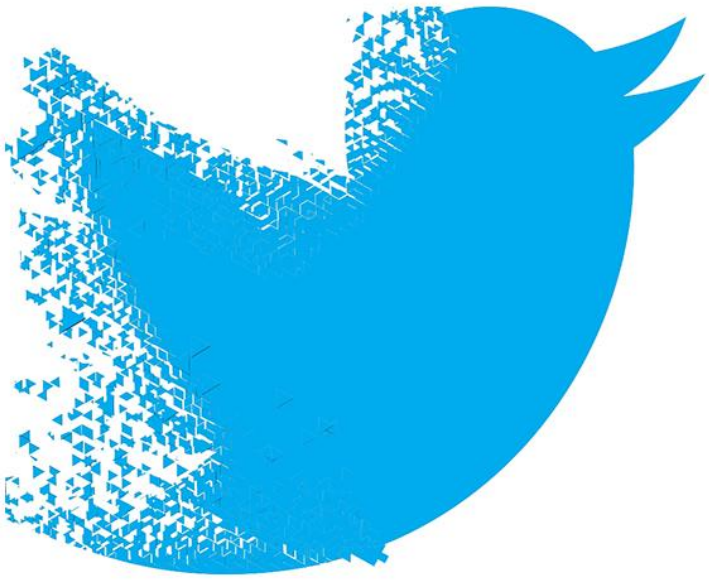


Recuperación de Información y Minería de Texto

Análisis de sentimientos en Twitter

Manuel Jesús Jiménez Navarro | 48121208X | jimeneznavarro@posgrado.uimp.es

Índice

Resumen

Twitter se trata de una red social en el que personas pueden compartir mensajes cortos de 280 caracteres. El valor de esta red social consiste en que es un buen escaparate de lo que ocurre en la sociedad en un momento dado. Esta red social ha sido objetivo de numerosos estudios en el campo de la sociología y psicología. Algunas de las aplicaciones son la predicción de crímenes, suicidios o incluso análisis de opiniones [1] [2] [3].

El análisis de sentimientos se trata de una técnica de minería de textos cuyo objetivo es clasificar opiniones y actitudes relacionadas con tópicos diferentes.

Resulta obvio que la inclusión de análisis de sentimientos en Twitter puede aportar múltiples beneficios. Por ello, en el presente documento se realizará un análisis de una muestra de tuits y posteriormente crear una serie de modelos con el objetivo de poder realizar la clasificación de estos tuits. Para ello, se ha creado un flujo de trabajo que, dado el conjunto de documentos provenientes de Twitter, realiza el análisis de los datos y obtiene un conjunto de modelos junto a sus correspondientes resultados.

Descripción de los datos

Los datos provienen de la plataforma Kaggle [4], una plataforma orientada al aprendizaje de tareas de minerías de datos. El conjunto de datos contiene 1600000 tuits extraídos de la API de Twitter entre el 6 de abril y 17 de junio de 2019. Los tuits obtenidos solo se encuentran en inglés. Las propiedades de los datos son:

- Id
- Fecha
- Flag que indica si se ha realizado alguna consulta sobre la API
- Autor del tuit
- Texto
- Sentimiento (positivo o negativo)

De todos los atributos, los únicos con los que se trabajarán a continuación serán el texto y el sentimiento.

El método con el que se etiquetaron los tuits fue a través de supervisión distante. EL proceso de supervisión distante etiqueta una serie de datos aplicando una serie de reglas sobre los datos no etiquetados. En el caso de los datos actuales, se usaron los emoticonos y una muestra pequeña de tuits manualmente etiquetados para este proceso. Toda la información se encuentra en su artículo [5].

En definitiva, el problema se enmarca dentro de clasificación binaria supervisada en el contexto de minería de textos.

Procesado de los datos

Para realizar el análisis correcto de los datos, es necesario realizar un procesamiento. Este procesamiento se realizará en varias fases:

- Eliminación de ruido.
- Tokenización.
- Eliminación de palabras vacías.
- Lemmatización.
- Eliminación de letras repetidas

Eliminación de ruido

La eliminación de ruido es una tarea específica de cada problema. Mediante esta tarea, se eliminan partes de los textos no deseadas (ruido). En el caso de los tuits, se definirá como ruido a los enlaces, nombres de otros usuarios de twitter, dígitos y cualquier cosa que no sea un conjunto de letras en mayúscula y minúscula.

Tokenización

La tokenización es el proceso en el que se pasa de trabajar a nivel de frase o texto, a trabajar a nivel de palabra. Para ello, lo que se realizará será separar el texto por espacios.

