ISSN 1870-4069

Minería de opiniones sobre textos en Twitter

Belém Priego Sánchez, Luis Poot Terán

Universidad Autónoma Metropolitana unidad Azcapotzalco,
Departamento de Sistemas,
Ciudad de México, México
abps@azc.uam.mx,lept28@hotmail.com

Resumen. Actualmente, la información se encuentra en constante expansión gracias a la posibilidad que ha proporcionado el Internet y las herramientas de comunicación que éste tiene. Con lo cual, es posible obtener desde textos científicos hasta opiniones que una persona tiene sobre cierto producto. Sin embargo, conocer e interpretar toda esta información, principalmente los comentarios, no es tarea sencilla por lo que se hace necesaria la construcción de herramientas que permitan beneficiar y ayudar con dicha tarea. En este artículo se presenta una aplicación de Internet enriquecida que permite procesar comentarios de Twitter; el objetivo es analizar y clasificar los textos, escritos, emitidos por los usuarios de un producto. Dicha herramienta trabaja en tiempo real y permite clasificar opiniones a partir de una frase de búsqueda y mediante la obtención de la acción de ésta, es decir, mediante la identificación del verbo.

Palabras clave: minería de opiniones, clasificación de opiniones, identificación de información.

Text Opinion Mining in Twitter

Abstract. Nowadays, information is continuously growing in internet due to the great communication tools that already exist in this worlwide platform. We are actually able to obtain any kind of documents from Internet which ranges from scientific documents to single opinions about a certain product. The semantic interpretation of such documents is, however, a very difficult task which requires to foster the construction of natural language processing tools. In this paper, we present a enriched application able to process Tweets with the aim to analyze and clasify text excerpts such as customer opinions of some product. The tool developed is working real-time and allows to classify customer opinions from a query constructed on the basis of a part-of-speech analysis of the input tweet identifying two components: verbs and nouns.

Keywords: opinion mining, classification, natural language processing.

1. Introducción

Día a día la cantidad de información generada por los usuarios de Internet está en constate crecimiento, ésta va desde textos científicos hasta opiniones proporcionadas por una persona sobre un producto. Por lo que su análisis puede dar resultados útiles para afrontar problemas actuales. Sin embargo, obtener estos resultados no es tarea sencilla; en consecuencia, se han desarrollado nuevas tecnologías y herramientas.

El análisis de texto surge del Procesamiento del Lenguaje Natural. El cual "pretendía que una computadora interpretara cualquier texto con el fin de encontrar la semántica asociada al conjunto de palabras que lo conforman" [2]. A lo largo de los años se ha utilizado el análisis de texto para extraer patrones basados en palabras y temas significativos, obteniendo datos cuantitativos. Logrando por ejemplo predicciones sobre el comportamiento de un individuo, así también influyendo en su toma de decisiones de forma imperceptible.

Las redes sociales pueden brindar gran cantidad de información (comentarios que se generan a través del uso de las redes sociales) sobre gustos, opiniones, noticias, eventos, entre otros. Las personas comparten gran parte de sus actividades diarias en las redes sociales, la información, generada por este tipo de tecnologías, puede ser usada por dependencias, como empresas del sector privado para saber si un producto es del agrado de sus consumidores o un partido político para conocer si sus propuestas son aceptas por la mayoría de la población.

En el presente artículo se presenta un sistema web capaz de analizar opiniones en la red social Twitter. Para llevar acabo dicho análisis se deben tener en cuenta dos aspectos fundamentales: el tema a buscar y su valoración. En el primer aspecto se descargarán tweets con el objeto de búsqueda. El segundo aspecto examinará la información de cada tweet transformándolo por medio de lexicones (listas de palabras obtenidas del mismo tweet). Se analizarán los lexicones obtenidos para llegar a una respuesta porcentual sobre el objeto de búsqueda y su valoración. La herramienta presentada, en este artículo, es una aplicación de Internet enriquecida que permite procesar comentarios de Twitter con el fin de analizar y clasificar los textos escritos emitidos por los usuarios de un producto. Dicha herramienta trabaja en tiempo real y permite clasificar opiniones a partir de una frase de búsqueda y mediante la obtención de la acción de ésta, es decir, mediante la identificación del verbo.

