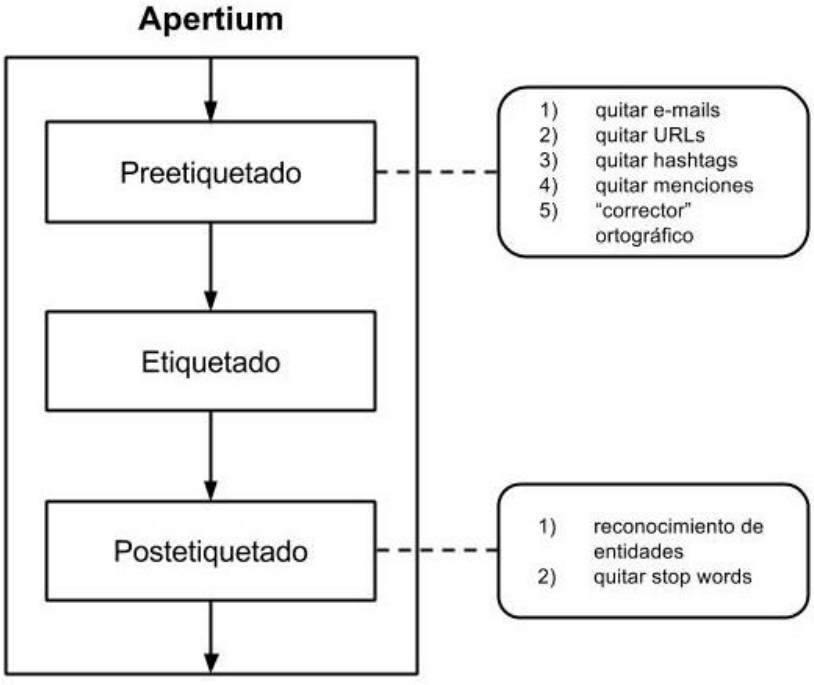


Fig. 3: Módulo de Apertium

3.2.1.1. Preetiquetado

Es frecuente que un texto contenga un conjunto de características que añadan una serie de complejidades a la hora de realizar el análisis de sentimiento. Estas características están determinadas por la presencia de URLs, *hashtags*, menciones, errores ortográficos… Es necesario, por tanto, normalizar el texto para un buen funcionamiento del clasificador.

Eliminación de e-mail

En ocasiones se dan informaciones que contienen una dirección de correo electrónico en el texto. Además de que carecen de valor al realizar el análisis de sentimiento, si no quitásemos estos bloques de palabras el clasificador las podría tomar como palabras clave, haciendo que las probabilidades condicionadas del resto de palabras disminuyan.

Eliminación de URL

En los textos se encuentran referencias a otras páginas (enlaces) que a la hora de realizar el análisis semántico carece de valor alguno. Para simplificar el proceso es conveniente prescindir de todas las URLs que aparezcan en cada tweet.

Eliminación de hashtag

La presencia de cualquier hashtag (carácter '#' seguido de un texto sin espacio) provoca que el texto que acompaña al carácter '#' no pueda ser reconocido por el etiquetador, lo cual generará un fallo en cadena en el resto de las informaciones. Si no se hiciera esta corrección, el etiquetador no reconocería el token a no ser que éste estuviera contemplado en su diccionario interno. La mejor solución para solventar este problema es detectar los *hashtag* y suprimir el carácter '#'.

Eliminación de menciones

Con las menciones existe la misma problemática que sucede con los hashtags, a excepción de que en vez de ser una '#' el carácter que inicia la mención, es una '@'. La solución adoptada para la solución del problema es la misma que la expuesta en el caso anterior.

"Corrector" ortográfico

En Twitter y Facebook es frecuente encontrar palabras mal escritas. En muchas ocasiones estos errores ortográficos son repeticiones indefinidas de una o más letras dentro de una misma palabra. Este módulo implementa un algoritmo que corrige la repetición de dos o más letras, dejando a lo sumo dos repeticiones en las situaciones más comunes que se pueden dar según las reglas definidas por la RAE.

Entrada	ssssssi, nooooooo, holaaaa
Salida	si, no, hola

3.2.1.2. Etiquetado

Una vez se ha normalizado el texto de entrada, es preciso etiquetar (*part-of-speech tagging*) la salida del módulo de *preetiquetado*. Para realizar esta labor se hará uso de un etiquetador, concretamente del etiquetador que posee *Apertium*.

Apertium es un sistema de traducción automática que tiene como una de sus principales características permitir el análisis de la entrada y la salida de cada uno de los módulos que lo componen. Esto permite usarlos de forma independiente y reorganizar su flujo de datos para la obtención de propósitos diferentes a los originarios.

Debido a que nuestro objetivo no es traducir ningún texto, no necesitaremos hacer uso de todos los módulos que se usan en *Apertium*. Sin embargo, sí haremos uso de determinados módulos, especialmente el etiquetador que mencionábamos antes. La Fig. 4 muestra los módulos de *Apertium* que se usarán en nuestro sistema así como su flujo de datos.

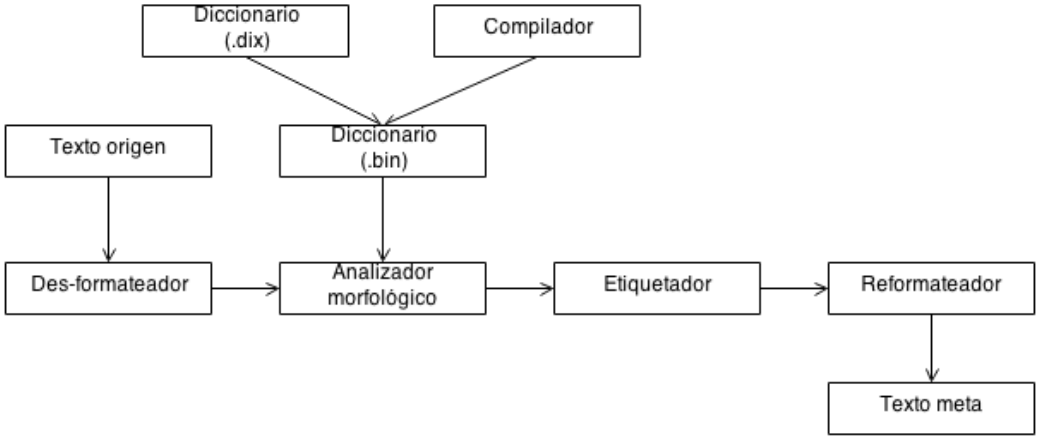


Fig. 4: Diagrama de flujo de los módulos utilizados de Apertium

MÓDULO	FUNCIÓN
Des-formateador	Separa el texto a traducir de la información de formato.
Analizador morfológico	Segmenta el texto en formas superficiales (FS) (las unidades léxicas tal como se presentan en los textos) y entrega para cada FS una o más formas léxicas (FL) consistentes en un lema (la forma base comúnmente usada para las entradas de los diccionarios clásicos), la categoría léxica (nombre, verbo, preposición, etc.) y la información de flexión morfológica (número, género, persona, tiempo, etc.).
Etiquetador	Elige mediante un modelo estadístico, uno de los análisis de una palabra ambigua de acuerdo con su contexto.
Re-formateador	Reintegra la información de formato original al texto traducido

Con la obtención del texto meta concluye la etapa de etiquetado.

3.2.1.3. Postetiquetado

La etapa postetiquetado consta de dos pasos:

1. Almacenar las etiquetas generadas por el etiquetador en una estructura de datos.

2. Eliminar aquellas palabras que o bien sean el sinónimo por el cual hicimos la búsqueda o bien sea una *stop word*.

Nuestro cometido al finalizar esta fase consiste en obtener las palabras clave del texto origen en su forma normalizada, ya que estas palabras claves son esenciales tanto para el entrenamiento como para la clasificación del texto.

Almacenar las etiquetas en una Estructura de Datos

Debido a que la fase de etiquetado es común a cualquier técnica que se aplique en el ámbito de análisis de sentimiento, resulta de gran interés almacenar toda la información generada por el etiquetador en una estructura de datos propia; de forma que la elaboración de esta estructura simplificará el desarrollo de futuras técnicas.