Eliminación de palabras vacías

Las palabras vacías son aquellas palabras que no aportan información acerca del sentimiento del texto y por lo tanto pueden ser obviadas. Para ello, se obtendrán las palabras vacías inglesas (stopwords) y se eliminarán aquellos tokens que estén dentro de este conjunto.

Lemmatización

La lematización consiste en el proceso de transformar un token a su raíz con el objetivo de eliminar aquellas palabras redundantes de los textos.

Eliminación de letras repetidas

Dada la naturaleza informal de la red social, en ocasiones la escritura de exagera repitiendo letras o conjuntos de letras. Por ejemplo, algunas de las palabras que pueden usarse y que transformarse son “hahahha”, “nooo”, “nono”, “looooolll”. Estas palabras y todas las combinaciones pueden transformarse para reducir el número de tokens y mejorar por lo tanto el análisis.

Resultados

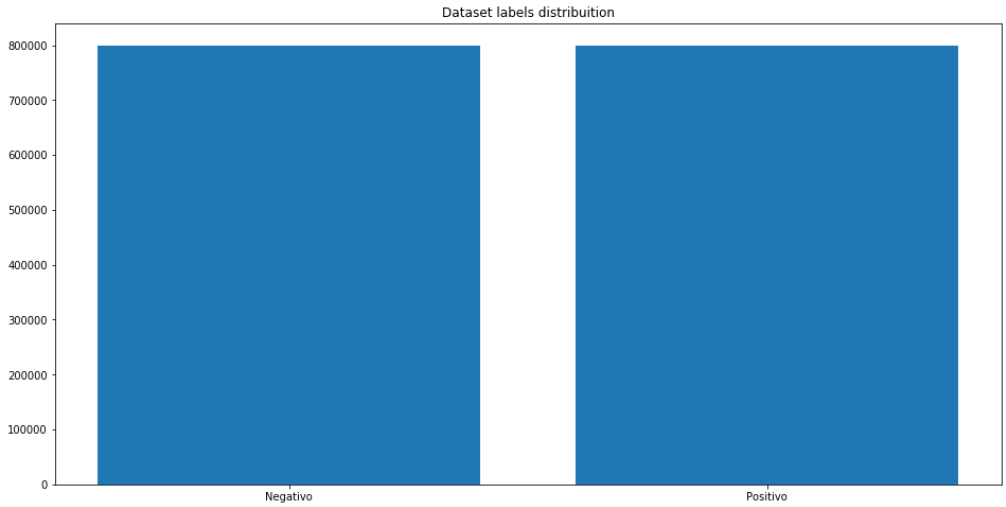
Tras la aplicación de este tratamiento algunos tuits se quedaron vacíos y fueron eliminados. A continuación, se mostrará los resultados antes y después del proceso de preprocesado.

	target	text
0	NEGATIVE	@switchfoot http://twitpic.com/2y1zl - Awww, t...
1	NEGATIVE	is upset that he can't update his Facebook by ...
2	NEGATIVE	@Kenichan I dived many times for the ball. Man...
3	NEGATIVE	my whole body feels itchy and like its on fire
4	NEGATIVE	@nationwideclass no, it's not behaving at all....

	target	text
0	Negativo	aww bummer shoulda got david carr third day
1	Negativo	upset updat facebook text might cri result sch...
2	Negativo	dive mani time ball manag save rest go bound
3	Negativo	whole bodi feel itchi like fire
4	Negativo	behav mad see

Análisis de los datos

En primer lugar, se mostrará la distribución de los datos respecto el sentimiento.



Como se observa los datos se encuentran prácticamente balanceados, los valores para los tuits negativos son 796128 y para los positivos son 795895.

Si se realiza un top de las palabras que más se repiten los resultados son los siguientes:

	word	count
0	go	130087
1	get	106139
2	day	102162
3	good	90234
4	work	83526
5	like	80334
6	love	78571
7	today	67313
8	time	64002
9	got	59950

Como se observa, las palabras más repetidas podrían considerarse positivas o neutras. Se analizarán las palabras clasificadas como positivas y negativas por separado mediante un wordmap para obtener una idea de las palabras que más se repiten en cada una.

