ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
INGENIERÍA INFORMÁTICA: MENCIÓN EN SISTEMAS DE LA INFORMACIÓN

MINERÍA DE OPINIONES, BASDADA EN PATRONES SEMÁNTICOS, EN TWITTER

OPINION MINING, BASED ON SEMANTIC PATTERNS, ON TWITTER

Realizado por
Pablo Artacho Torres
Tutorizado por
José Ignacio Peláez Sánchez
Departamento
Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, SEPTIEMBRE 2014

Fecha defensa:
El Secretario del Tribunal

Resumen: La minería de opinión o análisis de sentimiento es un tipo de análisis de texto que pretende ayudar a la toma de decisiones a través de la extracción y el análisis de opiniones, identificando las opiniones positivas, negativas y neutras; y midiendo su repercusión en la percepción de un tópico. En este trabajo se propone un modelo de análisis de sentimiento basado en diccionarios, que a través de la semántica y de los patrones semánticos que conforman el texto a clasificar, permite obtener la polaridad del mismo, en la red social Twitter. Para el conjunto de datos de entrada al sistema se han considerado datos públicos obtenidos de la red social Twitter, de compañías del sector de las telecomunicaciones que operan en el mercado Español.

Palabras claves: minería de datos, minería de opinión, análisis de sentimiento, Twitter, patrones semánticos.

Abstract: Sentiment analysis or opinion mining is a type of text analysis that aims to help decision-making through the extraction and sentiment analysis, identifying the positive, negative and neutral opinions, and measuring its impact on the perception of a topic. This work proposes a model based on dictionaries, that through the semantic and the semantic patterns that define the text to be classified, a polarity is obtained. For all necessary input data for the system, we have considered public data from the social network Twitter, taking various telephone companies operating in the Spanish market.

Keywords: data mining, opinion mining, sentiment analysis, Twitter, semantic patterns

Índice

1. Introducción

La web Social ha cambiado la forma en la que se genera y se consume la información, posibilitando que todos nosotros seamos generadores de contenidos que compartimos en nuestro día a día con aquellas personas que deseamos o con el público en general.

Las previsiones sobre el crecimiento de información digital siguen al alza, en gran parte gracias al aumento de las tendencias como la movilidad, el cloud computing, el consumo de vídeo, el uso de redes sociales, etc. Desde 2006 se han generado más datos de los que la humanidad había producido en todo su recorrido anterior. La explosión no ha hecho más que comenzar. Se estima que para 2020 circularán 35.2 ZB frente a los 1.8 que se alcanzaron en el año 2011.

Este espectacular crecimiento del volumen de datos en internet va de la mano al crecimiento de dispositivos conectados a la red. Esto provoca cambios en las empresas, las cuales están evolucionando hacia lo que se denomina *Big Data*, para explotar sus datos y obtener un valor añadido en sus negocios.

Las opiniones y comentarios vertidos por los consumidores sobre productos y servicios son de gran importancia para las empresas. Por un lado, permiten conocer la satisfacción de los clientes sobre los productos o servicios ofertados y, por otro, son de gran utilidad para delimitar los distintos segmentos del mercado en función de las características generales de los diferentes tipos de consumidores de un producto o servicio.

El diseño y desarrollo de nuevas tecnologías capaces de procesar eficientemente esta información ha despertado el interés de la comunidad científica y empresarial, surgiendo toda una disciplina de investigación conocida como *minería de opinión*.

El *análisis de sentimiento* (minería de opinión) se centra en el estudio de contenidos no estructurados y en la obtención de una *polaridad* (positiva, neutra, negativa). Actualmente existen dos perspectivas bien diferenciadas para abordar el problema; las técnicas de aprendizaje automático y los sistemas basados en diccionarios.

El modelo que se propone a lo largo del trabajo se centra en la obtención de datos vertidos en la red social de Twitter para realizar un análisis semántico basado en diccionarios, de forma que a través de la semántica de los mismos, podamos obtener su polaridad.

La memoria del proyecto se encuentra divida en cinco apartados. En el primero de ellos, denominado *Estado del arte*, se explicará cómo es enfocado el análisis semántico en la actualidad. En el segundo se presentará el modelo que se llevará a cabo. Para ello utilizaremos el método deductivo, partiendo de lo general, es decir, de la arquitectura del sistema, hasta llegar a lo particular. En el apartado de *Evaluación* se expondrán los resultados obtenidos con la puesta en marcha del sistema. Concluido este apartado, se

abordarán las implementaciones realizadas a lo largo del proyecto: base de datos, lectura de Twitter e interfaz. En el último apartado, se mostrarán las conclusiones obtenidas.

2. Estado del arte

El análisis de sentimiento, también conocido como minería de opinión es el área de estudio que analiza las opiniones de la gente, sus sentimientos, evaluaciones, actitudes y emociones hacia entidades tales como productos, servicios, empresas, individuos, temas y atributos. En la actualidad existe una gran diversidad de nombres para hacer alusión al mismo ámbito de estudio (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 Dave, Lawrence y Pennock en 2003 y Nasukawa y Yi, en el mismo año respectivamente; las primeras investigaciones relativas a opiniones y sentimientos aparecieron a partir del año 2000, como se puede comprobar en el 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.

A la hora de realizar minería de opinión es conveniente saber que para que una opinión pueda ser analizada deben existir en ella dos atributos elementales:

- El objetivo. Hace alusión al tópico del cual se busca el sentimiento.
- El Sentimiento. Evalúa la polaridad referida al objetivo.

Para abordar este problema, una de las soluciones más comunes está basada en la premisa expuesta a continuación:

> Si un texto contiene más palabras positiva que negativas, es un texto positivo; si este contiene más palabras negativas que positivas se trata de un texto negativo

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 requiere.

Calcular la polaridad de un texto en función de las palabras de sentimiento (*sentiment words*) que este contenga resulta insuficiente, dado que la **semántica** influye a la hora de determinar la polaridad del mismo. Se entiende por semántica, el estudio de la relación entre palabras, frases, signos y sus denotaciones. 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.

La minería de opinión puede enfocarse desde diferentes perspectivas dependiendo del nivel 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 en 2002 y Turney también en 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. Esta clasificación está altamente relacionada con clasificación de subjetividad (Wiebe, Bruce y O'Hara, 1999), que distingue entre frases denominadas frases objetivas (que representan hechos) de frases subjetivas (representan opiniones).
- **Nivel Entidad-Aspecto**, es un análisis de sentimiento con mayor granularidad. Partiendo de un tópico, selecciona los aspectos que se desean analizar y evalúa el sentimiento para cada uno de ellos. En un principio se denominó *feature level* (feature-based opinion mining and summarization) (Hu y Liu, 2004).

Para hacer el análisis de sentimiento más complicado aún si cabe, en la literatura se distinguen dos tipos de opiniones (Jindal and Liu, 2006b):

- *Regular opinions*
- *Comparative opinions*

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 anteriormente explicado, se exponen los problemas más comunes referentes a la minería de opinión.

- **Contexto**, dependiendo del contexto una misma palabra puede presentar polaridades diferentes.
- **Ambigüedad del sentimiento**, del mismo 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 para resolver estas problemáticas radica en el uso de un diccionario. En la literatura se documentan dos procedimientos para elaborar los diccionarios. Taboada y colaboradores en 2011 y Tong y colaboradores en 2001 explican que estos pueden ser creados de forma manual. Por el contrario, Kanayama y colaboradores en 2006, Kayiy Kitsuregawa en 2007, Turney en 2002 y Turney y Littman 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 está basado en diccionarios, de forma que a través de la semántica de dichos diccionarios se determinará la polaridad (positiva, neutra, negativa) de textos extraídos de *Twitter* (análisis a nivel de documento). La polaridad estará representada por un valor comprendido entre [0,10]. De este modo, el sistema recibirá como entrada un texto proveniente de un tweet y generará como salida la polaridad asociada a éste, en función de la estructura del texto y los *patrones semánticos* definidos.

Para la elaboración del sistema contamos con dos conjuntos de tweets manualmente clasificados, siendo la muestra total entre los dos conjuntos de 1416 elementos de información:

- Conjunto de entrenamiento
- Conjunto de prueba

El conjunto de entrenamiento nos servirá de referencia para elaborar el diccionario de sentimiento y los patrones semánticos, y para comparar los resultados obtenidos a través del algoritmo con los conseguidos con los resultados provenientes de los datos manualmente clasificados. El conjunto de prueba se compone de una muestra totalmente independiente de tweets que nos servirá para determinar la tasa de acierto.

3.1 Elaboración de diccionarios

El sistema que se ha desarrollado hará uso de dos diccionarios bien diferenciados, tanto por su uso como por su contenido.

- Diccionario de Apertium
- Diccionario de Sentimiento

El diccionario de español de *Apertium* es un recurso online disponible que usa su etiquetador para realizar las labores de etiquetado (el uso de este recurso, así como las funciones de Apertium serán detalladas en posteriores epígrafes).

El *diccionario de sentimiento* será elaborado de forma manual y estará formado por un listado de palabras asociadas a una categoría gramatical y a una polaridad. Esta estructura nos permitirá determinar la polaridad de una palabra siempre y cuando sepamos la categoría gramatical a la que pertenece.