Teniendo en cuenta las especificaciones del módulo de etiquetado de *Apertium* es posible diseñar una estructura de datos que sea capaz de albergar toda la información generada por el etiquetador. El diseño de la estructura queda reflejado en la Fig. 5.

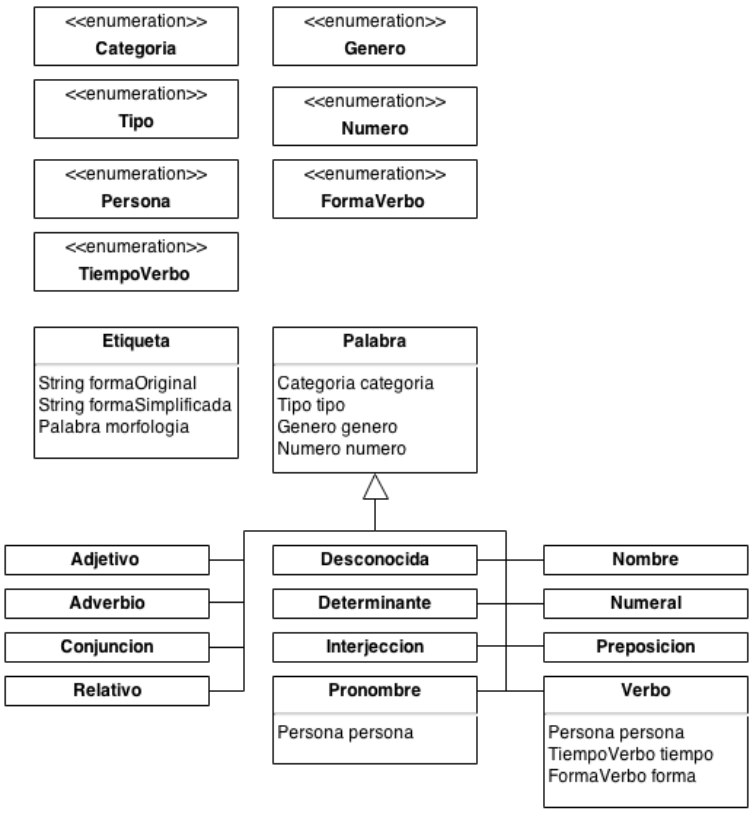


Fig. 5: Estructura de datos para almacenar las etiquetas

Una vez esté definida la estructura de datos en la que se almacenará toda la información generada por el *etiquetador*, se debe desarrollar un parseador que teniendo como entrada la salida de nuestra fase de *Etiquetado*, sea capaz de albergar la información en nuestra estructura definida.

Eliminar sinónimos y stop words

Existen ciertas palabras que se repiten con mucha frecuencia en los textos, tales como preposiciones, determinantes, conjunciones, etc.; son las llamadas *stop words*. Además, estas palabras no aportan, en primera instancia, sentimiento alguno y aparecerán de forma prácticamente simétrica en todas las clases. Si tenemos en cuenta que las probabilidades condicionadas de las palabras dependen, en parte, del tamaño del diccionario de palabras clave, se puede observar que al no tomarlas como palabras clave las probabilidades asociadas al resto de palabras aumentará. Así, la clasificación de un texto en una u otra clase dependerá en mayor medida de palabras tales como 'bueno', 'malo', 'mucho' o 'vivir' en vez de depender de que el texto tenga un 'de' o un 'a'.

Con los sinónimos pasa algo parecido. No tiene sentido que el sinónimo de un texto forme parte de las palabras clave, debido a que en unos textos habrá un sinónimo y en otros otro distinto del anterior y la probabilidad de que un texto sea de una clase u otra dependerá del sinónimo del que se esté hablando. Por esto la opción que se ha tomado ha sido la de detectar y suprimir los sinónimos en las etiquetas.

3.2.2. Entrenamiento

Para realizar el entrenamiento del clasificador Naïve Bayes es necesario tener un conjunto de documentos o textos previamente clasificados de forma manual, este conjunto se denomina conjunto de entrenamiento. En base a los textos del conjunto de entrenamiento este módulo calculará las *probabilidades a priori* de las clases y las *probabilidades condicionadas* de las palabras clave a cada una de las clases.

Como decíamos anteriormente, la probabilidad a priori la calcularemos como la frecuencia relativa del número de documentos de una clase respecto del total de documentos.

$$P(c_k) = \frac{|d_k|}{|D|}$$

Una vez los textos de entrenamiento han pasado el módulo de *Apertium*, todas las palabras distintas que han superado filtro de sinónimos y *stop words* serán tomadas como las palabras clave del sistema. Esto significa que serán ellas las encargadas de decidir, junto con la probabilidad a priori de las clases, qué clase determinamos para un texto no clasificado aún.

Para construir la probabilidad condicionada de una palabra a una clase necesitaremos el número de ocurrencias de esa palabra en textos de esa clase y el número de ocurrencias de esa palabra en todos los textos. Si *n(a, c)* es el número de ocurrencias de la palabra *a* en textos de la clase *c* y *n(c)* es el número de ocurrencias de todas las palabras clave en textos de clase *c*, el cálculo de la probabilidad condicionada queda como sigue:

$$P(a_i|c_k) = \frac{n(a_i, c_k) + 1}{n(c_k) + |A|}$$

Es decir, el número de ocurrencias de la palabra a_i en textos de la clase c_k más uno, dividido entre el número total de ocurrencias de las palabras en los documentos de la clase c_k más el tamaño del conjunto de palabras clave.

3.2.3. Clasificador

Una vez pasado el módulo de *Apertium*, la clasificación de nuevas informaciones irá en función de la probabilidad a priori de las clases y de aquellas etiquetas que coincidan con las palabras claves que obtuvimos en el entrenamiento. Por tanto, la fórmula de la hipótesis MAP (máxima a priori), explicada en anteriores epígrafes, quedaría así:

$$c_{MAP} = \arg\max_{c \in \theta_c} p(c|a_1, \dots, a_n) = \arg\max_{c \in \theta_c} p(c) \prod_{i=1}^{n} p(a_i|c)$$

Donde $\{a_1,\ldots, a_n\}$ son aquellas etiquetas del nuevo documento que existen en el conjunto de palabras clave, es decir, que tienen una probabilidad asociada, y $p(c)$ y $p(a_i|c)$ son las probabilidades calculadas en el entrenamiento.

Una vez obtenidas las probabilidades para cada clase, nos quedaremos con aquella que tenga una probabilidad mayor.

Es de tener en cuenta que si ninguna etiqueta coincide con el conjunto de palabras clave, la clasificación se regirá por la probabilidad a priori de las clases. Es por esto que no es recomendable que el número de documentos de entrenamiento sea el mismo para todas las clases.

4. Evaluación del sistema

Para evaluar el modelo propuesto se realizará un análisis en base a un conjunto de documentos previamente clasificados de forma manual. Este conjunto estará dividido en dos conjuntos: el conjunto de entrenamiento y el de prueba.

Se ha utilizado una proporción de 40:60 para los conjuntos de entrenamiento y prueba, es decir, que de 1155 documentos clasificados 462 se usarán para la fase de entrenamiento y 693 para obtener la tasa de acierto del sistema. Ambos conjuntos son totalmente disjuntos y por tanto, ningún documento que forme parte del entrenamiento tomará partido de la evaluación.

Una vez acabadas ambas fases (entrenamiento y prueba), los resultados obtenidos sostienen que el sistema tiene una tasa de acierto del 77.49%.

5. Implementación

5.1. Lectura de información

En esta sección se estudiarán las diferentes posibilidades para obtener información de Twitter y Facebook, así como se estipularán los términos de búsqueda que se van a usar.

5.1.1. Lectura de Twitter

Esta sección contendrá una breve introducción a los conceptos más comunes a la hora de hablar de Twitter, así como un análisis de las diferentes alternativas existentes para la captura de información de dicha red social. En el último epígrafe se mostrará la solución adoptada de entre las planteadas.