2. Minería de opiniones

El análisis de sentimientos es una tarea de clasificación de textos dentro del área del procesamiento del lenguaje natural, cuyo objetivo consiste en detectar la polaridad (positiva, negativa o neutra), de una opinión dada por un cierto usuario [5]. El conocer la opinión que una persona tiene hacia un producto o servicio es de gran ayuda para toma de decisiones, ya que permite, entre otras cosas, que posibles consumidores verifiquen calidad del producto o servicio antes de utilizarlo.

El análisis de la polaridad, en cualquier tipo de comentario, es una tarea que está teniendo un gran auge, debido a que actualmente existe un fuerte interés en determinar automáticamente si las opiniones publicadas en medios públicos tienen un carácter positivo o negativo. La minería de opiniones se enfoca en determinar la polaridad de las publicaciones para, generalmente, dar seguimiento a la reputación de una entidad. En este artículo se aborda esta problemática mediante el desarrollo de una aplicación que permite procesar comentarios de Twitter emitidos de un producto; categorizando éstos en opiniones positivas y negativas.

2.1. Trabajos relacionados

En esta sección se presentan los trabajos reportados en la literatura que permiten, de cierto modo, realizar una comparativa entre lo que se presenta y lo existente. Las opiniones, fabricadas mediante una conversación o comentario, es posible consultarlas a través de foros, blogs o redes sociales, éstas últimas siendo la novedad y teniendo el mayor auge actual. Un trabajo basado en una colección de entradas de blogs, es el presentado en [1] que realiza el análisis de sentimientos y minería de opiniones en dichas entradas, mostrando la relevancia de los sistemas de aprendizaje automático como recurso para la detección de información de opinión. Siguendo con el auge actual, las redes sociales, se tiene el artículo [6] que proporciona información sobre la importancia y la usanza de la información publicada en redes sociales, así como una forma de hacerlo; esta importancia con el objetivo de realizar un análisis más detallado de la minería de opiniones y el análisis de sentimientos. Un trabajo asociado a la minería de opiniones y su alto grado de investigación en los últimos años es el presentado en [3] el cual plantea un sistema que extrae, procesa e identifica sentimientos para clasificar las opiniones sobre un dominio específico (hoteles).

En [4] se aborda el tema de minería de opiniones y análisis de sentimientos, lo cual permite identificar las opiniones de los usuarios expresando comentarios positivos, negativos o neutros y citas subyacentes al texto; se realiza la búsqueda de comentarios con opiniones propias de los usuarios con el fin de llegar a un resultado favorable o desfavorable del tema a buscar y, además, realizando una clasificación de sentimientos de los usuarios.

Hacer uso de los datos generados por personas, comentarios, es una gran oportunidad para ganar tiempo en las decisiones tomadas, debido a que proporcionan información que puede ser utilizada en diferentes ámbitos. A partir de los datos adquiridos se puede realizar un análisis automático y generar estadísticas sobre la opinión colectiva (positiva o negativa) de un producto, servicio o persona [5]. Dicho análisis es de gran utilidad para los analistas de medios, desde la disminución de tiempos hasta la disminución de costos. Por ello, en este artículo se ayuda a la automatización de ciertos procesos de análisis para empresas, consumidores, por citar algunos ejemplos. En las siguientes secciones se presenta el desarrollo del sistema que permite la minería de opiones de tweets.

