

Prediction of Tourist Traffic to Peru by using Sentiment Analysis in Twitter Social Network

Ricardo Linares
UNSA
Arequipa, Perú
rlinaresjuarez@gmail.com

José Herrera
UNSA
Arequipa, Perú
jherreraq@gmail.com

Ana Cuadros
UNSA
Arequipa, Perú
anamariacuadros@gmail.com

Luis Alfaro
UNSA
Arequipa, Perú
lalfaroc@gmail.com

Abstract—This work involves the use of tweets, from Twitter social network in which the users manifest the desire to travel to the country of Peru, to build a predictive tool of tourist traffic. To make this task has been made automated collection of tweets using web crawling and has been used Naive Bayes algorithm for sorting tweets as part of sentiment analysis. In the final part, we shown the results of the application of the tool for predicting the influx of tourists to Peru.

Index Terms—Prediction, Naive Bayes, Tourist Traffic, Sentiment Analysis, Twitter, Data Mining, Social Network.

I. INTRODUCCIÓN

Las oportunidades que brinda el análisis de sentimientos son enormes, se puede aplicar a diversos temas, por ejemplo[7]:

- Analizar o cuantificar el éxito de un partido político o una asociación.
- Conocer la opinión de los usuarios de cierta marca, producto o personaje público.
- Realizar predicciones de lo que puede ocurrir en el futuro a partir de los comentarios de los usuarios.
- Realizar estudios de tendencia de mercado.

Además, en los últimos años ha aumentado exponencialmente el uso de las redes sociales en todo el mundo. En tales redes sociales los usuarios vierten opiniones de cualquier tipo y sobre cualquier tema. Dentro del conjunto de las redes sociales las de *microblogging*, en concreto Twitter, no han parado de crecer, tanto en el número de usuarios como en el tráfico de datos. En España, en los últimos meses el uso de Twitter ha aumentado de forma acelerada y se puede apreciar que existe una gran cantidad de información que se genera en internet y no se debe dejar de aprovechar la oportunidad de procesarla[8].

Para realizar el trabajo propuesto, se ha realizado una serie de pasos que son los siguientes:

- 1) Elección de la red social, Twitter o Facebook.
- 2) Construcción del WebCrawler, el cual se encargará de recolectar los tweets durante cada hora en internet.
- 3) Limpieza de los tweets, para eliminar las etiquetas HTML, los acentos, los caracteres raros, cambiar a minúsculas todo.
- 4) Clasificación de los tweets, clasificándolos entre positivo o negativo.

5) Implementación del algoritmo de Naive Bayes, utilizando el lenguaje de programación Python.

6) Ejecución del clasificador y análisis de resultados.

La motivación del trabajo realizado es la de construir una herramienta que sea capaz de poder predecir la afluencia de turistas al Perú, y así poder alertar al sector turístico sobre la demanda de servicios que ocasionará dicha afluencia. Así muchas empresas gastronómicas, hoteles, ferias y todas aquellas empresas que prestan servicios al turista podrán planificar un servicio acorde a la demanda.

Las cifras del tráfico de turismo, están en aumento, ya son casi medio millón de personas que cruzan la frontera mensualmente.

II. ESTADO DEL ARTE

Sentitext es un software basado en conocimiento, y su funcionamiento se apoya en tres bases de datos. La principal de ellas es la de palabras (Words), que contiene más de 10.000 entradas. A cada palabra se le ha asignado una valencia que indica la carga afectiva de la misma; más específicamente, una palabra puede tener valencia -2, -1, 1 o 2, dependiendo de si su carga afectiva es muy negativa, negativa, positiva o muy positiva, respectivamente. Para la inserción de las palabras se ha recurrido a un diccionario de sinónimos con palabras semillas[1].