5.1.1.1. Conceptos básicos

- Tweet: Los tweets son las unidades de información que se emiten a través de Twitter cuya característica principal es la singularidad de que la longitud del texto no puede ser superior de 140 caracteres.
- Retweet: Mecanismo mediante el cual un usuario tiene la posibilidad de dar mayor difusión a un tweet, ya que mediante esta acción el tweet se estaría compartiendo con sus seguidores, que lo visualizarían en su timeline.
- Mención: Funcionalidad que permite citar a un usuario en un mensaje usando el símbolo '@' seguido del nombre del usuario.
- Respuesta: Un tweet escrito por un usuario de Twitter puede ser rebatido mediante otro tweet por otro miembro de la comunidad. De esta forma el nuevo mensaje aparecerá en el timeline de los usuarios comunes a ambos.
- Mensaje Directo: Es un mensaje privado entre dos usuarios de Twitter. Para poder enviar un mensaje privado a un usuario es necesario que ambos estén siguiéndose mutuamente.
- Hashtag: Un hashtag es una cadena de texto precedida por el símbolo '#'. Esta funcionalidad es útil cuando se quieren agrupar conversaciones referentes a una misma temática.
- Followers: Usuarios que siguen a un determinado usuario. Todos los mensajes escritos por un usuario, quedarán reflejados en el timeline de sus followers.
- Following: Es el concepto análogo a followers. Este concepto hace referencia a las personas a las que un miembro de Twitter

sigue. Los mensajes que emiten quedarán reflejados en su timeline.

- Timeline: Su traducción es línea temporal o cronología y en ella aparecen todos los tweets de las personas a las que un usuario está siguiendo además de los que el propio miembro escribe.

- Trending Topic: Son los temas más hablados del momento, los temas de actualidad.

5.1.1.2. Alternativas para la lectura de Twitter

Twitter proporciona múltiples APIs para facilitar el acceso a sus datos. De las existentes, para nuestro objetivo de recabar tweets nos encontramos con dos que cumplen con nuestro propósito:

- REST API
- API STREAMING

Cada una de ellas posee una serie de características y limitaciones, las cuales presentaremos a continuación.

5.1.1.2.1. REST API

Proporciona una gran cantidad de interfaces que engloban las distintas funcionalidades que ofrece Twitter. Estas interfaces son: Timeline, Tweets, Search, Streaming, Direct Messages, Friends & Followers, Users, Suggested Users, Favorites, Lists, Saved Searches, Places & Geo, Trends, Spam Reporting, OAuth, Help. En la Fig. 6 podemos ver que el funcionamiento de la REST API va por medio de peticiones, es decir, el servidor HTTP hace una petición y Twitter envía la respuesta.

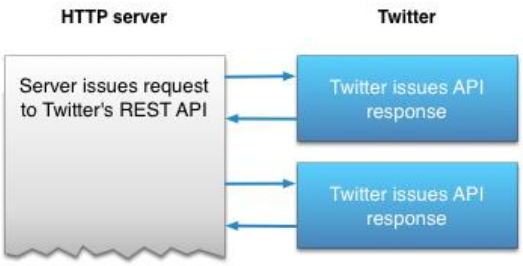


Fig. 6: Funcionamiento de la REST API

De todas las interfaces anteriormente citadas, la que más se adecua a nuestro objetivo es la interfaz de búsqueda, *'Search'*. A continuación se muestra un listado extraído de la documentación oficial de Twitter con los distintos parámetros que acepta esta interfaz.

Parameters	
Q Required	A UTF-8, URL-encoded search query of 500 characters maximum, including operators. Queries may additionally be limited by complexity.
Geocode Optional	Returns tweets by users located within a given radius of the given latitude/longitude. The location is preferentially taking from the Geotagging API, but will fall back to their Twitter profile. The parameter value is specified by "latitude, longitude, radius", where radius units must be specified as either "mi" (miles) or "km" (kilometers). Note that you cannot use the near operator via the API to geocode arbitrary locations; however you can use this geocode parameter to search near geocodes directly. A maximum of 1,000 distinct "sub-regions" will be considered when using the radius modifier.
Lang Optional	Restricts tweets to the given language, given by an ISO 639-1 code. Language detection is best-effort.
Locale Optional	Specify the language of the query you are sending (only ja is currently effective). This is intended for language-specific consumers and the default should work in the majority of cases.
result_type optional	Optional. Specifies what type of search results you would prefer to receive. The current default is "mixed." Valid values include: * mixed: Include both popular and real time results in the response. * recent: return only the most recent results in the response * popular: return only the most popular results in the

	response.
Count Optional	The number of tweets to return per page, up to a maximum of 100. Defaults to 15. This was formerly the "rpp" parameter in the old Search API.
Until optional	Returns tweets generated before the given date. Date should be formatted as YYYY-MM-DD. **Keep in mind that the search index may not go back as far as the date you specify here.**
since_id optional	Returns results with an ID greater than (that is, more recent than) the specified ID. There are limits to the number of Tweets which can be accessed through the API. If the limit of Tweets has occurred since the since_id, the since_id will be forced to the oldest ID available.
max_id optional	Returns results with an ID less than (that is, older than) or equal to the specified ID.
include_entities optional	The entities node will be disincluded when set to false.
Callback Optional	If supplied, the response will use the JSONP format with a callback of the given name. The usefulness of this parameter is somewhat diminished by the requirement of authentication for requests to this endpoint.

En la sección para desarrolladores de Twitter (dev.twitter.com) se encuentra una entrada que resume las buenas prácticas en el uso de dicha interfaz.

Las limitaciones más importantes asociadas al uso de esta API son las siguientes:

- La ventana temporal de consulta estará limitada (entre 6-9 días anteriores)
- Dependiendo del tipo de la clave usada, bien por usuario o bien por aplicación, existe un número máximo de peticiones en ventanas temporales de 15 minutos. La cantidad de peticiones es de 180 y 450 respectivamente.

5.1.1.2.2. API STREAMING

El conjunto de APIs Streaming que proporciona Twitter posibilita el acceso de baja latencia al Stream global de Tweets. Una implementación adecuada de un cliente de streaming, como la de la Fig.7, nos irá entregando tweets y eventos conforme vayan sucediendo sin la necesidad de estar haciendo *polling* a un terminal REST.

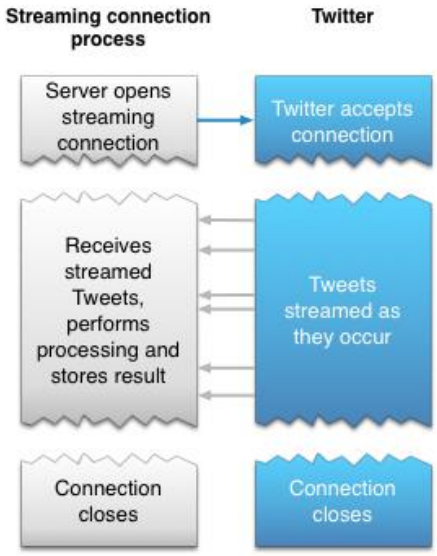


Fig. 7: Funcionamiento de la Stream API

Twitter suministra una serie de streams diferenciados, cada uno de ellos con un propósito distinto. A continuación, se listan los streams existentes:

- **Public Stream**: secuencia de datos públicos que fluyen a través de Twitter. Conveniente para realizar seguimiento de usuarios o temas específicos, así como *minería de datos*.
- **User Stream**: flujo de un solo usuario, que contiene más o menos todos los datos correspondientes a su timeline.
- **Site Stream**: versión multi-usuario de la User Stream. Destinado a los servidores que tiene que conectarse a Twitter en nombre de muchos usuarios concurrentemente.

Para la elaboración de este trabajo, el Stream que a priori parece más adecuado sería el público.

Las limitaciones más destacadas de su uso son las siguientes:

- El número máximo de tweets recibidos es equivalente a una pequeña fracción del volumen total de tweets generados en un instante determinado, entorno al 1%.
- No se puede estar abriendo y cerrando el stream con alta frecuencia, bajo riesgo de bloqueo a nivel de dirección IP.

5.1.1.2. Solución adoptada

Teniendo en cuenta el análisis anterior y nuestras necesidades para llevar a cabo el proyecto, la API escogida será Streaming API.