3. Propuesta para la minería de tweets

La popularidad y el uso de las redes sociales, por personas de diversas edades y clases sociales, permiten que el desarrollo de este trabajo se lleve a cabo de una forma adecuada; debido a que actualmente las personas plasman su juicio sobre un producto o servicio de una manera sencilla y recurrente. Una de las formas en las que se transmite una opinión es mediante la escritura de un tweet y en ocasiones resulta ser más fácil escribir éste, de lo que se opina, que mandar un email o hacer una llamada. Por lo que recabar y analizar esta información es más sencillo para las empresas.

Esta plataforma puede tener alcances incluso de estudiar y predecir el comportamiento humano basado en lo que escribe, por ejemplo, conocer el estado anímico de un individuo para evitar atentados en escuelas como lo es en Estados Unidos de Norte América, evitar el suicidio de un adolescente, por citar algunos ejemplos.

Esta sección presenta la propuesta llevada a cabo para la minería de opiniones en tweets. La metodología está compuesta de cuatro módulos: extracción, preprocesado, clasificación y análisis y resultado, descritos a continuación.

3.1. Extracción de tweets

Este módulo es el encargado de recolectar los tweets, en tiempo real, que se analizarán; básicamente está compuesto de dos etapas principales: conexión y extracción.

La etapa de conexión consiste en realizar la conexión con los servidores de Twitter; para llevar a cabo dicho proceso, inicialmente se crea una cuenta de desarrollador de Twitter y posteriormente se hace el registro del proyecto. Cuando el proyecto es aceptado, Twitter proporciona cuatro llaves (*Consumer Key, Consumer secret, Acces token y Acces token secret*) únicas y son totalmente confidenciales; éstas permiten el acceso del proyecto a los servidores de Twitter. El desarrollo del proyecto fue realizado en Java, en la figura 1 se muestra el código de la clase que permite la conexión a los servidores de Twitter.

La segunda etapa es la extracción de los tweets, la cual prosigue a la etapa de conexión, que consiste básicamente de extraer los tweets que coinciden con el tema de búsqueda proporcionado por el usuario. A cada tweet se le da un formato específico y es almacenado en un archivo de texto plano; una de las condiciones de descarga, gratuita, es que únicamente se pueden descargar cien tweets por día. Por tal motivo, este conteo es reiniciado a las 24 horas de cada día. En la figura 2 se muestra la clase usada para extraer los tweets y darles formato.

3.2. Preprocesado del corpus

Este módulo está compuesto de tres etapas: la primera que se encarga de identificar los metadatos de los tweets, la segunda que le da un formato a los

```
public void Tweet() throws TwitterException{
      Twitter twitter;
       ConfigurationBuilder cb = new ConfigurationBuilder();
       cb.setDebugEnabled(true)
                 .setOAuthConsumerKey("Consumer Key")
                 .setOAuthConsumerSecret("Consumer Secret")
                 .setOAuthAccessToken("Access Token")
                 .setOAuthAccessTokenSecret("Access Token Secret");
       twitter = new TwitterFactory(cb.build()).getInstance();

   }
```

Fig. 1. Clase, en Java, que permite la conexión a Twitter.

tweets para distinguirlos unos de otros y la tercera que se encarga de procesar los tweets.

Al extraer los tweets, se observa que el corpus está compuesto de diferentes metadatos como son apodo (*nickname*) que el usuario elige y puede cambiar, seguido por un nombre de usuario (*username*) el cual es único e inicia con el carácter @; posteriormente, se encuentra la fecha del día que se escribió el tweet y se presenta en el formato *ddmesaaaa* y finalmente, el tweet escrito por el usuario.

Una vez identificados los metadatos de los tweets, el siguiente paso fue el etiquetamiento; éste consiste en que cada tweet fue etiquetado con las etiquetas $< INICIOTWEET >$ y $< /FINTWEET >$ que permiten delimitar el inicio y el fin del tweet. Posteriormente, se procede a identificar las palabras separadas por un espacio en blanco de cada oración en los tweets; este proceso se denomina Tokenización. El resultado de la tokenización es un nuevo corpus, el cual facilita la lematización de los tweets. El proceso de lematización se realizó con TreeTagger[1] con el archivo de parámetros o modelo del idioma español.