El Análisis de Sentimientos es una disciplina que recientemente ha originado interés en la comunidad científica generando algunos recursos como WordNet Affect, SentiWordNet, Micro-WNOP, estos dos últimos con relaciones semánticas con WordNet Emotion triggers por Balahur and Montoyo. Estos recursos contienen palabras sueltas cuya polaridad y emoción no son necesariamente aquellas anotadas dentro del recurso en un contexto más amplio[4].

Los trabajos de Análisis de Sentimientos sobre Twitter son mucho más recientes. Twitter aparece en el año 2006 y los primeros trabajos en este campo son del año 2009, cuando Twitter empieza a alcanzar popularidad. La competición SemEval-2013 también ha dedicado una tarea específica para el SA en Twitter, lo cual demuestra el notable interés

despertado en la comunidad científica en esta temática[5].

El objetivo del Análisis de Sentimientos es el de identificar y clasificar las opiniones expresadas en un texto. Existen dos grupos principales de aproximaciones que se pueden seguir: Aproximaciones basadas en léxicos (AS no supervisado) y las aproximaciones basadas en aprendizaje automático (AS supervisado). Las aproximaciones basadas en léxicos se centran en construir diccionarios de palabras etiquetadas. Este etiquetado asigna una puntuación para cada palabra y para cada polaridad, indicando cómo de estrecha es la relación entre esa palabra y esa polaridad. La manera más común de clasificar un texto utilizando estas puntuaciones es acumular los pesos de cada palabra, sumando los valores positivos y restando los negativos. Si la puntuación es positiva, el texto se clasifica como positivo, y si es negativa, se clasifica como negativo. Los diccionarios pueden crearse manualmente o automáticamente. Algunos ejemplos de léxicos son WordNet, SentiWordNet o JRC Tonicity. Sin embargo, es difícil recopilar y mantener un léxico universal, ya que una misma palabra en diferentes dominios puede expresar diferentes opiniones. [6].

III. ELECCIÓN DE LA RED SOCIAL

Un factor importante para el desarrollo del presente proyecto, ha sido la elección de la Red Social, se debía elegir entre Facebook o Twitter, las cuales son las redes sociales más importantes en la actualidad.

En Facebook, si utilizamos su buscador, podemos observar que sólo obtenemos personas, grupos, fan page, aplicaciones, etcétera, menos el contenido de las publicaciones de los Usuarios.

En cambio en Twitter, en su buscador obtenemos Tweets que son específicamente las publicaciones de los usuarios. Es por ello que utilizaremos la red social Twitter, por ser transparente para poder recolectar la información deseada. En Twitter como se puede apreciar en la figura 1, al realizar una búsqueda, nos devuelve los tweets, las publicaciones de los usuarios, justamente es la información que necesitamos.

IV. CONSTRUCCIÓN DEL CRAWLER

Elegida la red social, ahora debemos de recolectar la información y almacenarla en una base de datos.

A. Diseño de la Base de Datos

Se ha diseñado una simple base de datos, utilizando el software MySQL, en las que se almacenan las KeyWords, los tweets raws y los tweets limpios.

El diseño es una base de datos relacional, con las tablas requeridas para la recolección de tweets, el diseño puede



Fig. 1. Ejemplo de una búsqueda en Twitter.

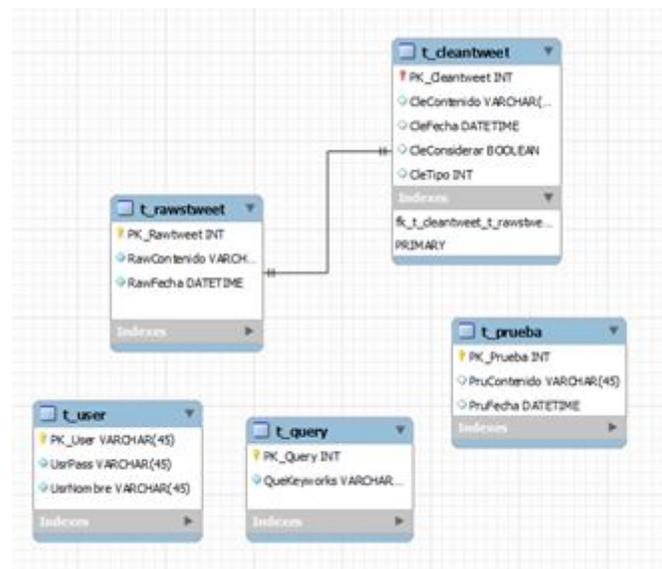


Fig. 2. Ejemplo de una búsqueda en Twitter.

verse en la figura 2.