En la tabla 1 mostrada a continuación se presenta un ejemplo con algunas posibles combinaciones entre categorías gramaticales y las polaridades que estas pueden adquirir.

Tabla 1. Combinaciones entre categorías gramaticales y polaridades

POLARIDAD / ***CAT.GRAMATICAL***	MP	P	DEFAULT	N	MN	I
ADJETIVO	x	x	x	x	x	x
NOMBRE	x	x	x	x	x	
ADVERBIO	x	x	x	x	x	
VERBO	x	x	x	x	x	x
NEGACION			x			
ENTIDAD			x	x		
DEFAULT			x			

Para esclarecer los datos mostrados en la tabla anterior, se explicará de forma más detallada el significado de las categorías gramaticales que se recogen en el diccionario, así como sus correspondientes polaridades.

Las categorías gramaticales adjetivo, nombre, adverbio y verbo se corresponden a sus semejantes existentes en el castellano. En contraposición, los términos entidad, negación y default definen una serie de categorías establecidas por nosotros, las cuales se detallan a continuación:

- ENTIDAD se refiere al tópico del que se busca un sentimiento.
- La categoría NEGACION está formada por las palabras que niegan el contexto de una frase.
- Utilizamos DEFAULT para especificar cualquier palabra que no esté encasillada en ninguna de las categorías descritas.

Tabla 2. Ejemplo de categorías definidas por nosotros

ENTIDAD	Sería bueno... Telefónica España estaría lanzando un canal de series antes de terminar el año
NEGACION	Madrugar todos los días no puede ser bueno
DEFAULT	Hola, Color favorit... — Lila. Huevos fritos con patatas fritas. Ono million p

Aunque el sistema sólo clasifique en tres polaridades (positivo, negativo, neutro), a lo largo del proceso se manejará una mayor granularidad. De este modo, las polaridades quedan definidas como se detalla:

- Las polaridades *MP*/*P* se corresponden a la polaridad positiva y equivalen a muy positivo y positivo respectivamente.
- *MN*/*N* se corresponden a la polaridad negativa, siendo muy negativo y negativo.
- Se usa la polaridad DEFAULT para representar las palabras que aun teniendo una categoría no determinan ningún sentimiento.
- La polaridad I (inversor) identifica a las palabras capaces de invertir el sentimiento de las palabras a las que acompaña.

A continuación se muestran unos ejemplos de la polaridad I:

Tabla 3. Ejemplos de la polaridad I

NEGACION	Telefónica *cancela* el despido de 300 de sus empleados	Positivo
DEFAULT	Telefónica *cancela* su expansión por Latinoamérica	Negativo

Una vez que se han aclarado todos los términos que componen el diccionario, se detalla cómo se llevó a cabo la elaboración del mismo.

1. Partiendo de los tweets del conjunto de entrenamiento se extraerán todas las palabras que determinan el sentimiento del texto.
2. Se usarán como referencia algunos de los diccionarios de sentimientos más notorios disponibles en Internet.
3. Se realizarán derivaciones morfológicas a las palabras que en este apartado componen el diccionario.
4. Finalmente, todas las palabras que conforman el diccionario serán consideradas semillas, permitiendo la expansión del diccionario mediante el uso de sinónimos.

3.1.1 Estructura del diccionario de sentimiento

El diccionario se encuentra almacenado en un fichero *xml* que está definido con la siguiente estructura:

Tabla 4. Estructura del diccionario de sentimiento

Estructura	Ejemplo
<diccionario> <tag valor=”Categoria+Polaridad”> <palabra valor=”entrada”/> </tag> </diccionario>	<diccionario> <tag valor=’VerboP”> <palabra valor=”amar” /> </tag> </diccionario>

3.2 Obtención de patrones semánticos

Para calcular el sentimiento de los distintos textos que alimentan al sistema será necesario tener un conjunto de *patrones semánticos* asociados a un *valor* $\in$ *[0,10].* Esta asociación permitirá clasificar el texto de entrada en función de estos patrones.

Un patrón semántico queda definido como una sucesión de las posibles combinaciones entre categorías gramaticales y polaridades del diccionario que determinan de forma parcial o total la polaridad del texto. A continuación se muestra un ejemplo de cómo obtener un patrón dado un texto de entrada.

Tabla 5. Ejemplo de patrón a partir de texto

Texto	[Tópico]	es	bueno
Patrón (EtiquetaSentimiento)	ENTIDADDEFAULT	VERBODEFAULT	ADJETIVOP

Teniendo el patrón del texto, se decidirá qué elementos del mismo son los que determinan la polaridad del texto para un tópico. Además se le asociará un valor que indique su polaridad. En este ejemplo el patrón semántico que determinaría la polaridad el texto, sería el siguiente:

Tabla 6. Patrón semántico

Patrón Semántico		**Valor**
ENTIDADDEFAULT	ADJETIVOP	7

Una vez presentados los patrones semánticos, queda por elaborar de forma manual un conjunto de estructuras similares a las mostradas en la figura anterior (Patrón Semántico + Valor). Actuando de forma análoga a la elaboración del diccionario de sentimiento, haremos uso del *conjunto de entrenamiento* para la obtención de los patrones semánticos.

El procedimiento que se ha seguido para la elaboración de los patrones semánticos se explica a continuación:

1. Partiendo de los tweets del conjunto de entrenamiento se definirán un conjunto de patrones semánticos y sus respectivas valoraciones que determinan el sentimiento del documento de entrada.
2. Haciendo uso de los patrones semánticos ya existentes, se añadirán los patrones semánticos complementarios que no se encuentren definidos.
3. Finalmente se harán combinaciones lógicas para obtener patrones semánticos que no se hayan determinado hasta este momento.

3.3 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, se muestra la modularización del sistema y se detallan las funciones de cada uno de los módulos.

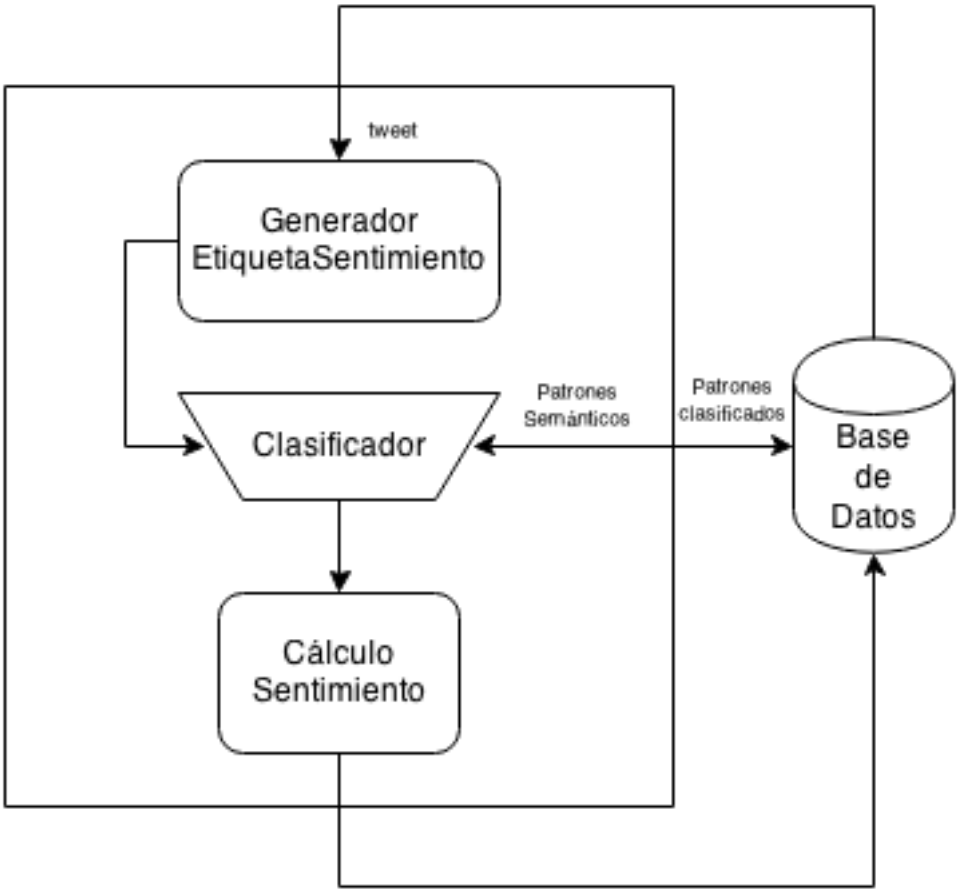


Ilustración 1. Arquitectura del sistema

- *Generador EtiquetaSentimiento:* teniendo como entrada el texto de un tweet, este módulo realiza una serie de procesos de normalización del texto para posteriormente elaborar el etiquetado a través del cual se generarán las EtiquetaSentimiento correspondientes al texto de entrada.
- *Clasificador:* recibe como entradas las EtiquetaSentimiento y los patrones semánticos almacenados en la base de datos para realizar la clasificación de las EtiquetaSentimiento en función de los patrones semánticos.
- *Calcular Sentimiento:* teniendo como entrada los patrones semánticos que clasificaron al texto inicial, se aplican una serie de funciones matemáticas para obtener la valoración final.