En la propia página de Twitter se puede encontrar la siguiente recomendación:

> *"Si tu aplicación requiere de repetidas peticiones a la Search API, deberías considerar el uso de Streaming API".*

Aclarada la solución que se adoptará, es importante detallar que se hará uso de la librería **Twitter4J.** Se trata de una librería no-oficial para Twitter API, que permite integrar de forma sencilla los servicios de Twitter en aplicaciones Java.

5.1.2. Lectura de Facebook

Esta sección contendrá una breve introducción a los conceptos más comunes a la hora de hablar de Facebook así como la solución tomada para la lectura de información de esta red social.

5.1.2.1. Conceptos básicos

- Perfil personal: Perfil individual para navegar por el sitio e interactuar con el resto de objetos de Facebook. La relación entre usuarios es de amistad recíproca, no es posible establecer una relación de amistad si uno no acepta la petición de amistad. Toda la privacidad de quién ve qué en Facebook está basada en estas amistades.
- Página: También llamadas en ocasiones perfiles públicos, son usadas para promocionar marcas, empresas, artistas, etc. Los usuarios con perfil pueden hacerse fan de una página, de

manera que los nuevos contenidos que publique el administrador de la misma le aparezcan en su muro como si de otra amistad se tratase. Sólo el administrador de la página puede escribir una nueva publicación.

- Grupo: Son análogos a los clubes del mundo offline, habiéndolos públicos, bajo previa aceptación y privados. A diferencia de las páginas el administrador es un individuo con perfil personal y todos los afiliados al club pueden escribir nuevas publicaciones.
- Muro: Lugar donde aparecen las novedades en posts de amigos, de páginas de las que se es fan, de grupos a los que se está afiliado e inclusive las propias aportaciones.
- Post: Comentario que se hace en el muro de un amigo, página, grupo o evento. Estos posts también pueden ser en respuesta a un primer post, es decir, sólo existe un nivel de anidamiento.
- Like: Contador de un post que indica el número de usuarios a los que les ha gustado o están de acuerdo con tu comentario.
- Compartir: Crea un nuevo post a partir de otro copiando el contenido del original.

5.1.2.2. Alternativas para la lectura de Facebook

Facebook tiene a disposición de desarrolladores (en developers.facebook.com) dos APIS para interactuar con sus datos:

- **Ad API**: proporciona funcionalidades para la gestión de anuncios.
- **Graph API**: principal forma de lectura y escritura mediante aplicaciones en Facebook.

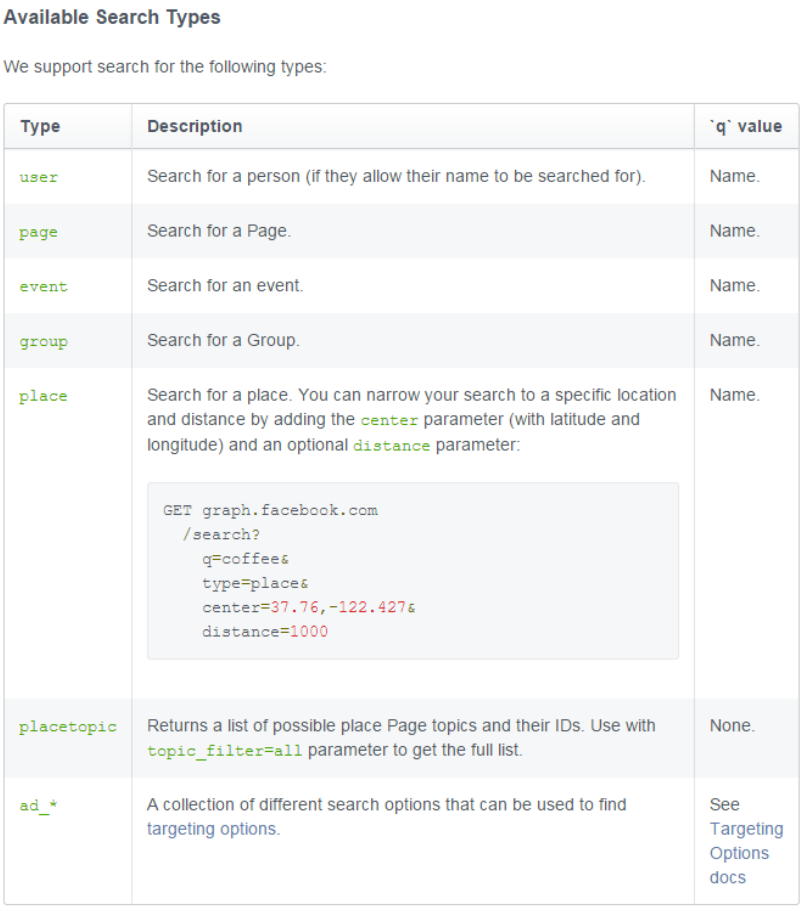
De estas dos la única válida para nuestro propósito de leer y guardar información es la *Graph API*, por lo que sólo hablaremos de ella a continuación.

5.1.2.3. Graph API

Proporciona una gran cantidad de interfaces que engloban las distintas funcionalidades que ofrece Facebook. Estas interfaces son: AccountMethods, ActivityMethods, AlbumMethods, LikeMethods BatchRequestsMethods, CheckinMethods, CommentMethods, DomainMethods, EventMethods, FamilyMethods, FavoriteMethods, FQLMethods, FriendMethods, GameMethods, GroupMethods,

InsightMethods, LinkMethods, LocationMethods, MessageMethods, NoteMethods, NotificationMethods, PageMethods, PermissionMethods, PhotoMethods, PokeMethods, PostMethods, QuestionMethods, RawAPIMethods, SearchMethods, UserMethods, SubscribeMethods, TestUserMethods y VideoMethods.

De estas interfaces la que a primera impresión nos convendría sería la SearchMethods, sin embargo, como se puede ver en la Fig. 8, captura de la propia página de Facebook, los distintos tipos de objetos que se pueden buscar no incluyen posts.



Available Search Types

We support search for the following types:

Type	Description	`q` value
`user`	Search for a person (if they allow their name to be searched for).	Name.
`page`	Search for a Page.	Name.
`event`	Search for an event.	Name.
`group`	Search for a Group.	Name.
`place`	Search for a place. You can narrow your search to a specific location and distance by adding the `center` parameter (with latitude and longitude) and an optional `distance` parameter: `GET graph.facebook.com /search? q=coffee& type=place& center=37.76,-122.427& distance=1000`	Name.
`placetopic`	Returns a list of possible place Page topics and their IDs. Use with `topic_filter=all` parameter to get the full list.	None.
`ad_*`	A collection of different search options that can be used to find targeting options.	See Targeting Options docs

Fig. 8: Objetivos posibles de la SearchMethods

Debido a que no se pueden hacer búsquedas de posts mediante la SearchMethods, nos surge la necesidad de buscar otras alternativas:

- Escoger una lista de usuarios y acceder a sus posts.
- Escoger una lista de páginas y acceder a sus posts.

Nos decantaremos por esta última, ya que si accedemos a las páginas públicas de las entidades de las que vamos a buscar información es bastante más probable que encontremos textos referentes a éstas que si buscamos en los posts de usuarios. Además debido a las restricciones que tiene Facebook no es posible leer información del muro de un usuario que no esté en relación de amistad contigo, mientras que sí lo es del muro de las páginas públicas.

5.1.3. Términos de búsqueda

Los términos por los se buscará información en Twitter y Facebook se encuentran recogidos en la siguiente tabla.

ENTIDAD	**SINONIMO**
CaixaBank	CaixaBank, La Caixa
Banco Santander	Banco Santander
Bankia	Bankia
Banco Sabadell	Banco Sabadell, Banco de Sabadell
Banco Popular	Banco Popular

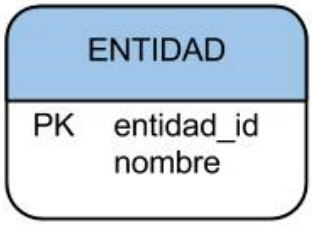
5.2. Diseño del modelo de la base de datos

En este epígrafe se hará un breve resumen para presentar el que será nuestro modelo Entidad/Relación a lo largo del trabajo. Al final del trabajo se adjunta un anexo con la especificación detallada del modelo.