Como se puede apreciar, se han creado 5 tablas, en la tabla "t_user" se almacena el usuario y contraseña del usuario que usará el crawler; en la tabla "t_query" se almacenan las palabras cables como "viaje a peru", "viajare a peru", "ire a peru"; en la tabla "t_prueba" se almacenan cada 5 minutos la hora y fecha, esto a modo de prueba para verificar que el crawler haya estado funcionando correctamente en cada hora; en la tabla "t_rawstweet" se almacenan los tweets tal y como se han extraído de Twitter mediante crawling y finalmente en "t_cleantweet" se almacenan los tweets normalizado luego de haber pasado por un proceso de filtrado de etiquetas

Fig. 3. 000Webhost.com - servicio de hosting gratuito.

html y otros elementos que no queremos que se almacenen. La tabla "t_rawstweet" y "t_cleantweet" están relacionadas puesto que por cada tweet raw (tal y como fue extraído) le corresponde un tweet normalizado (limpio de código html, tildes, números, etc.).

B. Configuración de Servidor Web

El crawler debe de ejecutarse a cada momento de forma automática, para ello este web crawler debe de estar alojado en un servidor web en internet, para tal fin se ha creado un sitio web en un hosting gratuito como lo es "000Webhost.com", en el cual hemos creado una cuenta, y se ha procedido a la creación del sitio, con un dominio gratuito: "www.mineria-apq.comyr.com". El servicio que ofrece "000Webhost.com" es completamente gratis, de todos los servidores gratuitos para alojar páginas web, este es el más completo, sólo muestra publicidad en sitios web alojados gratuitamente.

En este servidor gratuito se ha configurado e implementado la base de datos utilizando el sistema gestor de bases de datos MySQL, cuenta con una poderosa interfaz de administración y soporta las funciones avanzadas que requeriremos más adelante.

C. Codificación del WebCrawler en PHP

Para el proceso de WebCrawling, se ha utilizado una librería gratuita de WebCrawling, escrita en PHP, la librería se llama "Ultimate Web Scrapping" de propiedad de Barebones CMS, esta librería emula por completo un navegador, permite

Fig. 4. Ultimate Web Scraper - Barebones CMS.

recuperar contenido html, entre sus principales características se encuentran:

- Facilidad para emular varios manejadores WebBrowser.
- Extracción HTML de formularios y soporte para manipularlos.
- Completa emulación de cURL.
- Soporte de transferencias de archivos.
- Soporte diferentes protocolos: SSL, HTTP, HTTPS, CONNECT.
- Incluye un simple HTML DOM para fácilmente analizar y extraer el contenido deseado.
- Sigue cuidadosamente el IETF RFC estándar.

Con esta librería implementamos nuestro web crawler, sólo es necesario proporcionarle la URL del sitio a crawlear y el patrón de recuperación. Para ello debemos de analizar la estructura HTML de Twitter y luego de un profundo análisis determinamos que el patrón HTML que nos devuelve los Tweets solicitados es: "**p[class='js-tweet-text tweet-text']**".

Con este patrón nos devuelve cada uno de los tweets, pero no normalizados, con acentos, mayúsculas y todo lo demás.