3.3. Clasificación del tweet basado en la valoración

Este módulo está compuesto de dos etapas, la primera es la identificación del verbo principal de la búsqueda y la segunda la clasificación de los tweets a partir de este verbo.

Para el caso de la etapa de identificación del verbo, se analizó la oración de búsqueda, que es ingresada por el usuario al sistema. El proceso es similar al de tokenización y lematización de los tweets, sin embargo, ahora el objetivo es identificar el verbo de la búsqueda ingresada. Esta identificación se realiza a partir de las etiquetas que el lematizador proporcionó y se selecciona el lema, que es lo que permite identificar la acción de búsqueda.

[1] Disponible en: http://www.cis.uni-muenchen.de/ schmid/tools/TreeTagger/

```java
public class TweetEx {
    private Twitter twitter;
    private SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");

    public TweetEx() {
        twitter = TwitterFactory.getSingleton();
    }

    public List<Status> query(String query) throws TwitterException {
        QueryResult search = twitter.search(new Query(query));
        List<Status> tweets = search.getTweets();
        return tweets;
    }

    public void escribirTweet(Status status) throws IOException {
        File A = new File("/Resultado/Busqueda.txt");
        FileWriter w = new FileWriter(A);
        BufferedWriter bw = new BufferedWriter(w);
        PrintWriter wr = new PrintWriter(bw);
        wr.write(String.format(status.getUser().getScreenName())+" "+sdf.format(status.getCreatedAt())+"\n");
        wr.write(status.getText()+"\n\n");
    }

    public void escribirTweet(List<Status> status) throws IOException {
        for (Status tweet : status) {
            escribirTweet(tweet);
        }
    }

}
```

Fig. 2. Clase, en Java, que permite la extracción de tweets.

La siguiente etapa consiste en clasificar los tweets de acuerdo al verbo principal de búsqueda, resultado obtenido en una etapa preliminar, mediante la trasformación de este en su antónimo. Para la clasificación de los tweets se realizó una búsqueda sobre el corpus de tweets lematizados, en la que se deseaba encontrar el verbo original o su antónimo, obteniendo dos corpara para cada una de las búsquedas efectuadas en Twitter. De esta manera, se crean los corpora de tweets donde en el primer grupo se encuentran los tweets que contienen la valoración original (verbo principal) denominado *Corpus_Positivo* y en el segundo los tweets que contienen su antónimo denominado *Corpus_Negativo*.

3.4. Análisis y visualización del resultado

En este módulo se realizó un conteo de los tweets que se encontraban en cada corpora (Corpus_Positivo y Corpus_Negativo). Posteriormente, se hizo la sustracción del total de la suma de los dos grupos con el total de tweets descargados, para poder verificar el porcentaje de error; es decir, cuántos tweets no se clasificaron. Esto se puede expresar con la ecuación error de clasificación (1).

$$Erc = Td - (Tp + Tn). \tag{1}$$

Donde Erc corresponde al resultado de los tweets erróneamente clasificados, Td el total de tweets descargados, Tp el número de tweets del Corpus_Positivo y Tn el número de tweets del Corpus_Negativo.

Finalmente, se visualizan los resultados en una interfaz gráfica, en la figura 3 se muestra la pestaña principal con un ejemplo de búsqueda que permite

visualizar de todo el proceso del sistema; desde la entrada de texto, donde el usuario escribe la oración de búsqueda, hasta la visualización de los resultados en una gráfica de pastel.