3.3.1 Generador EtiquetaSentimiento

Este módulo contiene a su vez una serie de módulos internos, cada uno de ellos cumple con unas funciones determinadas. La siguiente figura representa la estructuración interna del módulo.

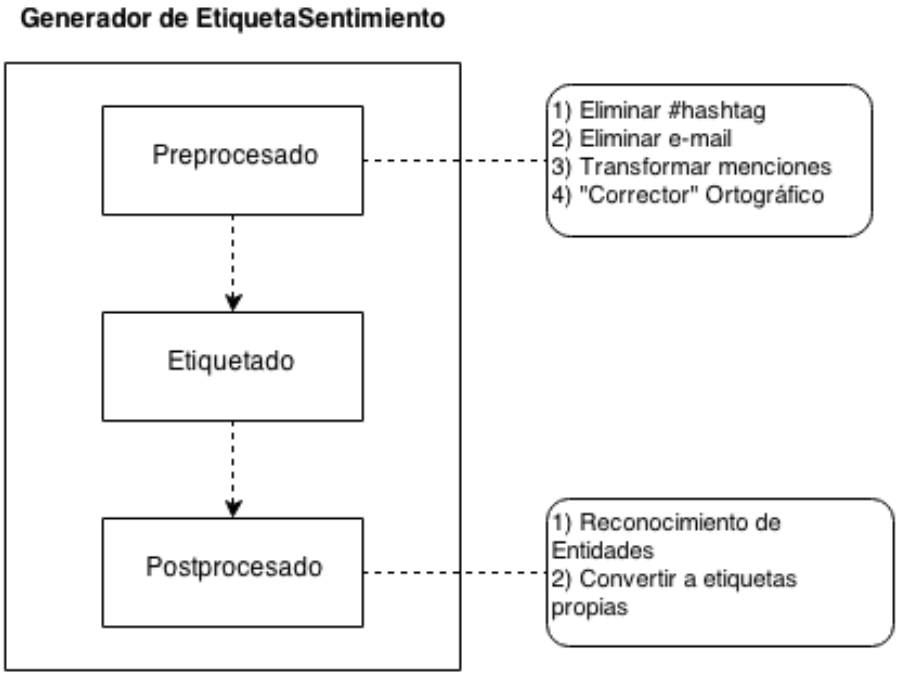


Ilustración 2. Módulos de Generador EtiquetaSentimiento

3.3.1.1 Preprocesado

Es frecuente que un tweet 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 por ello que *normalizar el texto* resulta una labor básica para el buen funcionamiento de etapas posteriores.

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. Esto genera fallos a la hora de clasificar los patrones.

Tabla 7. Ejemplo de cómo tratar hashtag

Entrada	Esta conexión no es lenta sino lo siguiente *#movistar*
Salida	Esta conexión no es lenta sino lo siguiente # *movistar*

Si no se hiciera esta corrección, el etiquetador no reconocería el token #movistar a no ser que éste estuviera contemplado en su diccionario. Si bien lo anteriormente expuesto es una solución válida, resulta una alternativa poco escalable. Por ello, la solución escogida para solventar este problema es detectar los *hashtag* de un tweet y añadir un espacio entre el carácter # y la palabra que lo acompaña.

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 '@'.

Tabla 8. Ejemplo de cómo tratar menciones

Entrada	tengo un problema con las caídas de internet @Movistar
Salida	tengo un problema con las caídas de internet @ Movistar

La solución adoptada para la solución del problema es la misma que la expuesta en el caso anterior.

Eliminación de URL

En los *tweets* 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.

Tabla 9. Ejemplo de eliminación de URL

Entrada	¿Conoces Talentum Universities, el programa de becas de Telefónica para jóvenes hasta 30 años? http://ow.ly/AB5K6
Salida	¿Conoces Talentum Universities, el programa de becas de Telefónica para jóvenes hasta 30 años?

Corrección de palabras con terminación exagerada

En Twitter es frecuente encontrar palabras mal escritas. En muchos casos estos errores siguen un patrón predefinido, tratándose por lo general de las repeticiones indefinidas de una o varias letras dentro de una misma palabra.

Sabiendo esto es posible implementar un algoritmo que teniendo en cuenta las reglas del español elimine cualquier repetición de dos o más letras en una palabra, dejando dos repeticiones las situaciones más comunes que determina la RAE.

Tabla 10. Ejemplo de corrección ortográfica

Entrada	holaaaaa, casasssss, caosss
Salida	hola, casas, caos

3.3.1.2 Etiquetado

Una vez se ha normalizado el texto de entrada, es preciso etiquetar (*part-of-speech tagging*) la salida del módulo de preprocesado. 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 del ya mencionado etiquetador. La siguiente figura muestra los módulos de Apertium que se usarán en nuestro sistema así como su flujo de datos.

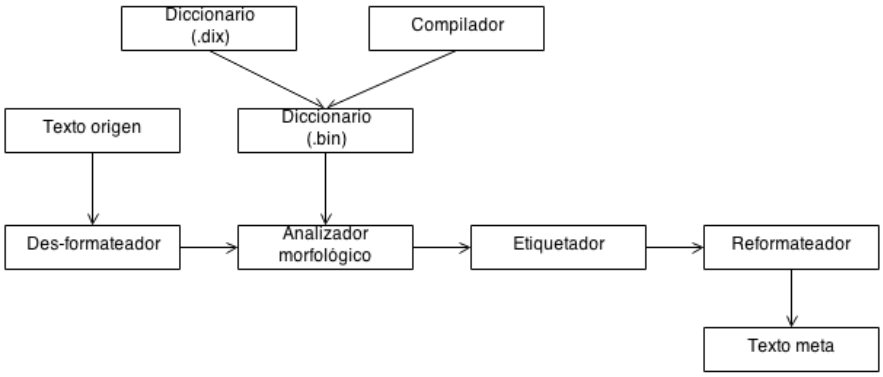


Ilustración 3. Flujo de datos y módulos para el Etiquetado

Tabla 11. Función de los módulos del Etiquetado

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 y la información de flexión morfológica.
Etiquetador	Elige usando un modelo estadístico (modelo oculto de Markov), uno de los análisis de una palabra ambigua de acuerdo con su contexto.
Re-formateador	Reintegra la información de formato original al texto traducido

A modo de explicación se muestra un ejemplo en el que dado un texto origen se comprueba la entrada (equivalente a la salida del módulo anterior) y salida de cada uno de los módulos.

Tabla 12. Funcionamiento del módulo de Etiquetado

MÓDULO	SALIDA
Texto origen	Telefónica toma malas decisiones
Des-formateador	Telefónica toma malas decisiones.[]
Analizador morfológico	Telefónica /Telefónica<np><al> /Telefónica<np><al> /Telefónico<adj><f><sg>$ Toma /toma<n><f><sg> /tomar<vblex><pri><p3><sg> /tomar<vblex><imp><p2><sg> Malas /malo<adj><f><pl> Decisiones /decisión<n><f><pl> /.<sent> []
Etiquetador	Telefónica/Telefónica<np><al> toma/tomar<vblex><pri><p3><sg> malas/malo<adj><f><pl> decisiones/decisión<n><f><pl> ./.<sent> []
Re-formateador	Telefónica/Telefónica<np><al> toma/tomar<vblex><pri><p3><sg> malas/malo<adj><f><pl> decisiones/decisión<n><f><pl> ./.<sent>
Texto meta	*Telefónica/Telefónica<np><al>* *toma/tomar<vblex><pri><p3><sg>* *malas/malo<adj><f><pl>* *decisiones/decisión<n><f><pl>* *./.<sent>*

* Para hacer más sencilla la lectura, a la salida mostrada se le han quitado una serie de caracteres especiales que Apertium usa como delimitadores.

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

3.3.1.3 Postprocesado

La etapa de postprocesado consta de dos pasos:

1. Almacenar las etiquetas generadas por el etiquetador en una estructura de datos.
2. Convertir las etiquetas de la estructura de datos a *etiquetas de sentimiento*.

Nuestro cometido al finalizar esta fase consiste en obtener el *patrón* del texto origen, que estará formado por una lista de *EtiquetaSentimiento* del mismo.

Almacenar la información generada 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.