Command to run:

Minute (s):

Hour (s):

Day(s):

Months(s):

Weekday(s):

List of current cron jobs:

Time	Command to run	Action
25 ****	php -f /home/a4852542/public_html/jobtwt7.php	<input type="button" value="X"/>
20 ****	php -f /home/a4852542/public_html/jobtwt6.php	<input type="button" value="X"/>
15 ****	php -f /home/a4852542/public_html/jobtwt5.php	<input type="button" value="X"/>
10 ****	php -f /home/a4852542/public_html/jobtwt4.php	<input type="button" value="X"/>
5 ****	php -f /home/a4852542/public_html/testjob.php	<input type="button" value="X"/>

Fig. 5. Configuración Cronjobs - Ejecución automática de Crawlers.

D. Configuración de Cronjobs

Para que este proceso sea automático se ha utilizado cronjobs, los cuales permiten realizar una búsqueda de tweets cada hora cronológica. En total se configurado 5 cronjobs, 4 para realizar el webcrawler a Twitter y 1 para testear el funcionamiento del servidor.

La restricción del servidor gratuito es que sólo permite configurar un máximo de 5 webcrawler. Y el tiempo mínimo de ejecución es de 5 minutos, pero para el proyecto se ha utilizado una configuración de una hora.

E. Recolección de Tweets

El crawler puesto en marcha funciona correctamente, y ha empezado a recolectar Tweets automáticamente en la base de datos, por un periodo de 15 días. Se puede apreciar una muestra de los tweets no normalizados en la figura 6.

Podemos notar que los tweets obtenidos contienen códigos HTML, tildes, caracteres raros, están en mayúscula o minúsculas, contienen enlaces, cifras. Todos estos elementos dificultan la clasificación de los tweets. Y con esta subsección ya tenemos nuestro crawler funcionando. Mediante este crawler hemos obtenido 1000 tweets, los cuales forman nuestra data para poder realizar el análisis de sentimientos con el algoritmo de Naive Bayes.

RawContenido

```
<p class="js-tweet-text tweet-text">Hoy <strong>en</strong> la tarde, <strong>estari</strong> <strong>en</strong> el Congreso anual de Supervisores y Líderes de Seguridad que c
minero más grande del <strong>PerÃ</strong>. </p>
<p class="js-tweet-text tweet-text"><strong>En</strong> breve <strong>estarÃ</st</strong>
<strong>en</strong> el programa Puntos de Vista de TV <strong>PerÃ</strong> 7.3
href="http://4.hidemiyass.com/ip-2/encoded/Oi8vdC5jby9pam5IVHZsejhr" class="twitter
data-pre-embedded="true" dir="ltr" >pic.twitter.com/ijneTvlz8Q</a></p>
<p class="js-tweet-text tweet-text"><a href="http://4.hidemiyass.com/ip-
2/encoded/czovL3R3aXR0ZXluY29tL2RhbnpwYWVlZyQ%3D%3D" class="twitter-atreply pr
dir="ltr" ><s>@</s><b>danzpaiva</b></a></a> Bueno, nunca me funciona. <strong>Esta
pendiente de que a las seis empiece, pero creo que <strong>en</strong> mi cable estã
de <strong>Peru</strong></p>
<p class="js-tweet-text tweet-text">Buenos días gente! Hoy <strong>estarÃ</strong>
<strong>en</strong> <a href="http://4.hidemiyass.com/ip-
2/encoded/czovL3R3aXR0ZXluY29tL2FtZXJpY2F0d9wZQ1" class="twitter-atreply pretty-
s>@</s><b>americatv_</strong>peru</strong></b></a> <strong>en</strong> el p
href="http://4.hidemiyass.com/ip-
2/encoded/czovL3R3aXR0ZXluY29tL0FMX0FJukvT2ZpY2lhbA%3D%3D" class="twitter-a
link" dir="ltr" ><s>@</s><b>AL_AIRE_Oficial</b></a> no se lo pierdan!</p>
```

Fig. 6. Ejemplo de tweets recolectados no normalizados.