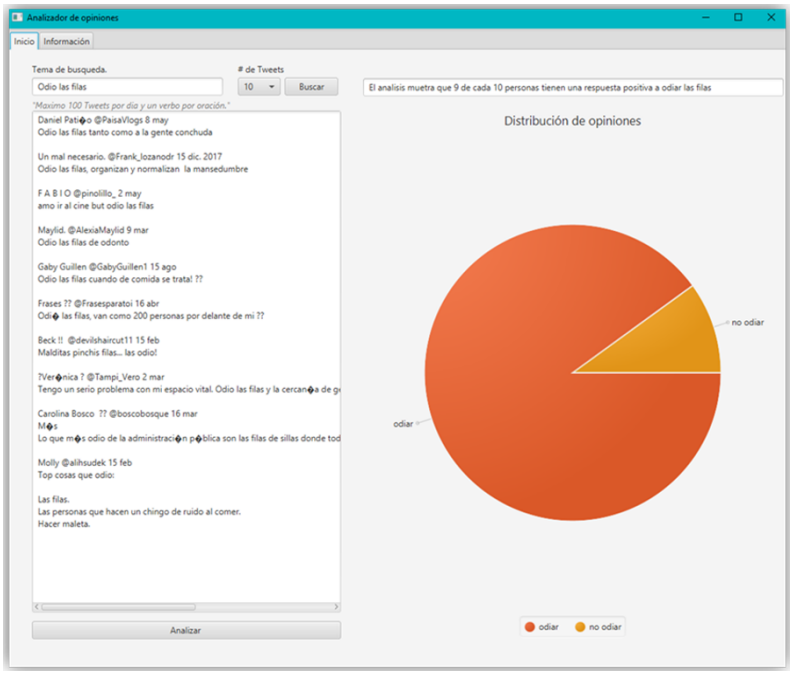

Fig. 3. Pestaña principal del sistema de minería de opiniones.

El sistema está construido de tal manera que si alguno de los datos solicitados no se incluye; por ejemplo, al momento de que un usuario no seleccione el número de tweets a buscar (10, 20, 40, 50, 80 o 100 tweets). Éste mostrará un mensaje al usuario indicando que algunos de los campos están vacíos. Además, se agregó una función en el módulo de clasificación, para evitar errores cuando se está buscando el verbo y no se encuentre. Es decir, cuando un usuario haya escrito una oración sin verbo.

4. Resultados obtenidos

En esta sección se muestran una serie de pruebas que permiten mostrar los resultados del sistema; gracias a que el sistema permite seleccionar el número de tweets (N) a descargar, es posible ejecutar la misma consulta con diferentes tamaños de corpora. En este sentido, los valores que se han corroborado son para N =10, 50, y 100 tweets, teniendo un total de 400 tweets diferentes de análisis y

descargados en diferentes tiempos, resaltando que el sistema funciona en tiempo real.

La primera prueba se consistió en ejecutar diez veces el sistema, en diferentes tiempos, con el tema de búsqueda "odiar las filas" para un tamaño de N=10, obteniendo un total de 100 tweets para dicho tema. La siguiente prueba consitió en buscar el tema de búsqueda "me gusta la navidad" para N=50 en dos ejecuciones diferentes, recolectando en total 100 tweets. El tercer experimento consistió en utilizar como oración de búsqueda "Quiero que México cambie" para N=100 en dos ejecuciones de sistema, analizando un total de 200 tweets en diferentes tiempos.

Como primeras pruebas del funcionamiento del sistema, se utilizaron alrededor de 1,000 tweets, sin embargo, para el análisis de dichas pruebas únicamente se utilizaron 400 tweets. De éstos se obtuvieron alrededor de 8,000 tokens. La Tabla 1, muestra los resultados de las diferentes pruebas realizadas.

Tabla 1. Resultados de las diferentes pruebas realizadas.