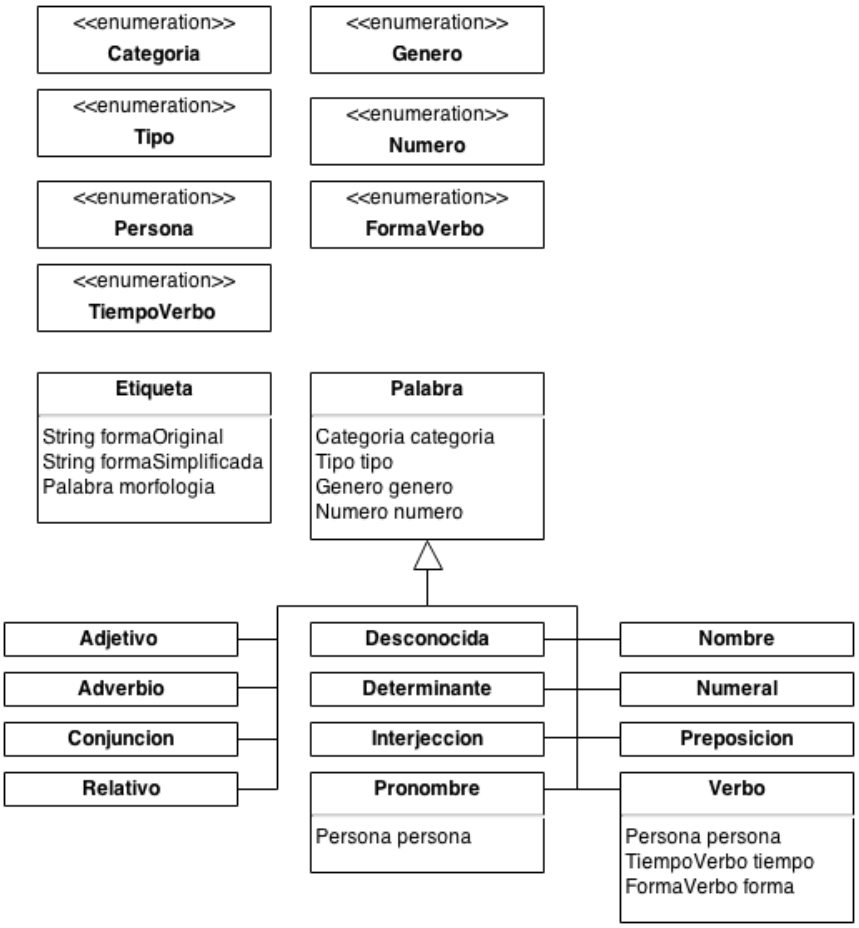


Ilustración 4. Estructura de datos

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.

Tabla 13. Ejemplo de funcionamiento del parseador

PARSEADOR			
ENTRADA *(String)*			
Telefónica/Telefónica<np><al> *toma/tomar<vblex><pri><p3><sg>* *malas/malo<adj><f><pl>* *decisiones/decisión<n><f><pl>*			
SALIDA *(List <Etiqueta>)*			
$Etiqueta_1$	formaOriginal	Telefónica	
	formaSimplificada	Telefónica	
	Palabra	Categoría	NP
		Tipo	AL
		Genero	DEFAULT
		Numero	DEFAULT
$Etiqueta_2$	formaOriginal	toma	
	formaSimplificada	tomar	
	Palabra	Categoría	VBLEX
		Tipo	DEFAULT
		Genero	DEFAULT
		Numero	SG
		Persona	P3
		Tiempo	PRI
		Forma	DEFAULT
$Etiqueta_3$	formaOriginal	malas	
	formaSimplificada	malo	
	Palabra	Categoría	ADJ
		Tipo	DEFAULT
		Genero	F
		Numero	PL
$Etiqueta_4$	formaOriginal	decisiones	
	formaSimplificada	decisión	
	Palabra	Categoría	N
		Tipo	DEFAULT
		Genero	F
		Numero	PL

Convertir las Etiquetas a EtiquetaSentimiento

Hasta el inicio de este apartado todos los pasos realizados son comunes a la gran mayoría de técnicas de análisis de sentimiento. Es a partir de este momento donde se hará uso de los conceptos específicos referentes a nuestra técnica.

El primer paso consiste en transformar las *Etiquetas* generadas a *EtiquetaSentimiento*. A continuación se muestra un diagrama que busca esclarecer el concepto de EtiquetaSentimiento.

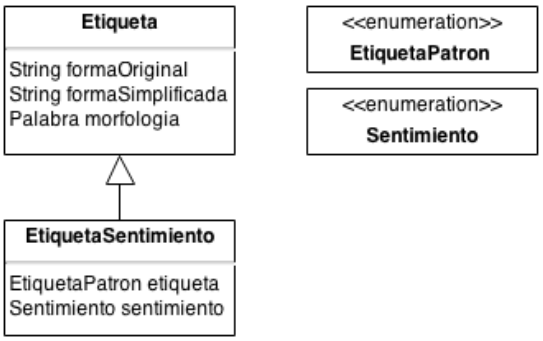


Ilustración 5. Modelo de EtiquetaSentimiento

La clase EtiquetaSentimiento además de contener la Etiqueta a partir de la cual se generó, tendrá una categoría gramatical que vendrá definida por una EtiquetaPatron y un sentimiento determinado por un valor perteneciente a un conjunto de valores predefinidos.

En la siguiente tabla se muestra un listado de posibles EtiquetaSentimiento.

Tabla 14. Posible combinación entre EtiquetaPatron y Sentimiento

ETIQUETASENTIMIENTO	
ETIQUETAPATRON	**SENTIMIENTO**
ADJETIVO	MP,P,DEFAULT,N,MN
NOMBRE	MP,P,DEFAULT,N,MN
VERBO	MP,P,DEFAULT,N,MN
ADVERBIO	MP,P,DEFAULT,N,MN
ENTIDAD	DEFAULT,N
NEGACION	DEFAULT
DEFAULT	DEFAULT

Introducido el concepto de EtiquetaSentimiento, queda explicar el algoritmo que transformará una Etiqueta a una EtiquetaSentimiento. Este se divide en dos pasos:

1. En función de la categoría de la Etiqueta que queremos transformar, se debe determinar su equivalente EtiquetaPatron.
2. Consultar en el diccionario de sentimiento si la formaSimplificada de la Etiqueta a transformar es entrada en el diccionario. De ser así, se le asignará el sentimiento que el diccionario determine, por el contrario se le asignará DEFAULT.

En la siguiente tabla se pueden observar las correspondencias que existen entre la categoría de una Etiqueta y una EtiquetaPatron.

Tabla 15. Correspondencias entre Etiqueta y EtiquetaPatron

CATEGORIAS	ETIQUETASENTIMIENTO	
	ETIQUETAPATRON	SENTIMIENTO
ADJ	ADJETIVO	MP,P,DEFAULT,N,MN
N	NOMBRE	MP,P,DEFAULT,N,MN
NP	NOMBRE	MP,P,DEFAULT,N,MN
	ENTIDAD	DEFAULT,N
VBSER	VERBO	MP,P,DEFAULT,N,MN
VBHAVER	VERBO	MP,P,DEFAULT,N,MN
VBLEX	VERBO	MP,P,DEFAULT,N,MN
VBMOD	VERBO	MP,P,DEFAULT,N,MN
ADV	ADVERBIO	MP,P,DEFAULT,N,MN
	NEGACION	DEFAULT
OTRAS*	DEFAULT	DEFAULT

* Otras: resto de categorías que puede devolver el etiquetador y que no se contemplan para definir una EtiquetaSentimiento

Aplicando lo expuesto a nuestro ejemplo concreto tenemos que:

Tabla 16. Ejemplo de funcionamiento del módulo de postprocesado

Etiqueta a EtiquetaSentimiento			
Etiqueta :: Categoria	Etiqueta:: formaSimplificada	EtiquetaSentimiento	
		EtiquetaPatron	Sentimiento
NP	Telefónica	ENTIDAD	DEFAULT
VBLEX	tomar	VERBO	DEFAULT
ADJ	malo	ADJETIVO	N
N	decisión	NOMBRE	DEFAULT

Al final de la fase de postprocesado, el patrón del texto que se ha generado es el siguiente:

Tabla 17. Salida del módulo de postprocesado

PATRON			
ETIQUETASENTIMIENTO$_1$	**ETIQUETASENTIMIENTO$_2$**	**ETIQUETASENTIMIENTO$_3$**	**ETIQUETASENTIMIENTO$_4$**
ENTIDADDEFAULT	VERBODEFAULT	ADJETIVON	NOMBREDEFAULT

3.3.2 Clasificador mediante patrones semánticos

Una vez obtenido el patrón de un texto es necesario clasificarlo mediante los *patrones semánticos*, para obtener el conjunto de patrones semánticos que clasifican al patrón del texto.

Para la clasificación del patrón del texto es necesario explicar bajo qué condiciones un patrón semántico clasifica a un patrón. Para ello, partiremos de un ejemplo.

Dado el siguiente patrón de un texto

Tabla 18. Ejemplo de patrón de un texto

PATRON
ENTIDADDEFAULT VERBODEFAULT ADJETIVON NOMBREDEFAULT

Y este conjunto de patrones semánticos

Tabla 19. Ejemplo de patrones semánticos

PATRON SEMÁNTICO			
ID	**PATRON**	**VALORACIÓN**	**LONGITUD**
1	ENTIDADDEFAULT ADJETIVON	3	2
2	ENTIDADDEFAULT ADVERBIOP ADJETIVON	1	3
3	ENTIDADDEFAULT ADJETIVOP	7	2
4	ENTIDADDEFAULT VERBON ADJETIVOP	3	3
5	ENTIDADDEFAULT ADJETIVON NOMBREDEFAULT	3	3

Diremos que un patrón es clasificado por un patrón semántico siempre y cuando el patrón semántico esté incluido en el patrón en el mismo orden. El patrón puede contener al patrón semántico de forma separada, es decir, pueden existir separaciones intermedias en el patrón con respecto al patrón semántico.