V. LIMPIEZA DE TWEETS

Los Tweets son contenido mezclado con etiquetas HTML, emoticones, enlaces, etc. Para que el clasificador de Naive Bayes trabaje adecuadamente, es necesario normalizar cada uno de los Tweets.

Este proceso consiste en normalizar cada uno de los Tweets, para ello debemos de eliminar las etiquetas html, eliminar los enlaces, los emoticones, los enlaces propios de twitter que empiezan con numeral, cambiar todo a minúsculas, eliminar las cantidades, quitar los acentos.

Se ha sistematizado este requerimiento mediante un script en PHP, el cual se encarga de todo el proceso. Y almacena cada tweet normalizado en otra tabla aparte de la base de datos. El script convierte un tweet raw en un tweet normalizado de forma muy rápida y sin perder información crucial.

Podemos apreciar una muestra de los tweets normalizados en la figura 7.

VI. CLASIFICACIÓN DE TWEETS

Con los tweet normalizados, se ha procedido a clasificar manualmente los tweet, uno a uno hasta completar los 1000 tweets requeridos, donde un tweet positivo es aquel en el que se muestre un claro interés por viajar al Perú, o que ya está programado su viaje a Perú y un tweet negativo es aquel en el que no se muestra ningún tipo de interés de viajar al Perú, también se considera en este grupo a los tweets incoherentes, los que son spam, o no se encuentran dentro del contexto deseado.

Luego se ha creado un script en PHP, capaz de ubicar cada tweet en una de las dos carpetas, según le corresponde por su clasificación, guardándolos como un archivo plano en formato ".txt".

tweet	CleContenido
239	cinco dias durara el viaje del caracas a peru via patria podrida
240	viaje a peru ida y vuelta salida domingo de agosto mas informacion hasta
241	el viaje continua hoy por medio peru, rumbo a lima
242	por esta epoca del año empezabamos a preparar el viaje de los miticos , desde peru se os hecha de menos que no
243	no tengo ed fisica porque la profesora se fue de viaje a peru ojala que caiga el avion
244	con mama fuimos a la verduleria y el chabon nos conto de que con su mujer hicieron un viaje hasta peru en bicicleta,y que estuvieron 7 meses
245	nadie se salva de la situacion del pais viaje de mochileros tendra el para llegar a peru
246	bueno despues de ese viaje a peru recorriendo sus lindas playas, vuelvo a mi hermoso pais con el trabajo acumulado pero feliz

Fig. 7. Ejemplo de tweets normalizados.

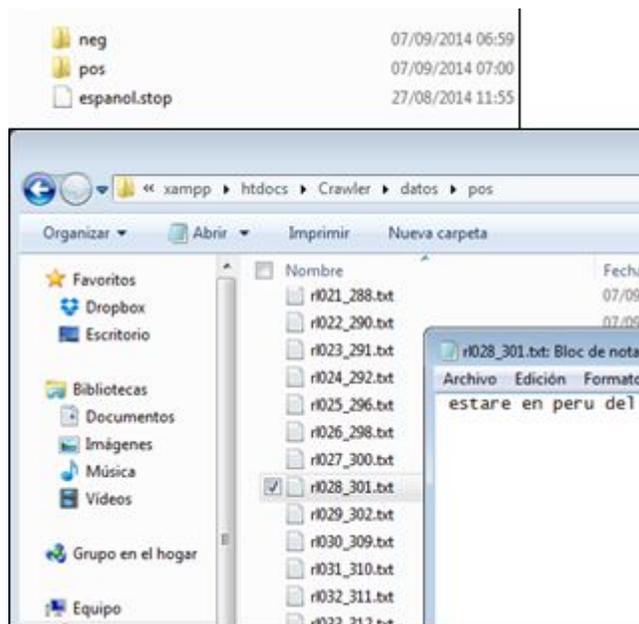


Fig. 8. Carpetas pos y neg contenedoras de los 1000 tweets.