Tamaño del corpus	**No. de pruebas realizadas**	**No. de tokens lematizados**	**Porcentaje de tweets erróneamente clasificados**
10	10	2,000	5 %
50	2	2,000	15 %
100	2	4,000	27 %

Como puede observarse en la Tabla 1 el porcentaje de tweets erróneamente clasificados tiene una relación directa con el tamaño del corpus, es decir, si el tamaño del corpus aumenta o disminuye, el porcentaje de tweets erróneamente clasificados se ve afectado de la misma manera. Este comportamiento puede deberse a varios factores como lo son:

- El módulo de extracción de tweets los almacena en codificación ANSI, provocando que los caracteres no identificados se conviertan en signos de interrogación. Cuando se lematiza, TreeTagger no logra reconocerlos y puede provocar un tweet sin clasificar.
- Existen problemas de ortografía, en algunos casos, los usuarios no separan las palabras adecuadamente, provocando que en el proceso de lematización, el lema no sea encontrado y TreeTagger devuelva la etiqueta $< unknow >$.
- TreeTagger no reconoce la escritura de los tweets que no está hecha en español. Por lo que asigana la etiqueta $< unknow >$ (caso similar al anterior).

Finalmente, todas las pruebas se realizaron satisfactoriamente ya que siempre se cumplió con la función de analizar y clasificar los comentarios.

5. Conclusiones y perspectivas

El procesamiento de lenguaje natural es una área con un gran campo de acción, el presente trabajo es una muestra de lo que se puede realizar. Como se

pudo observar, se logró desarrollar, satisfactoriamente, el sistema para el análisis de opiniones en la red social Twitter, llegando a extraer tweets, lematizar tokens, clasificar y analizar corporas y finalmente, visualizar los resultados.

El sistema realiza el análisis adecuadamente, sin embargo, conforme aumenta el tamaño del corpus de tweets, los tweets erróneamente clasificados también lo hacen. El menor porcentaje de error en la clasificación de los tweets, se encontró en las pruebas realizadas con el corpus de tamaño N=10 tweets. Este porcentaje fue del 5 %, no obstante, el resultado no se puede considerar representativo por la muestra tan pequeña de opiniones. Se puede considerar guardar el resultado para búsquedas posteriores pero el ligar los resultados puede llevar a una conclusión errónea, ya que se pueden presentar variaciones de opiniones a través del tiempo.

La información del funcionamiento de TreeTagger en Java es muy escasa y poco descriptiva, lo que dificulta utilizar esta herramienta. Sin embargo, TreeTagger está completo con todos sus módulos, solo es necesario el ensayo y error para encontrar el modo adecuado de la aplicación deseada. Para mayor éxito en la lematización, se recomienda entrenar a TreeTagger o crear un archivo de parámetros con emoticones y abreviaturas y posiblemente realizar el almacenamiento en formato UNICODE.

El presente trabajo puede ser ampliado al análisis de emociones sobre textos en Twitter, polarizando los tokens en el proceso de lematización y reincorporándolos a su oración original. Esto crearía oraciones polarizadas y con su correcto análisis se podría conocer el estado de ánimo del usuario.

Referencias

1. Fernández, J., Boldrini, Gómez, E.J.M., Martínez-Barco, P.: Análisis de sentimientos y minería de opiniones: el corpus EmotiBlog. Procesamiento del Lenguaje Natural, Revista no. 47, pp. 179–187 (2011)
2. García Menier, E.: Análisis De Textos Por Computadora. Boletín de Lingüística 18(25), 121–134 (2006)
3. Henríquez, C.: Minería de Opiniones basado en la adaptación al español de ANEW sobre opiniones acerca de hoteles. Procesamiento de Lenguaje Natural, Revista no. 56, pp. 25–32 (2016)
4. Khairnar, J., Kinikar, M.: Machine Learning Algorithms for Opinion Mining and Sentiment Classification. IJSRP 3(6), 1–6 (2013)
5. Priego, B., Pinto, D., Castro, M., León, M.: Análisis de la polaridad en comentarios de estudiantes universitarios sobre el desempeño de sus profesores. Pistas Educativas, no. 130, pp. 946–961 (2018)
6. Sneka, G.: Algorithms for Opinion Mining and Sentiment Analysis: An Overview. International Journal of Advanced Research in Computer Science and Software Engineering 6(2), 1–5 (2016)