5.2.1. Lectura

Las informaciones, bien tweets o posts de Facebook, quedarán almacenadas en la tabla INFORMACION. Para cada instancia de la tabla se almacenará un identificador, su texto, el usuario que generó la información, la fecha y hora de creación y dos flags, uno para discernir la procedencia (Twitter o Facebook) y otro para saber si ya ha sido valorado o no.

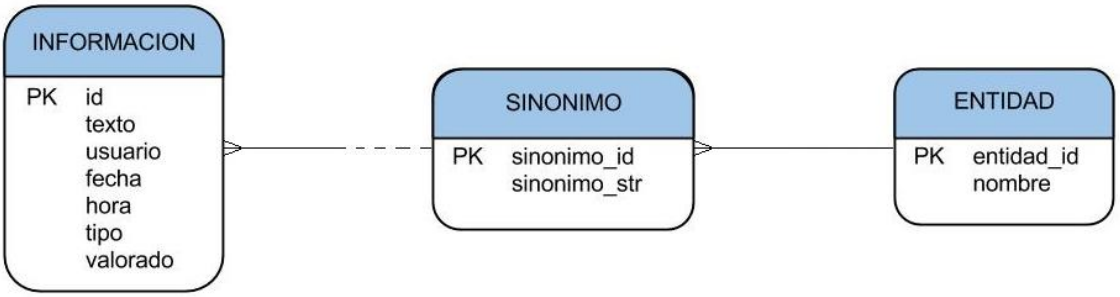
Cada una de las personas, empresas, organizaciones o temas de los que vamos a buscar información se encuentran recogidas en la tabla ENTIDAD. De estas entidades se guardará un identificador y el nombre de la entidad.



Las distintas formas en las que podemos encontrar información de una Entidad son las instancias que forman la tabla SINONIMO. Para cada uno de estos sinónimos tendremos un identificador y el sinónimo de la entidad.

Con estas tres tablas ya es posible realizar la lectura de Información para unas Entidades en base a unos Sinónimos (o términos de búsqueda) de las mismas.

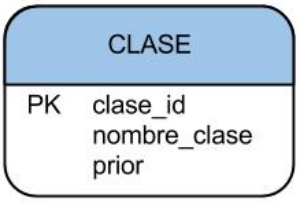
Las relaciones entre entidades de esta parte del modelo son las que se muestran en la siguiente imagen:



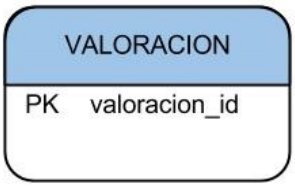
5.2.2. Clasificador Naïve Bayes

Para completar el modelo E/R se añadirán tablas necesarias para realizar el análisis de sentimiento mediante la técnica de Naïve Bayes.

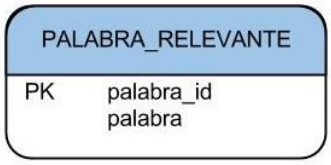
Las distintas clases en las que se van a clasificar las informaciones compondrán la tabla CLASE, donde cada una tendrá un identificador, el nombre de la clase y la probabilidad a priori.



Las clasificaciones que se harán de las informaciones se almacenarán en la tabla VALORACION, las cuales sólo tienen como atributo un identificador.

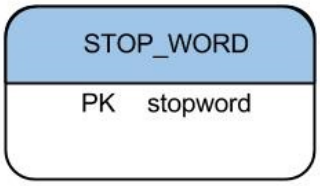


Las palabras que confeccionan el diccionario base de la clasificación se guardarán en la tabla PALABRA_RELEVANTE, donde cada instancia tendrá dos atributos, un identificador y la palabra correspondiente.

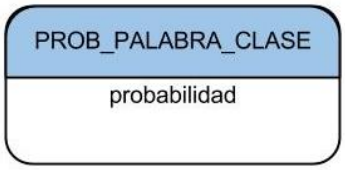


Será necesario guardar también las palabras vacías o *stop words* para filtrarlas durante la clasificación. Se guardarán en la tabla

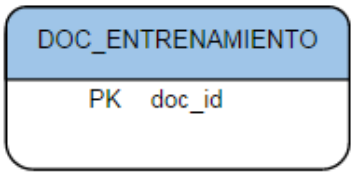
STOP_WORD, donde tendrá la stop word que también será el identificador.



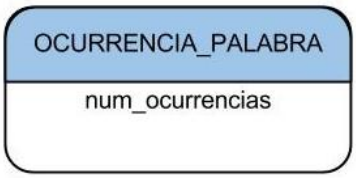
Para poder clasificar usando Naïve Bayes nos hace falta guardar las probabilidades asociadas a cada palabra_relevante en la tabla PROB_PALABRA_CLASE, donde el atributo será dicha probabilidad.



Las informaciones que forman el conjunto de entrenamiento se guardarán en DOC_ENTRENAMIENTO, donde tendrán un identificador.



Para realizar el entrenamiento previo necesitaremos tener el número de ocurrencias de las palabras en los documentos de entrenamiento. Esto quedará recogido en la tabla OCURRENCIA_PALABRA donde tendrán como atributo el número de apariciones de esa palabra en un documento determinado.



Finalmente el modelo completo para la realización de la lectura y de la clasificación queda como en la Fig. 9.

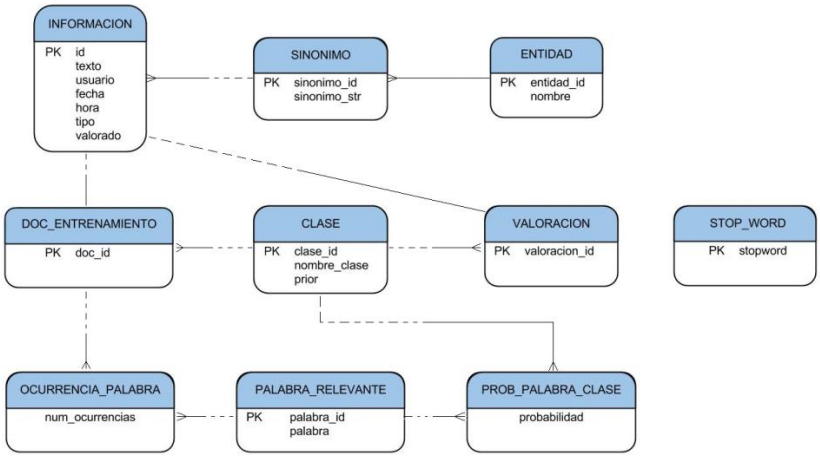


Fig. 9: Modelo completo de la Base de Datos

5.3. Elaboración de la base de datos

Definido nuestro modelo E/R (documentación más detallada en el anexo técnico), se puede implementar todo lo especificado. Para ello haremos uso de las siguientes herramientas:

- Oracle SQL Developer Data Modeler
- Microsoft SQL Server 2008 R2

La primera herramienta nos ayudará a modelar el modelo E/R con todas las especificaciones existentes y tras realizar unos pasos de ingeniería, obtener un script DDL para así poder crear la base de datos en el sistema escogido.

La segunda herramienta, Microsoft SQL Server 2008 R2, es el SGBD escogido para realizar este proyecto. Una vez realizado el modelo E/R, realizaremos la ingeniería con objetivo de obtener el modelo Relacional para finalmente generar el Script DDL para SQL Server 2008 mediante la herramienta de exportar (este script se encuentra en el disco físico).

Una vez tenemos el script, generamos una nueva base de datos para el proyecto y lo ejecutamos. El resultado es el que podemos ver en la Fig. 10.