Al finalizar se tienen dos carpetas, "pos" y "neg", conteniendo en cada una de ellas 500 tweets, con estos archivos ya tenemos los datos suficientes para poder ejecutar el algoritmo clasificador de Naive Bayes.

En la figura 8 se puede apreciar como se han creado cada uno de los tweets.

```

C:\xampp\htdocs\Crawler>G:\naiveBayes.py -f datos
[INFO] Performan 5-fold cross-validation on data set: datos
[INFO] Fold 0 Accuracy: 0.646465
[INFO] Fold 1 Accuracy: 0.570000
[INFO] Fold 2 Accuracy: 0.635000
[INFO] Fold 3 Accuracy: 0.630000
[INFO] Fold 4 Accuracy: 0.640000
[INFO] Accuracy: 0.634273
C:\xampp\htdocs\Crawler>

```

Fig. 9. Ejecucion del algoritmos Naive Bayes.

VII. IMPLEMENTACIÓN DE NAIVE BAYES

Para poder realizar el Análisis de Sentimientos, debemos de adaptar el algoritmo de clasificación de Naive Bayes, para que pueda determinar según probabilidad cuando una palabra es una opinión positiva o una opinión negativa.

Se ha implementado un script en Python, el cual lee los 1000 tweets y realiza el proceso de clasificación de cada uno de los tweets, donde se realiza una clasificación palabra a palabra contenida en cada tweet.

Para evitar clasificar las palabras comunes como los artículos, las preposiciones y las palabras comunes, se hace uso de un archivo "spanish.stop", el cual es un archivo de texto plano que contiene todas las palabras que el algoritmo debe de obviar en el momento de la clasificación. Por ejemplo: "ese, la, es, tu, yo".

VIII. ANÁLISIS DE RESULTADOS

Luego de haber recolectado 1000 tweets y haberles aplicado el script de limpieza, luego haberlos clasificado en tweet positivo (si viaja) o tweet negativo (si no viaja) y de haber ubicado cada uno de ellos en un archivo de texto plano, luego en su carpeta según corresponda (pos o neg), finalmente ejecutamos el algoritmo de Naive Bayes.

En una consola ejecutamos el programa en Python con nombre "NaiveBayes.py" y con los parámetros "-f" y "datos" (ubicación de las carpetas pos y neg que contienen los tweets almacenados en un archivo de texto plano cada uno de ellos).

Finalmente podemos ver los resultados arrojados por el programa en la figura 9. Como se puede apreciar se han ejecutado un total de 5 corridas del algoritmo, en la que se han analizado 100 tweets positivos y 100 tweets negativos en cada corrida.

La precisión obtenida en promedio a sido de 63.42 por ciento, siendo esta cantidad inferior a la deseada.

La precisión no es la deseada, mínimamente para que el modelo de predicción sea confiable debería de tener una precisión de 80 por ciento como mínimo ya que es la misma

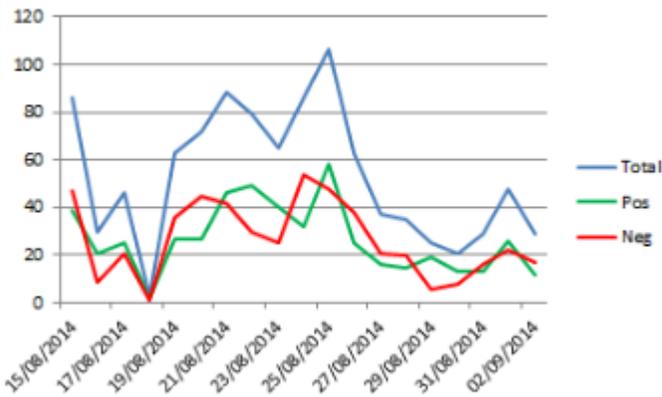


Fig. 10. Afluencia de tweets optimizada.

precisión de un clasificador humano.