Usando el ejemplo expuesto se tiene que:

- El patrón semántico con id 1 clasifica al patrón existiendo una separación intermedia en el patrón con respecto al patrón semántico (VERBODEFAULT)

- El patrón semántico con id 5 clasifica al patrón existiendo una separación intermedia en el patrón con respecto al patrón semántico. (VERBODEFAULT)

Cabe destacar que un patrón puede ser clasificado por cero o más patrones semánticos. Esto generará especial importancia en el *modo* usado a la hora de calcular el sentimiento.

En esta fase de clasificación se tendrán en cuenta un par de parámetros que nos serán de gran utilidad a la hora de realizar el cálculo del sentimiento.

- Dependiendo de la *longitud* del patrón semántico que clasifique al patrón, la probabilidad de acierto variará de forma directa (a mayor longitud del patrón semántico, mayor probabilidad de acierto).
- Se tendrá en cuenta la *máxima separación intermedia* existente entre el patrón y el patrón semántico. Esto condicionará la probabilidad de indirecta (a menor separación intermedia, mayor probabilidad de acierto).

Esta serie de parámetros conlleva una serie de implicaciones. Se empezará a clasificar por los patrones semánticos de mayor *longitud*, y una vez que el patrón sea clasificado por un patrón semántico de longitud X, no es necesario seguir clasificando con los patrones de longitud interior a X.

Aplicando los conceptos explicados al ejemplo anterior obtenemos que:

PATRON
ENTIDADDEFAULT VERBODEFAULT ADJETIVON NOMBREDEFAULT

ID	**PATRON**	**VALORACIÓN**	**LONGITUD**
1	ENTIDADDEFAULT ADJETIVON	3	2
2	ENTIDADDEFAULT ADVERBIOP ADJETIVON	1	3
3	ENTIDADDEFAULT ADJETIVOP	7	2
4	ENTIDADDEFAULT VERBON ADJETIVOP	3	3
5	ENTIDADDEFAULT ADJETIVON NOMBREDEFAULT	3	3

Se empezaría a clasificar por los patrones de longitud 3 (mayor longitud). Dado que el patrón es clasificado por el patrón semántico de id 5 (longitud 3) no es necesario seguir clasificando por los patrones de longitud menor (en este caso longitud 2 e inferiores). El resultado obtenido de esta fase sería que al patrón propuesto es clasificado únicamente por el patrón semántico identificado por el número 5.

3.3.3 Cálculo del sentimiento

Definidos los patrones semánticos que clasifican al patrón nuestro texto origen, es preciso determinar la polaridad final que se adjudicará al texto. Para ello se crearán tres métodos alternativos de cálculo de sentimiento que se detallan a continuación.

Dado el texto *'La empresa <Tópico> es buena y ofrece productos excelentes'*, el módulo de Generador de EtiquetaSentimiento produce la siguiente salida

Tabla 20. Salida de módulo EtiquetaSentimiento

TEXTO	PATRON
La	DEFAULTDEFAULT
empresa	NOMBREDEFAULT
[Tópico]	ENTIDADDEFAULT
es	VERBODEFAULT
buena	ADJETIVOP
y	DEFAULTDEFAULT
ofrece	VERBOP
productos	NOMBREDEFAULT
excelentes	ADJETIVOMP

Contando en la base de datos con los siguientes patrones:

Tabla 21. Ejemplo de patrones semánticos

ID	PATRON	VALORACIÓN	LONGITUD
2	ENTIDADDEFAULT ADJETIVOP VERBOP	7,5	3
3	ENTIDADDEFAULT VERBOP ADJETIVOMP	9	3
4	ENTIDADDEFAULT VERBON ADJETIVOP	3	3
5	ENTIDADDEFAULT ADJETIVON NOMBREDEFAULT	3	3

Tras clasificar el patrón generado del texto por el clasificador de patrones semánticos, se obtiene como resultado que los patrones de id 2 y 3 lo han clasificado con una valoración de 7,5 y 9 respectivamente.

3.3.3.1 Modo de la media aritmética

Como su propio nombre deja entrever, este método radica en asignar la valoración al texto por medio de la media aritmética de las valoraciones de los patrones semánticos que clasifican al patrón texto.

Tal y como se explicó en epígrafes anteriores, además de la polaridad del texto (positiva, neutra, negativa), seremos capaces de obtener la intensidad (grado de pertenencia a la polaridad: muy positiva, positiva, negativa, muy negativa, neutra).

Tabla 22. Correspondencia entre valores y polaridad

SALIDA	POLARIDAD
[0,5)	Negativa
5	Neutra
(5,10]	Positiva

Tabla 23. Cálculos realizados en el ejemplo

CÁLCULO REALIZADO	VALORACIÓN	POLARIDAD
(7,5 + 9) / 2	8,25	Positiva

3.1.1.2 Modo de conteo

Este modo intenta simular el método fundamentando en que la polaridad de un texto se calcula por medio del conteo de palabras de sentimiento que este contenga. En nuestro caso, en vez de realizar el conteo por palabras de sentimiento, se hará a través de un balance entre el número de patrones semánticos con valoración positiva y los patrones semánticos con valoración negativa.

Tabla 24. Correspondencia entre valores y polaridad

SALIDA	POLARIDAD
x <= 1	Negativa
0	Neutra
1 >= x	Positiva

Tabla 25. Cálculos realizados en el ejemplo

CÁLCULO REALIZADO	VALORACIÓN	POLARIDAD	INTENSIDAD
1 + 1	2	Positiva	Muy Positiva

3.1.1.3 Media ponderada en función de la máxima separación intermedia

En este método se pretende ponderar cada patrón semántico clasificado en función de su relevancia dentro del patrón del texto; que queda definida por la calidad del mismo.

La calidad de un patrón semántico dependerá de la máxima separación intermedia existente entre el patrón del texto y el patrón semántico, dado que la influencia que ejercen las palabras que componen un patrón semántico es mayor cuanto menor sea la separación intermedia entre ellas.

La separación intermedia sólo será llevada a cabo a través de aquellas categorías que no están formadas por una categoría DEFAULT, es decir, si en medio del patrón del texto tenemos estructuras formadas por DEFAULT, estas no serán contabilizadas en la separación intermedia.

Tabla 26. Funcionamiento de media ponderada

PATRON DEL TEXTO		**SEPARACIÓN INTERMEDIA**	
EtiquetaPatron	*Sentimiento*	*Id = 2*	*Id = 3*
DEFAULT	DEFAULT		
NOMBRE	DEFAULT		
ENTIDAD	DEFAULT	Inicio patron	Inicio patron
VERBO	DEFAULT	1	1
ADJETIVO	P	Reinicio cuenta	1
DEFAULT	DEFAULT	Descartado	Descartado
VERBO	P	Fin patron	Reinicio cuenta
NOMBRE	DEFAULT		1
ADJETIVO	MP		Fin patron
Separación máxima intermedia		1	3

Tabla 27. Cálculos realizados en el ejemplo

CÁLCULO REALIZADO	**VALORACIÓN**	**POLARIDAD**
((4-1)*7,5 + (4-3)*9)/4	7,785	Positiva

$$Valoracion_{texto} = \frac{\sum x_i w_i}{\sum w_i}\ , w_i = \left(\sum_{j=1}^{n} sep_j\right) - sep_i, \qquad sep = separacion\ maxima\ intermedia$$

* Si sep = 0, se aplica la media aritmética

4. Evaluación del sistema

Para evaluar el modelo propuesto, se ha realizado un análisis de cada uno de los modos para calcular el sentimiento en función del conjunto de entrenamiento y el conjunto de prueba. De este modo, de los 1416 elementos totales que disponíamos para evaluar el sistema, 900 conformarán el conjunto de entrenamiento y los 516 restantes el de prueba.

Como quedó explicado en epígrafes anteriores, el conjunto de entrenamiento nos servirá de referencia a la hora de elaborar el diccionario de sentimiento y los patrones semánticos. En contraposición, el conjunto de prueba será una muestra totalmente independiente de la conformada por el conjunto de entrenamiento. La existencia del conjunto de prueba permitirá obtener los resultados para la validación del sistema desarrollado.

En la siguiente tabla se muestra el resultado obtenido para cada una de los conjuntos por cada uno de los modos anteriormente explicados.

Tabla 28. Tasa de acierto

CONJUNTO	**ENTRENAMIENTO** (900)		**PRUEBA** (516)	
MÉTODO	**FALLOS**	**% ACIERTO**	**FALLOS**	**% ACIERTO**
Media	184	**79.55**	124	**75.96**
Conteo	208	76.88	137	73.44
Media ponderada	212	76.44	131	74.6

De los tres métodos propuestos, se puede dilucidar que el que ha proporcionado una mejor tasa de acierto ha sido el método de la media aritmética.

5. Implementación

5.1 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 *leer información* de dicha red. En la conclusión del epígrafe se mostrará la solución adoptada de entre las planteadas.

5.1.1 Conceptos básicos

- **Tweets**
  Los tweets son las unidades de información que se emiten a través de Twitter. Estas unidades poseen una singular característica que actúa de limitación; su longitud no puede ser superior a 140 caracteres.