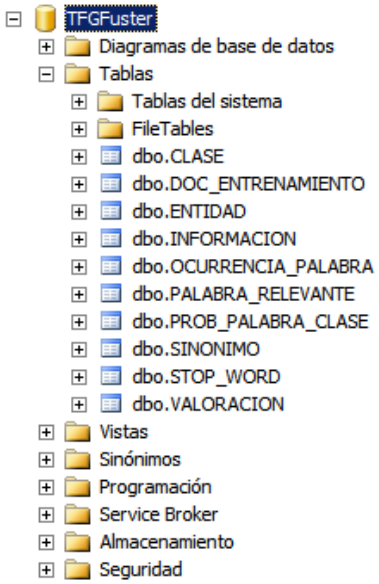


Fig. 10: Tablas generadas por el script en la Base de Datos

5.4. Comunicación Java-BBDD

Para poder manejar instancias de la base de datos desde Java se hará uso de Hibernate, que es un framework de este lenguaje que existe con ese fin. Hibernate es una herramienta de mapeo objeto-relacional (ORM) que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones en los beans de las entidades que permiten establecer estas relaciones. Además es software libre distribuido bajo licencia GNU LGPL. En la Fig. 11 podemos ver dos imágenes: una de las anotaciones en el modelo de las entidades en Java y una del fichero de mapeo XML.

```java
@Id
@GeneratedValue
@Column(name = "id")
public int getId() {
    return id;
}

public void setId(int id) {
    this.id = id;
}


@Column(name="texto")
public String getTexto() {
    return texto;
}

public void setTexto(String texto) {
    this.texto = texto;
```

```xml
<!-- SQL dialect -->
<property name="dialect">org.hibernate.dialect.SQLServer2008Dialect</pro

<!-- Enable Hibernate's automatic session context management -->
<property name="current_session_context_class">thread</property>

<!-- Disable the second-level cache  -->
<property name="cache.provider_class">org.hibernate.cache.internal.NoCac

<!-- Echo all executed SQL to stdout -->
<!-- <property name="show_sql">true</property> -->
<property name="format_sql">true</property>

<!-- Drop and re-create the database schema on startup -->
<!-- <property name="hbm2ddl.auto">update</property> -->

<mapping class="modelo.Informacion"/>
<mapping class="modelo.Entidad"/>
<mapping class="modelo.Sinonimo"/>
<mapping class="modelo.Valoracion"/>
<mapping class="modelo.StopWord"/>
<mapping class="modelo.Documento_Entrenamiento"/>
<mapping class="modelo.Clase"/>
<mapping class="modelo.Ocurrencias_Palabra"/>
<mapping class="modelo.Palabra_Relevante"/>
<mapping class="modelo.Prob_Palabra_Clase"/>
```

Fig. 11: Captura de las anotaciones de Hibernate en Java y del fichero XML

Para la generación de los ficheros de configuración y de mapeo de las clases de dominio se instalará el *plugin* de Hibernate para el entorno de desarrollo de Eclipse, que facilita la generación de todos estos ficheros por medio de una automatización asistida.

5.5 Interfaz

Mediante el diseño de la interfaz se propone mostrar de una forma sencilla y clara los resultados obtenidos durante el proyecto. Para ello se crearán dos pantallas principales. La primera contendrá una breve introducción al proyecto, mientras que la segunda permitirá visualizar los datos obtenidos de cada una de las entidades que teníamos como objetivo de análisis.

En la primera pantalla (Fig. 12) se pretende hacer una pequeña introducción a la minería de opinión, así como al sistema que se ha llevado a cabo en el proyecto.

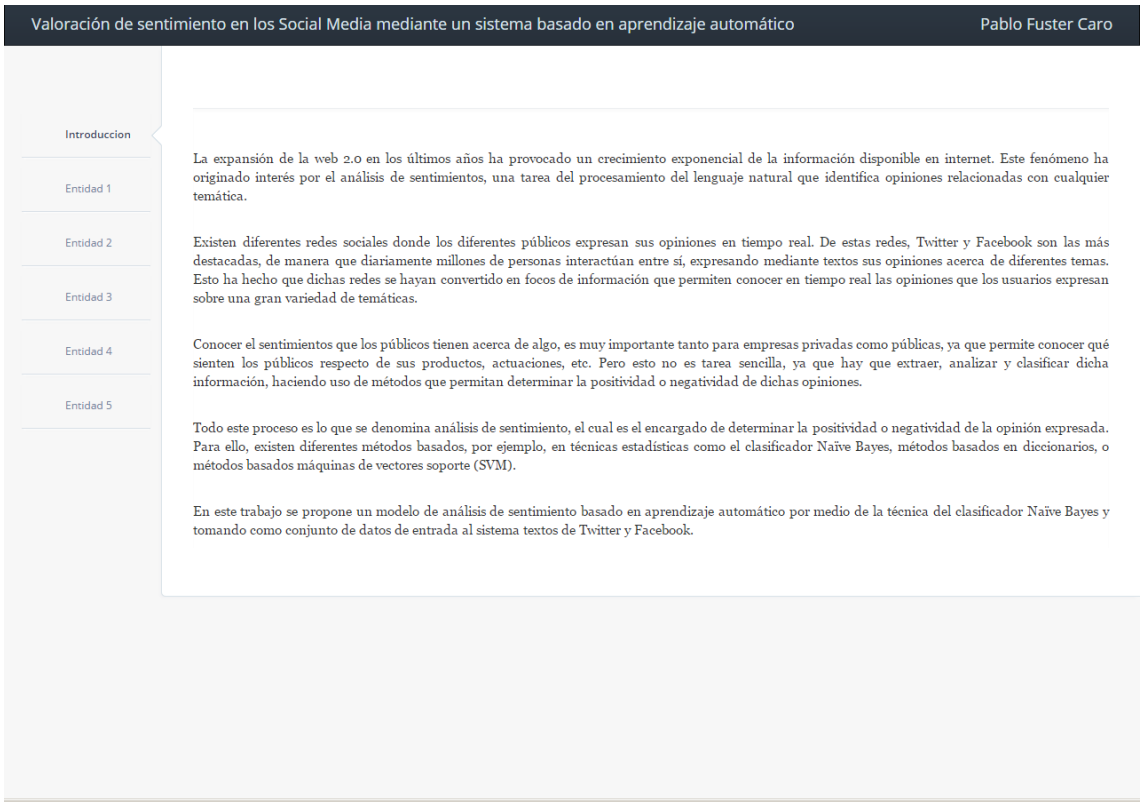


Fig. 12: Pantalla de Introducción de la Interfaz

La segunda pantalla (Fig.13) permite elegir entre las distintas entidades que se han evaluado en el trabajo y visualizar una serie de datos:

- Media de sentimiento diaria: muestra en un diagrama de barras la clasificación media de cada día durante una semana. Los valores que se le da a cada clase son:
    - Muy negativo: 0.
    - Negativo: 1.
    - Neutro 2.
    - Positivo: 3.
    - Muy positivo: 4.
- Cantidad de sentimientos: muestra la cantidad de documentos clasificados en cada clase durante el período de una semana.