Esto se debe a que el algoritmo clasificador de NaiveBayes funciona bien cuando se trata de textos u opiniones grandes, caso que no sucede con los tweets de twitter, se llega a esta conclusión debido al éxito en trabajos relacionados y documentados en la literatura. En este caso es mejor utilizar un Análisis de Sentimiento Basado en Léxico.

La negación es un elemento fundamental en el análisis de opiniones, que requiere un tratamiento especial, ya que una opinión negativa puede ser expresada con términos positivos negados o por el contrario, una opinión positiva puede expresarse a partir de la negación de términos negativos. Las oraciones "La película no me gustó" y "El personaje principal no era una mala persona" son claros ejemplos de los dos casos mencionados anteriormente[3].

Pero un gran aporte de este algoritmo se podemos apreciar en la figura 10, en la que podemos observar que el total de tweets se ha optimizado clasificándolo en cuáles son los positivos y cuáles son los negativos, se puede apreciar que la optimización influye bastante en el modelo de predicción.

Es aquí en donde se hace notable e importante la utilización de un algoritmo de análisis de sentimientos, puesto que los tweets en raw no demuestran un deseo u opinión en concreto, no sólo basta con cuantificarlos, sino que debemos de analizar cual es el sentimiento denotado en cada unidad. Así obtendremos un modelo predictivo mucho más confiable y acorde a la realidad.

IX. ANÁLISIS DE SENTIMIENTOS BASADO EN LÉXICO

Los primeros trabajos en el campo de análisis de sentimiento fueron desarrollados a partir de 2000. Los enfoques en esta tarea apelan tanto al paradigma simbólico de recursos léxicos y reglas de manipulación de símbolos al paradigma estadístico de aprendizaje automático[2].

Más recientemente, la tarea de análisis de sentimiento atrajo un mayor interés por soluciones más integradoras y

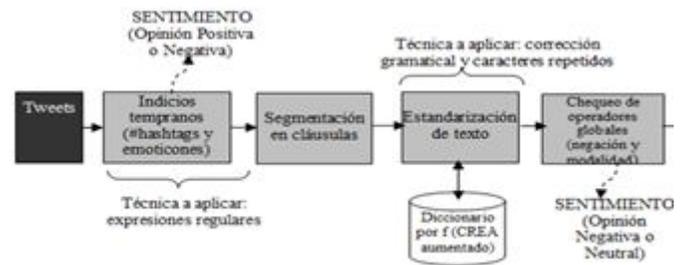


Fig. 11. Algoritmo Analizador Léxico - Primera parte.

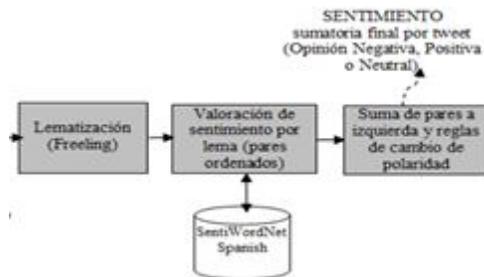


Fig. 12. Algoritmo Analizador Léxico - Segunda parte.

abarcativas. Los trabajos más exitosos recurrieron a recursos léxico-gramaticales bajo la forma de una ontología con valoración de sentimiento y diversas reglas de sintaxis básica. Enmarcado en el paradigma simbólico, Taboada Brooke, desarrolló el Semantic Orientation Calculator (SO-CAL), un conjunto de recursos léxicos con valoración de sentimiento que se ven afectados por intensificadores, índices de modalidad (negación y contrafáctica) y reglas de sintaxis básica. Esta solución integral para el inglés ha sido evaluada en distintos corpora de reseñas de dominio específico (hotelería, tecnología, películas, etc.) reportando medidas de precisión que varían entre 74% y 90%, resultados auspiciosos por demás.