- **Retweets**
  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 visualizarían el mismo en su timeline.

- **Menciones**
  Este concepto hace alusión a la funcionalidad que permite citar a un usuario en un mensaje. Para ello, es necesario usar el símbolo @ seguido del nombre del usuario.

- **Respuesta**
  Un tweet escrito por un usuario de Twitter puede ser rebatido por otro miembro de la comunidad. De esta forma el 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 queda definido por la cadena de texto que sigue al símbolo #. Su fama recae en la utilidad a la hora de agrupar conversaciones en torno a un mismo tema.

- **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 inverso 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 de tiempo 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.

5.1.2 Alternativas para la lectura de Twitter

Twitter proporciona múltiples APIs para facilitar el acceso a sus datos. De todas las existentes, para nuestro objetivo fundamental (extraer tweets) nos encontramos con dos que cumplen con el requisito inicial.

- REST API
- API STREAMING

Cada una de ellas posee una serie de características y limitaciones. Seguidamente se presentará una síntesis de las mismas.

5.1.2.1 REST API

Proporciona gran cantidad de interfaces que engloban las distintas funcionalidades que ofrece Twitter. Entre estas interfaces se encuentran las siguientes: Timeline, Tweets, *Search*, Streaming, Direct Messages, Friends & Followers, Users, Suggested Users, Favorites, Lists, Saved Searches, Places & Geo, Trends, Spam Reporting, OAuth, Help.

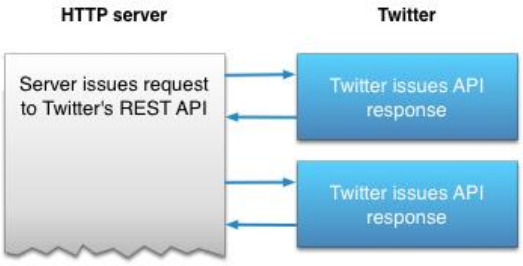


Ilustración 6. Funcionamiento REST API

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 la 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 geocodeparameter 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-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 occured 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 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 de la tipología de la clave usada (por usuario o por aplicación) existirá un número máximo de peticiones en ventanas temporales de 15 minutos (180,450 respectivamente)

5.1.2.2 API STREAMING

El conjunto de APIs Streaming que proporciona Twitter posibilita el acceso de baja latencia al Stream global de Tweets.

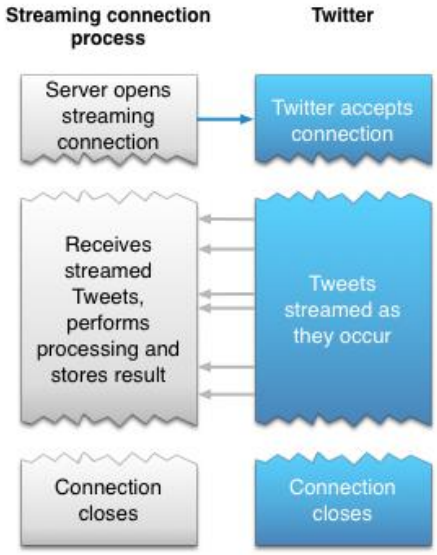


Ilustración 7. Funcionamiento de API Streaming

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

- Public Stream
- User Stream
- Site Stream

Para la elaboración de este trabajo, el Stream que a priori parece más adecuado sería el de usuarios. En la documentación oficial de Twitter se recoge la definición del mismo: permite el acceso al Stream de los datos públicos que fluyen a través de Twitter. Se recomienda su uso para seguir usuarios o temas específicos, así como para *minería de datos*.

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.

5.1.3 Solución adoptada

Teniendo en cuenta el análisis realizado 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 formar sencilla los servicios de Twitter en aplicaciones Java.

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

Tabla 29. Términos de búsqueda para Twitter

ENTIDAD	TAG
Telefónica	telefónica, telefonica, timofonica, timofónica, vomistar, movistar
Vodafone	vodafone, potafone, mierdaphone, mierdafone
Ono	ono, ono internet, ono telefono, ono línea, ono linea
Jazztel	Pazztel, jaztell, jaztel, jazztel

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 (especificación más detallada y exhaustiva en el anexo técnico adjunto a la memoria).

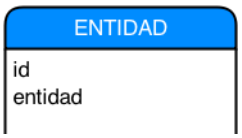
5.2.1 Lectura de Twitter

Los elementos básicos del proceso de lectura de Twitter (tweets) quedarán almacenados en una tabla TWEET.

Para cada uno de los elementos que componen la tabla se almacenarán una serie de atributos que resultan de especial interés para llevar a cabo el análisis de sentimiento. Estos son: tweet_id, fecha de publicación, usuario…

$$TWEET \begin{cases} \text{id} \\ \text{tweet} \\ usuario \\ fecha \\ es\_retweet \\ num\_retweet \\ valorado \end{cases}$$

Cada uno de los tópicos de los que se busca información se encuentran recogidos en una tabla ENTIDAD, que tendrá un identificador y un nombre que hace referencia al tópico.



$$ENTIDAD \begin{cases} \text{id} \\ entidad \end{cases}$$

Existe la posibilidad en la que al buscar información acerca de un sujeto existan distintos términos de búsqueda para los que se puedan encontrar información referente al mismo. Estos términos de búsqueda estarán almacenados en la tabla TAG.

$$TAG \begin{cases} \text{id} \\ tag \end{cases}$$

A continuación se expondrá un ejemplo con el objetivo de esclarecer la relación existente entre ENTIDAD y TAG.

> Imagínese que quisiéramos buscar información referente a Cristiano Ronaldo para realizar un estudio de su presencia en las redes sociales. En un principio parece válido buscar por Cristiano Ronaldo, sin embargo sería una búsqueda insuficiente debido a que existen infinidad de términos que se usan para hacerle referencia y que debieran ser incluidos en el estudio de cara a ofrecer los resultados veraces (CR7, CR9...).
>
> En el ejemplo anterior, la ENTIDAD (sujeto de búsqueda) sería Cristiano Ronaldo y sus TAG (términos de búsqueda) serían Cristiano Ronaldo, CR7, CR9, etc.

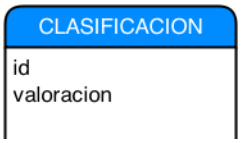
En este punto tendríamos un diseño *parcial* del modelo capaz de almacenar todos los TWEET extraídos en función a unos TAG en referencia a alguna ENTIDAD.

5.2.2 Análisis Semántico

Para completar el modelo E/R se añadirán una serie de tablas que nos ayuden a realizar el análisis semántico y el estudio de los de los resultados obtenidos. Será necesario guardar cada uno de los patrones semánticos junto con la valoración relacionada a dicho patrón. Dicha información se encontrará en la tabla PATRON.

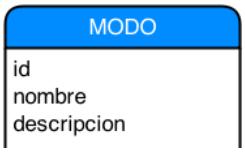
$$PATRON\begin{cases} \text{id} \\ \text{patron} \\ valoracion \end{cases}$$

Será necesario guardar cada una de las clasificaciones producidas por los distintos modos existentes para calcular la polaridad del texto. Estas se encontrarán en CLASIFICACION.



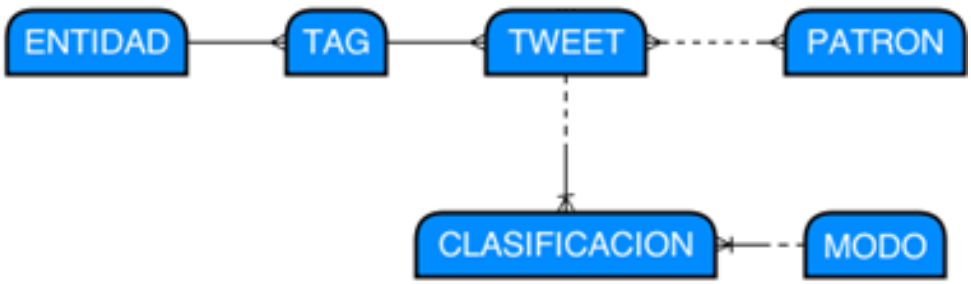
$$CLASIFICACION\begin{cases} id \\ \text{valoracion} \end{cases}$$

Para tener constancia de cada uno de los modos mediante los cuales se obtenga la polaridad del texto tendremos la tabla MODO.



$$MODO\begin{cases} \text{id} \\ \text{nombre} \\ observacion \end{cases}$$

Juntando los dos modelos parciales obtendremos el modelo E/R definitivo que usará nuestro sistema.