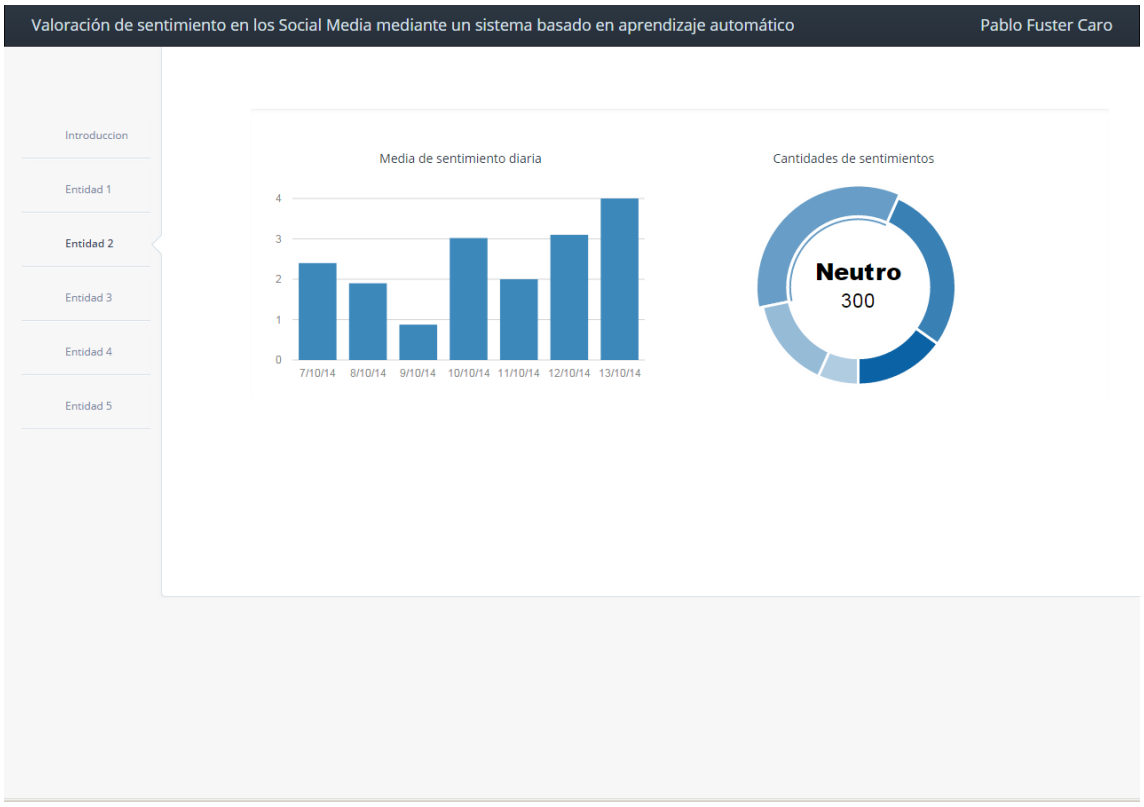


Fig. 13: Pantalla de Resultados de la Interfaz

6. Conclusiones

En este trabajo se pueden extraer dos conclusiones, unas primeras a nivel de iniciación a la investigación con el trabajo desarrollado; y unas segundas a nivel de formación por las herramientas, métodos, procesos que se han utilizado para la elaboración del presente trabajo. A continuación detallo cada una de ellas.

Conclusiones de Iniciación a la Investigación

En este trabajo se ha implementado un sistema el análisis de sentimiento para las redes sociales Twitter y Facebook. Para ello se ha implementado:

- Un sistema de lectura de la red social Facebook.
- Un sistema de lectura de la red social Twitter.
- Una base de datos para el almacenamiento de la información.
- Un algoritmo de aprendizaje automático para realizar análisis de sentimiento.
- Una interfaz web para la visualización de los resultados.

El modelo desarrollado ha presentado unos resultados favorables, ya que ha sido capaz de acertar el sentimiento y la intensidad del mismo en un 77.49%.

Conclusiones de Formación

Para el desarrollo de ese trabajo final de grado ha sido necesaria la utilización de sistemas de aprendizaje automático, de herramientas para la comunicación entre Java y la base de datos, de APIs para la retrospección de información de diferentes redes sociales y la utilización de bases de datos.

Durante la realización de este trabajo se han obtenido conocimientos de:

1. Manejo de bases de datos Microsoft SQL Server.
2. Intercambio de datos entre Java y la BBDD mediante Hibernate.

3. Lectura automática y gestión de la información en las redes sociales Facebook y Twitter.
4. Diseño e implementación de algoritmos de aprendizaje automático.
5. Análisis de sentimiento.
6. Ingeniería del software.

Futuros Trabajos

Como propuestas de mejora se podrían señalar las siguientes:

- Realización del sistema individualmente para cada medio, adecuando el sistema a cada lenguaje particular.
- Realización de otras técnicas de aprendizaje automático y unificarlas en un multiclasificador.

7. Referencias

Bing Liu. (2012). Sentiment Analysis and Opinion Mining. Morgan & Claypool Publishers. Recuperado de: http://www.cs.uic.edu/~liub/FBS/SentimentAnalysis-and-OpinionMining.pdf

Das Sanjiv y Mike Chen (2001). *Yahoo! for Amazon: Extracting market sentiment from stock message board.* Recuperado de: http://papers.ssrn.com/sol3/papers.cfm?abstract_id=276189

Dave Kushal, Steve Lawrence y Pennock. (2003). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. Recuperado de: http://www.kushaldave.com/p451-dave.pdf

Hu Minqing y Bing Liu. (2004). *Mining and summarizing customer reviews*. Department of Computer Science University of Illinois at Chicago. Recuperado de: http://www.cs.uic.edu/~liub/publications/kdd04-revSummary.pdf

Jindal, Nitin y Bing Liu. (2006). *Mining comparative sentences and relations*. American Association for Artificial Intelligence. Department of Computer Science University of Illinois at Chicago. Recuperado de: http://www.cs.uic.edu/~njindal/docs/aaai06-comp-relation.pdf

Kaji Nobuhiro y Masaru Kitsuregawa. (2007). *Building lexicon for sentiment analysis from massive collection of HTML documents*. Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 1075-1083. Recuperado de: http://www.aclweb.org/anthology/D07-1115

Kanayama, Hiroshi y Tetsuya Nasukawa. (2006). *Fully automatic lexicon expansion for domain-oriented sentiment analysis*. Association for Computancional Lunguistics, pp. 355-363. Recuperado de: http://dl.acm.org/citation.cfm?id=1610125

Morinaga, Satoshi, Kenji Yamanishi, Kenji Tateishi, y Toshikazu Fukushima (2002). *Mining product reputations on the web*. en Bing Liu. (2012). *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers.

Nasukawa, Tetsuya y J. Yi. (2003). *Sentiment analysis: Capturing favorability using natural language processing*. En: *Conference on Knowledge Capture,* en Bing Liu. (2012). Sentiment Analysis and Opinion Mining. Morgan & Claypool Publishers.

Pang Bo, Lillian Lee, y Shivakumar Vaithyanathan. (2002). *Thumbs up?: sentiment classification using machine learning techniques*. Recuperado de: http://www.cs.cornell.edu/home/llee/papers/sentiment.pdf

Peter D. Tourney (2002). *Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews*. Recuperado de: http://arxiv.org/ftp/cs/papers/0212/0212032.pdf

Richard M. Tong (2001). *An operational system for detecting and tracking opinions in on-line discussion*, en Bing Liu. (2012). Sentiment Analysis and Opinion Mining. Morgan & Claypool Publishers.

Taboada, Maite, Julian Brooke, Milan Tofiloski, Kimberly Voll, y Manfred Stede. (2011). *Lexicon-based methods for sentiment analysis.* Association for Computational Linguistics, **37**(2): pp. 267-307. Recuperado de: https://www.aclweb.org/anthology/J/J11/J11-2001.pdf

J.D. Power and Associates (2013). *Social Media Benchmark Study 2013*. Recuperado de: http://www.jdpower.com/press-releases/2013-social-media-benchmark-study

Turney, Peter D. y Micharel L. Littman. (2003). *Measuring praise and criticism: Inference of semantic orientation from association.* Recuperado de: http://cogprints.org/3164/1/turney-littman-acm.pdf

Wiebe, Janyce, Rebecca, F. Bruce y Thomas P. O'Hara. (1999). *Development and use of a gold-standard data set for subjectivity*. Recuperado de: http://www.aclweb.org/anthology/P99-1032

Anexos Técnicos

1. Descripción de entidades y atributos

ENT – 01	**INFORMACION**
Descripción	En esta tabla se encontrará almacenada cada unidad de información leída de Twitter o Facebook.
Atributos	ATR–01: id ATR–02: texto ATR–03: usuario ATR-04: fecha ATR-05: hora ATR-06: tipofuente ATR-07: valorado
ATR – 01	**INFORMACION :: id**
Descripción	Identificador único de cada información.
Tipo	BigInt
Comentarios	Clave primaria autoincrementada
ATR – 02	**INFORMACION :: texto**
Descripción	Texto que contiene la información almacenada.
Tipo	Varchar
Comentarios	Obligatorio
ATR – 03	**INFORMACION :: usuario**
Descripción	Nombre del usuario que emite la información
Tipo	Varchar
Comentarios	Obligatorio
ATR – 04	**INFORMACION :: fecha**
Descripción	Fecha de la emisión de la información
Tipo	Date
Comentarios	Optativo
ATR – 05	**INFORMACION :: hora**
Descripción	Hora de la emisión de la información
Tipo	Time
Comentarios	Optativo
ATR – 06	**INFORMACION :: tipofuente**
Descripción	Distingue si la información proviene de Twitter o Facebook.