Estos enfoques basados en recursos léxicos guiaron la investigación de SocialMetrix hacia un mecanismo de análisis de sentimiento para el español. Desafortunadamente, el obstáculo es la ausencia total de recursos léxicos para este idioma. Desde 2007, se encuentra disponible un poderoso recurso léxico para el inglés: SentiWordNet 2, una ontología con valoración de sentimiento en 3 dimensiones (positivo, negativo, neutral) y una amplia cobertura de acepciones. Definitivamente necesitábamos de algo similar para el español. Diversos trabajos en lenguas que no sean el inglés han apelado a enfoques que incluían etapas de traducción automática, pero sin lograr un éxito mayor.

Un algoritmo básico de un analizador basado en léxico es como se puede apreciar en las figuras 11 y 12.

X. TRABAJOS FUTUROS

Como trabajo futuro se encuentra la implementación de un analizador léxico, acorde al contexto de los viajes a Perú.

Realizar un análisis de sentimientos mediante el uso de Reglas y Análisis Léxico, implica crear un diccionario especializado de las palabras y frases que aparecen constantemente en la base de datos.

El análisis léxico es más preciso en cuanto a palabras o frases cortas, como los tweets, pero es menos eficiente cuando se trata de textos u opiniones más grandes.

La eficacia de usar este método, consiste en que tan preciso es el diccionario que armamos.

XI. CONCLUSIONES

Se ha considerado sólo los *tweets* en el idioma español, falta considerar los *tweets* redactados en otros idiomas como: Portugués, chino mandarín, inglés, etc. Ampliar la evaluación de mensajes cortos escritos en otros idiomas implica bastante esfuerzo y un análisis detallado de las restricciones que presenta cada idioma.

La mejor herramienta para realizar análisis de sentimientos en redes sociales es utilizar un analizador léxico y construir un buen diccionario. Esta herramienta a diferencia del clasificador propuesto es más preciso, debido a un análisis más profundo de cada término empleado gracias al diccionario con las definiciones correctas según el contexto.

En cuanto a aspectos técnicos, se recomienda el uso de servidores propios a fin de evitar limitaciones de servicios ofrecidos por *host* gratuitos de páginas web.

Los *tweets* son ambiguos, sarcásticos, incoherentes, emotivos, algunos son *spam*, en muchos casos están mal redactados, con graves faltas de ortografía, su naturaleza es la de ser mensajes cortos. Por ello para tener éxito en un análisis de sentimientos sobre las redes sociales, se debe de analizar profundamente el contexto y así construir un diccionario léxico que funcione eficientemente al momento de valorar el contenido del mensaje.

REFERENCES

- [1] A. Moreno, "Sentitext: sistema de análisis de sentimiento para el español," *Procesamiento de lenguaje natural.*, Vol. 45, pp. 297-298, Septiembre 2010.
- [2] F. Balbachan, and D. Dell'Era, "Análisis Automatizado de Sentimiento en Textos Breves de la Plataforma Twitter", *Socialmetrix*, June 2003.
- [3] E. Martínez, and M. Martin, *Desafíos del Análisis de Sentimiento*, Jaen: Sistemas Inteligentes de Acceso a la Información, 2007.
- [4] L. Dubian, "Análisis de Sentimientos sobre un Corpus en Español: Experimentación con un Caso de Estudio," *Argentine Symposium on Artificial Intelligence.*, 2013.
- [5] F. Pla, and L. Hurtado "Análisis de Sentimientos en Twitter," *Elirf-UPV.*, 2013.

- [6] J. Fernández, "Análisis de Sentimientos Multilingüe en la Web 2.0," *Actas V Jornadas TIMM.*, Vol. 45, pp. 19-21, January 2012.
- [7] C. Gonzalez, "Análisis de Sentimientos en Internet para Decisiones Estrategicas," *Icae.*, 2008.
- [8] E. Martinez, and M. Martin, "Análisis de Sentimientos," *Jaen.*, pp. 297-298, Septiembre 2010.