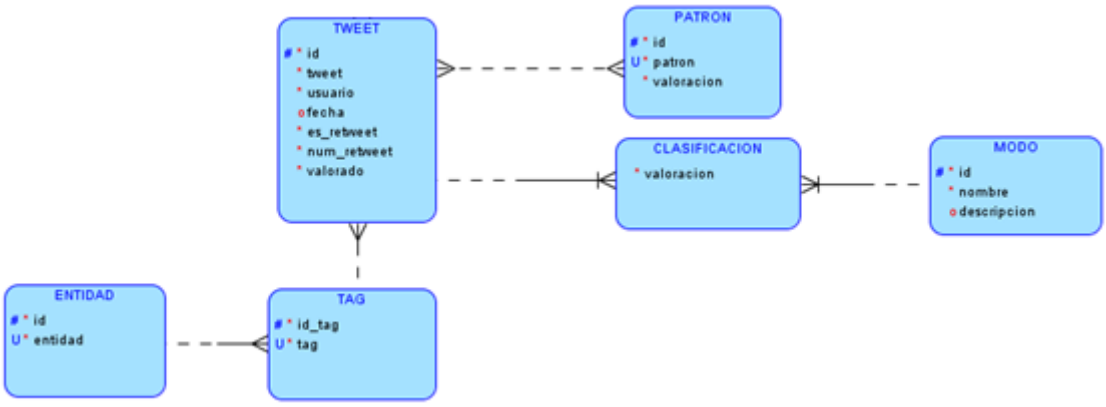


5.2.3 Elaboración de la base de datos

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:

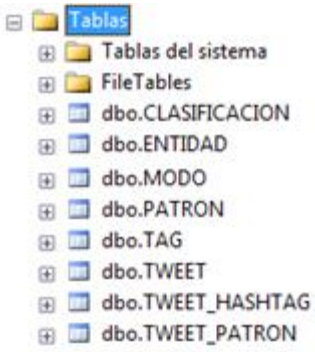
- Oracle SQL Developer Data Modeler
- Microsoft SQL Server 2008 R2

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.



5.3 Persistencia en la base de datos

Para poder manejar la persistencia de objetos desde java se hará uso de Hibernate. 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.

5.4 Interfaz

Mediante el diseño de la interfaz se pretende mostrar de una forma amigable los resultados obtenidos durante el proyecto. Para ello se crearán de tres pantallas principales, la primera de ellas será una introducción al proyecto, la segunda permitirá visualizar los datos de cada uno de los tópicos y la última se corresponderá con una descripción de los datos del autor.

A continuación se muestran los prototipos de cada una de las pantallas anteriormente citadas.

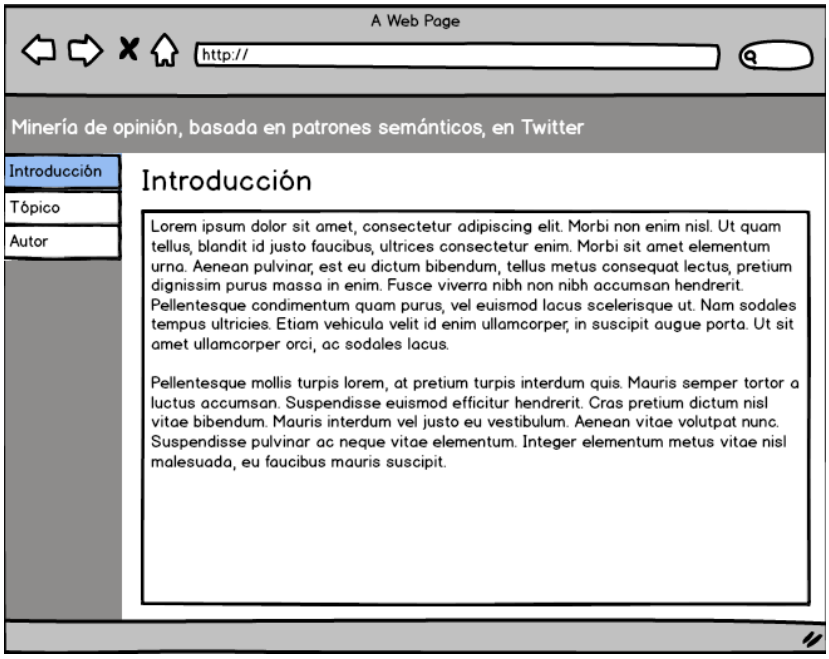


Ilustración 8. Prototipo pantalla introducción

En esta pantalla de introducción se pretende mostrar una presentación a la minería de opinión así como al sistema que se ha desarrollado en el proyecto.

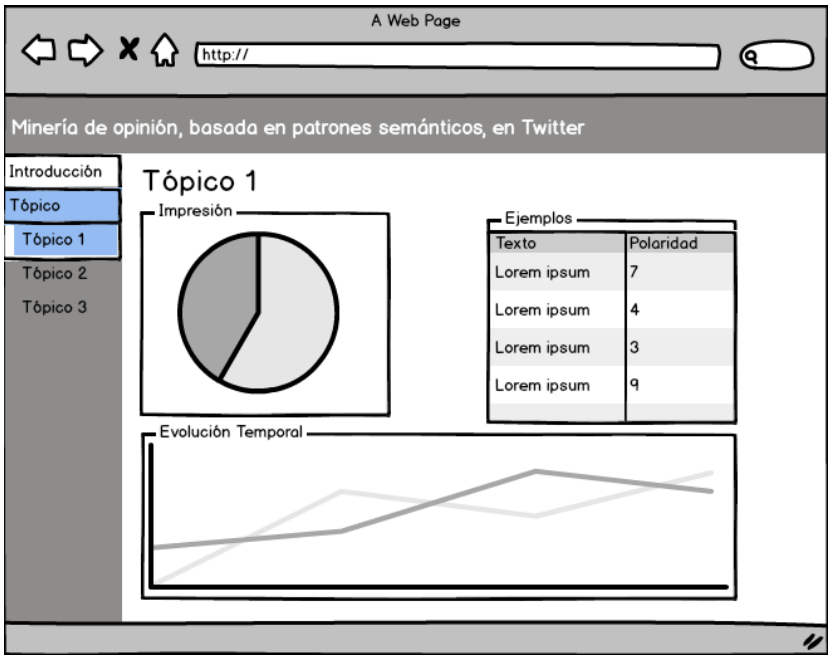


Ilustración 9. Prototipo pantalla tópico

En la siguiente pantalla (Tópico), será posible escoger entre los tópicos que se analizaron en el trabajo y visualizar una serie de resultados:

- Impresión: resume cuántos tweets positivos/negativos/neutros en el día.
- Ejemplos: expondrá una pequeña muestra de los textos y las polaridades asociadas del día.
- Evolución temporal: mostrará la media de sentimiento a lo largo de una semana.

Ilustración 10. Prototipo pantalla autor

En esta última pantalla se mostrarán los datos concernientes al autor del proyecto y su tutor.

Para la realización de la interfaz se ha hecho uso de las siguientes herramientas/tecnologías:

- JSP & Servlet
- Apache Tomcat
- Boostrap Admin

El resultado final tras la implementación se muestra a continuación.