Tipo	SmallInt
Comentarios	Obligatorio Valor 0: proviene de Twitter Valor 1: proviene de Facebook
ATR – 07	**TWEET :: valorado**
Descripción	Especifica si el tweet ha sido valorado ya o no
Tipo	Boolean
Comentarios	Obligatorio Valor inicial: Falso

ENT – 02	**ENTIDAD**
Descripción	Organismo, empresa o persona de las que se va a almacenar y valorar información.
Atributos	ATR-08: entidad_id ATR-09: nombre
ATR – 08	**ENTIDAD :: entidad_id**
Descripción	Identificador único de cada entidad
Tipo	Integer
Comentarios	Clave primaria autoincrementada
ATR – 09	**ENTIDAD :: nombre**
Descripción	Nombre de la entidad
Tipo	Varchar
Comentarios	Obligatorio Único

ENT – 03	**SINONIMO**
Descripción	Distintos términos por los que se van a buscar las diferentes entidades.
Atributos	ATR-10: sinonimo_id ATR-11: sinonimo_str
ATR – 10	**SINONIMO :: sinonimo_id**
Descripción	Identificador para cada sinónimo
Tipo	Integer
Comentarios	Clave primaria autoincrementada
ATR – 11	**SINONIMO :: sinonimo_str**

Descripción	Término de búsqueda asociado a una entidad
Tipo	Varchar
Comentarios	Obligatorio Único

ENT – 04	**CLASE**
Descripción	Distintas categorías en las que vamos a clasificar las informaciones
Atributos	ATR-12: clase_id ATR-13: nombre_clase ATR-14: prior
ATR – 12	**CLASE :: clase_id**
Descripción	Identificador para cada clase
Tipo	Smallint
Comentarios	Clave primaria autoincrementada
ATR –13	**CLASE :: nombre_clase**
Descripción	Nombre de la clase
Tipo	Varchar
Comentarios	Obligatorio Único
ATR – 14	**CLASE :: prior**
Descripción	Probabilidad a priori de que un documento sea de una clase
Tipo	Float
Comentarios	Obligatorio

ENT – 05	**STOP_WORD**
Descripción	Palabras vacías y sin valor en el procesamiento del lenguaje natural
Atributos	ATR-15: stopword
ATR – 15	**STOP_WORD :: stopword**
Descripción	Palabra catalogada como *palabra vacía*
Tipo	Varchar
Comentarios	Clave primaria

ENT – 06	**VALORACION**
Descripción	Contiene las clasificaciones de las informaciones
Atributos	ATR-16: valoracion_id
ATR – 16	**VALORACION :: valoracion_id**
Descripción	Identificador de cada valoración
Tipo	Integer
Comentarios	Clave primaria autoincrementada

ENT – 07	**PALABRA_RELEVANTE**
Descripción	Palabras clave que permitirán clasificar una información
Atributos	ATR-17: palabra_id ATR-18: palabra
ATR – 17	**PALABRA_RELEVANTE :: palabra_id**
Descripción	Identificador de cada palabra
Tipo	Integer
Comentarios	Clave primaria autoincrementada
ATR – 18	**PALABRA_RELEVANTE :: palabra**
Descripción	Cadena de caracteres que representa una palabra
Tipo	Varchar
Comentarios	Obligatorio Único

ENT – 08	**PROB_PALABRA_CLASE**
Descripción	Probabilidad de que una determinada palabra se de en una determinada clase
Atributos	ATR-19: probabilidad
ATR – 19	**PROB_PALABRA_CLASE :: probabilidad**
Descripción	Probabilidad condicionada de una palabra en una clase
Tipo	Float
Comentarios	Obligatorio

ENT – 09	**OCURRENCIA_PALABRA**
Descripción	Número de apariciones de una palabra en los textos de entrenamiento
Atributos	ATR-20: num_ocurrencias
ATR – 20	**OCURRENCIA_PALABRA :: num_ocurrencias**
Descripción	Veces que aparece una palabra en un texto de entrenamiento
Tipo	Integer
Comentarios	Obligatorio

ENT – 10	**DOC_ENTRENAMIENTO**
Descripción	Informaciones que constituyen el conjunto de entrenamiento
Atributos	doc_id
ATR – 20	**DOC_ENTRENAMIENTO :: doc_id**
Descripción	Identificador de cada documento del conjunto de entrenamiento
Tipo	Integer
Comentarios	Clave primaria autoincrementada

2. Descripción de relaciones entre entidades

REL – 01	**Origen**	ENT–02	**Mult**	**Destino**	ENT–03	**Mult**
definida_sobre	ENTIDAD		1	SINONIMO		1..*
Calificador	Tiene			Pertenece		
Descripción	Una entidad tiene uno o más sinónimos asociados. Un sinónimo pertenece a una única entidad.					
Optatividad	NO			NO		

REL – 02	**Origen**	ENT–03	**Mult**	**Destino**	ENT–01	**Mult**
definida_sobre	SINONIMO		1	INFORMACION		*
Calificador	Tiene			Habla de		
Descripción	Un sinónimo puede tener o no informaciones Una información habla de un sinónimo					

Optatividad	SI	NO

REL – 03	**Origen**	ENT–06	**Mult**	**Destino**	ENT–01	**Mult**
definida_sobre	VALORACION		1	INFORMACION		1
Calificador	Es de			Genera		
Descripción	Una valoración es de una información Una información genera una valoración					
Optatividad	NO			SI		

REL – 04	**Origen**	ENT–06	**Mult**	**Destino**	ENT–04	**Mult**
definida_sobre	VALORACION		*	CLASE		1
Calificador	Clasificada en			Tiene		
Descripción	Una valoración está clasificada en una clase Una clase puede tener o no valoraciones					
Optatividad	SI			NO		

REL – 05	**Origen**	ENT–10	**Mult**	**Destino**	ENT–01	**Mult**
definida_sobre	DOC_ENTRENAMIENTO		1	INFORMACION		1
Calificador	Es un			es		
Descripción	Un doc_entrenamiento es una información Una información puede ser un doc_entrenamiento					
Optatividad	NO			SI		

REL – 06	**Origen**	ENT–10	**Mult**	**Destino**	ENT–04	**Mult**
definida_sobre	DOC_ENTRENAMIENTO		*	CLASE		1
Calificador	Clasificado en			Tiene		
Descripción	Un doc_entrenamiento está clasificado en una clase Una clase puede tener cero o más de un doc_entrenamiento					
Optatividad	SI			NO		

REL – 07	**Origen**	ENT–08	**Mult**	**Destino**	ENT–04	**Mult**
definida_sobre	PROB_PALABRA_CLASE		*	CLASE		1
Calificador	Es de			Tiene		
Descripción	Una prob_palabra_clase es de una clase Una clase tiene cero o más prob_palabra_clase					
Optatividad	SI			NO		

REL – 08	**Origen**	ENT–08	**Mult**	**Destino**	ENT–07	**Mult**
definida_sobre	PROB_PALABRA_CLASE		*	PALABRA_RELEVANTE		1
Calificador	Es de			Tiene		
Descripción	Una prob_palabra_clase es de una palabra_relevante Una palabra_relevante tiene cero o más prob_palabra_clase					
Optatividad	SI			NO		

REL – 09	**Origen**	ENT–09	**Mult**	**Destino**	ENT–07	**Mult**
definida_sobre	OCURRENCIA_PALABRA		*	PALABRA_RELEVANTE		1
Calificador	Es de			Tiene		
Descripción	Una palabra_relevante tiene cero o más ocurrencia_palabra Una ocurrencia_palabra es de una palabra_relevante					
Optatividad	SI			NO		

REL – 10	**Origen**	ENT–09	**Mult**	**Destino**	ENT–10	**Mult**
definida_sobre	OCURRENCIA_PALABRA		*	DOC_ENTRENAMIENTO		1
Calificador	Es de			Tiene		
Descripción	Una ocurrencia_palabra es de un doc_entrenamiento Un doc_entrenamiento tiene cero o más ocurrencia_palabra					
Optatividad	SI			NO		