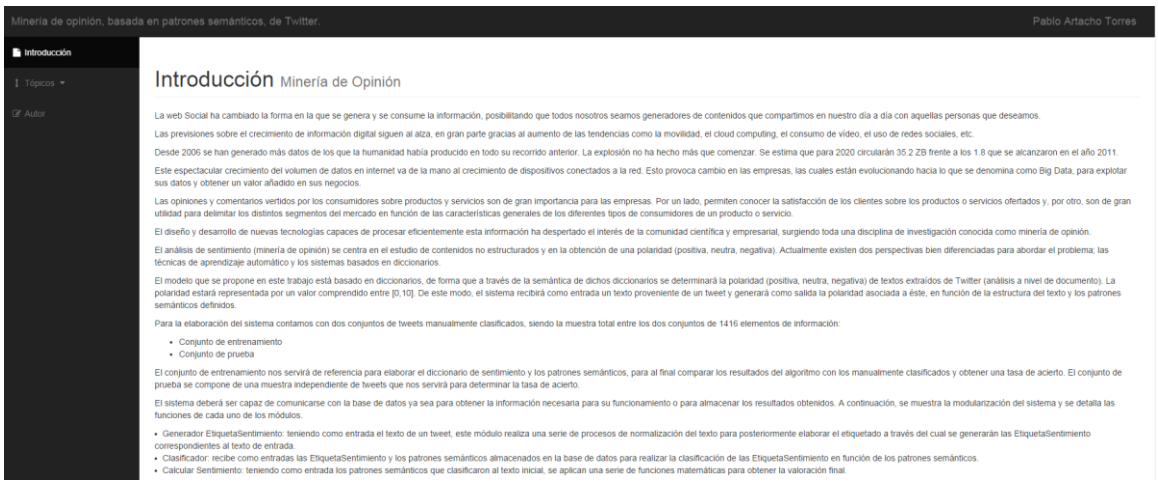


Ilustración 11. Pantalla introducción

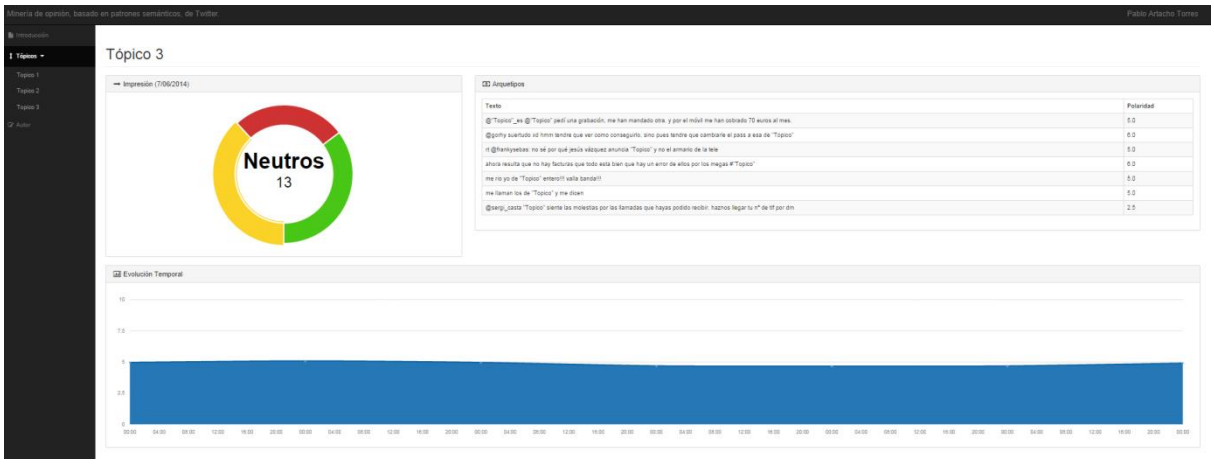


Ilustración 12. Pantalla tópico

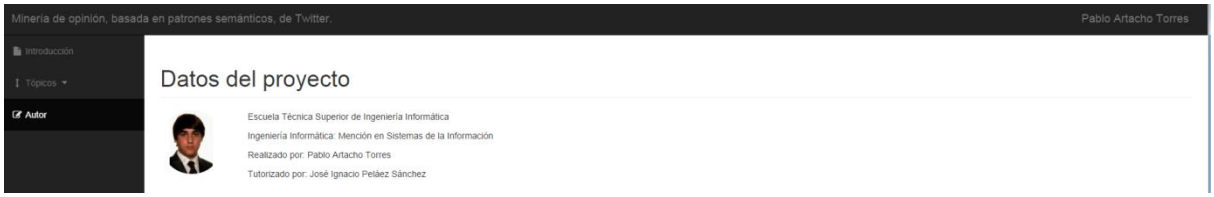


Ilustración 13. Pantalla autor

6. Conclusiones

En este trabajo se ha desarrollado un modelo para el análisis de sentimiento en la red social Twiter basado en diccionarios. El modelo propuesto se desarrolla mediante patrones semánticos, los cuales contienen la estructura de las expresiones que permiten obtener la polaridad de las mismas.

Para la validación del mismo, se han utilizado un conjunto de 1416 tweets, de los cuales 900 se han utilizado para el proceso de aprendizaje y 516 para la validación del sistema. La tasa de acierto ha sido de 75%.

A lo largo del trabajo se ha podido comprobar que el método realizado supone una buena aproximación a la hora de determinar la polaridad de los textos extraídos de Twitter. Comprobando los resultados obtenidos se puede vislumbrar que los fallos del sistema están provocados por verificar que mucho de los fallos del sistema son provocados bien por la falta de patrones semánticos y por la falta de las expresiones o en su mayoría, por la falta de expresiones propias de Twitter, o por la incapacidad de detectar determinadas además de expresiones mal escritas.

Como futuro trabajos, se pueden plantear los siguientes: Como propuestas de mejora se podrían señalar las siguientes:

Aumentar los patrones semánticos, mediante la incorporación de expertos en lenguaje. s

- Incrementar el número de palabras del diccionario
- Crear modelos híbridos con otros modelos.

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

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	**TWEET**
Descripción	En la tabla TWEET se encontrará almacenado cada unidad de información leída de Twitter (tweet).
Atributos	ATR–01: id ATR–02: tweet ATR–03: usuario ATR-04: fecha ATR-05: es_retweet ATR-06: num_retweet ATR-07: valorado
ATR – 01	**TWEET :: id**
Descripción	Identificador único de cada tweet. (autoasignado)
Tipo	BigInt
Comentarios	Clave primaria
ATR – 02	**TWEET :: tweet**
Descripción	Texto que contiene el tweet.
Tipo	Varchar
Comentarios	Obligatorio
ATR – 03	**TWEET :: usuario**
Descripción	Nombre del usuario que emite el tweet
Tipo	Varchar
Comentarios	Obligatorio
ATR – 04	**TWEET :: fecha**
Descripción	Hora y fecha de la emisión del tweet
Tipo	Date
Comentarios	Optativo
ATR – 05	**TWEET :: es_retweet**

Descripción	Indica si el tweet es un retweet
Tipo	Boolean
Comentarios	Obligatorio Valor inicial: Falso
ATR – 06	**TWEET :: num_retweet**
Descripción	Número de retweets.
Tipo	Integer
Comentarios	Obligatorio Valor Inicial: 0
ATR – 07	**TWEET :: valorado**
Descripción	Especifica si el tweet ha sido valorado o no
Tipo	Boolean
Comentarios	Obligatorio Valor inicial: Falso

ENT – 03	**ENTIDAD**
Descripción	Bajo el concepto de ENTIDAD almacenaremos cualquier empresa, persona u organismo, del que bien nos interesa almacenar información, o bien publica información de una entidad que sí nos interesa.
Atributos	ATR–10: id ATR–11: entidad
Comentarios	Ninguno
ATR – 10	**ENTIDAD :: id**
Descripción	Identificador para cada una de las entidades
Tipo	Integer
Comentarios	Clave primaria
ATR – 11	**ENTIDAD :: entidad**
Descripción	Nombre de la entidad
Tipo	Varchar
Comentarios	Obligatorio Único

ENT – 04	**TAG**
Descripción	Son cada una de las expresiones alternativas/sinónimos que podemos encontrar en las distintas fuentes de información para referirnos a una misma ENTIDAD.
Atributos	ATR–12: id_tag ATR–13: tag
Comentarios	Ninguno

ATR – 12	**TAG :: id_tag**
Descripción	Identificador para cada una de los tag
Tipo	Integer
Comentarios	Clave primaria
ATR – 13	**TAG :: tag**
Descripción	Nombre del tag
Tipo	Varchar
Comentarios	Obligatorio Único

ENT – 05	**PATRON**
Descripción	Cada uno de los patrones semánticos usados en la clasificación
Atributos	ATR–14: id ATR–15: patron ATR-16: valoracion
Comentarios	Ninguno
ATR – 14	**PATRON :: id**
Descripción	Identificador para cada una de los patrones
Tipo	Integer
Comentarios	Clave primaria
ATR – 15	**PATRON :: patron**

Descripción	Representación del patrón
Tipo	Varchar
Comentarios	Obligatorio Único
ATR – 16	**PATRON :: valoracion**
Descripción	Valoración asignada al patrón [0,10]
Tipo	Decimal
Comentarios	Obligatorio

ENT – 06	**CLASIFICACION**
Descripción	Cada uno de las clasificaciones
Atributos	ATR–17: valoracion
Comentarios	Ninguno
ATR – 17	**CLASIFICACION :: valoracion**
Descripción	Valoración para la clasificación
Tipo	Decimal
Comentarios	Obligatorio

ENT – 07	**MODO**
Descripción	Cada uno de las clasificaciones
Atributos	ATR–18: id ATR-19: nombre ATR-20: descripcion
Comentarios	Ninguno
ATR – 18	**MODO :: id**
Descripción	Valoración para la clasificación
Tipo	SmallInt
Comentarios	Clave Primaria
ATR – 19	**MODO :: modo**
Descripción	Nombre que identifica al modo

Tipo	Varchar
Comentarios	Obligatorio
ATR – 20	**MODO :: descripcion**
Descripción	Valoración para la clasificación
Tipo	Varchar
Comentarios	Optativo

2. Descripción de relaciones entre entidades

REL – 01	**Origen**	ENT–03	**Mult**	**Destino**	ENT–04	**Mult**
definida_sobre	ENTIDAD		1	TAG		*
Calificador	tiene			pertenece		
Descripción	Una entidad tiene cero o más tags asociados. Un tag pertenece a una única entidad.					
Optatividad	SI			NO		

REL – 02	**Origen**	ENT–04	**Mult**	**Destino**	ENT–01	**Mult**
definida_sobre	TAG		1	TWEET		*
Calificador	tiene			es de		
Descripción	Un tag tiene cero o más tweets. Un tweet tiene proviene de un único tag.					
Optatividad	SI			NO		

REL – 04	**Origen**	ENT–01	**Mult**	**Destino**	ENT–05	**Mult**
definida_sobre	TWEET		*	PATRON		*
Calificador	es clasificado			clasifica		
Descripción	Un tweet es clasificado por cero o más patrones. Un patrón clasifica a cero o más tweets.					
Atributos	distancia_maxima: Integer discrepancias: Integer					
Optatividad	SI			SI		
Comentarios	Ninguno					

REL – 05	**Origen**	ENT–01	**Mult**	**Destino**	ENT–07	**Mult**
definida_sobre	TWEET		*	MODO		*
Calificador	es clasificado por			clasifica		
Descripción	Un tweet se clasifica de cero o más modos. Un modo puede clasificar a cero o más tweets.					
Atributos	Valoración					
Optatividad	SI			SI		
Comentarios	La tabla intermedia deberá llamarse CLASIFICACION					