Mediante esta visualización se puede observar como existen ciertas palabras que son más frecuentes en tuits negativos como: “sad”, “miss”, “bad”, etc. Por otro lado, hay palabras más frecuentes solo en el lado positivo: “fun”, “awesom”, “thank”, etc. Sin embargo, existen un número grande de palabras que son compartidas: “love”, “work”, “today”, etc. Esto indica que analizar las palabras de forma individual no es una buena forma de realizar el análisis de sentimiento. Una palabra que aparentemente puede ser positiva como “haha” puede cambiar totalmente de sentido dependiendo del contexto en el que se mueva.

Modelos de análisis de sentimiento

En este apartado se ejecutarán una serie de modelos con el objetivo de comparar las distintas aproximaciones. El proceso será incremental, desde un modelo más sencillo a uno más complejo como las redes neuronales profundas.

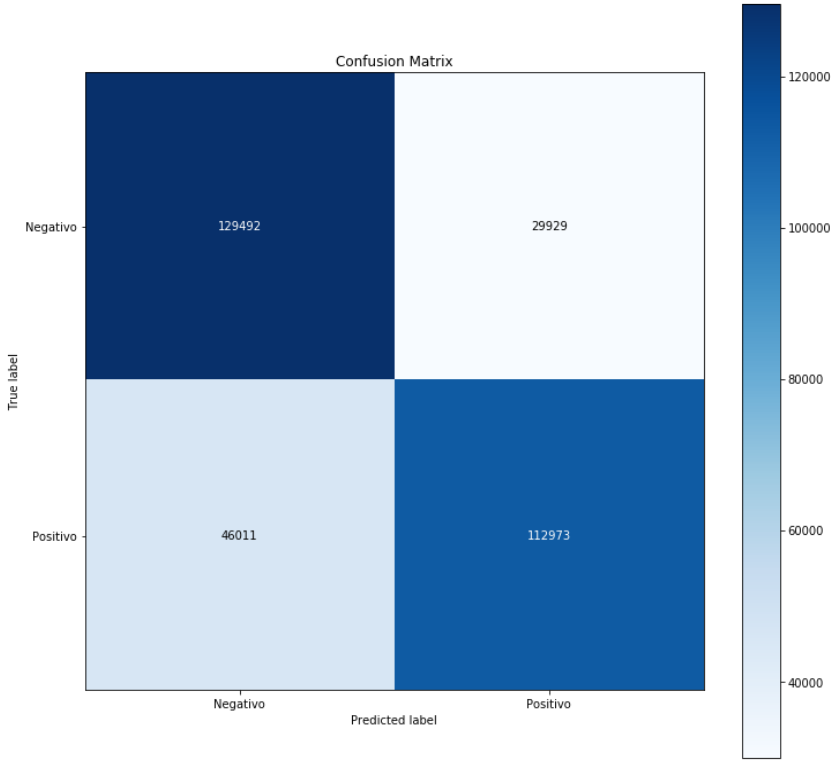
Naive bayes

Naive bayes de un modelo probabilístico de clasificación simple aplicable en numerosos ámbitos, entre ellos la minería de texto. Este modelo asume la independencia entre las variables predictoras sobre el objetivo. Se trata de un modelo clásico utilizado para la clasificación de texto y usado como baseline.

Tras la construcción del modelo con un conjunto de entrenamiento, se realizó una prueba sobre un conjunto independiente. Como resultado, se obtuvieron los siguientes resultados:

```
accuracy: 0.760261302429296
Most Informative Features
              tweeteradd = True           Positi : Negati =    444.9 : 1.0
                  hurtss = True           Negati : Positi =     56.3 : 1.0
                dividend = True           Positi : Negati =     46.4 : 1.0
                gratitud = True           Positi : Negati =     44.4 : 1.0
                   atcha = True           Positi : Negati =     44.4 : 1.0
                   sadd = True            Negati : Positi =     41.1 : 1.0
                   unlov = True           Negati : Positi =     34.6 : 1.0
                    ceci = True           Negati : Positi =     31.6 : 1.0
                  sadfac = True           Negati : Positi =     31.5 : 1.0
                  boohoo = True           Negati : Positi =     30.7 : 1.0
```

La matriz de confusión correspondiente es la siguiente:



La precisión es del 76% y se muestran una serie de palabras que mejor clasifican hacia positivo o negativo. Un ejemplo de palabra es “tweeteradd” cuyo ratio es 445 positivo a 1 negativo. Probablemente, esta palabra aparezca en los anuncios que twittter publica que normalmente poseen un tono positivo.

Estos resultados serán tomados como baseline en los próximos modelos.

Regresión logística

La regresión logística es un modelo estadístico usado para clasificación binaria en el que se predice la probabilidad de que un determinado ejemplo pertenezca a una clase u otra.

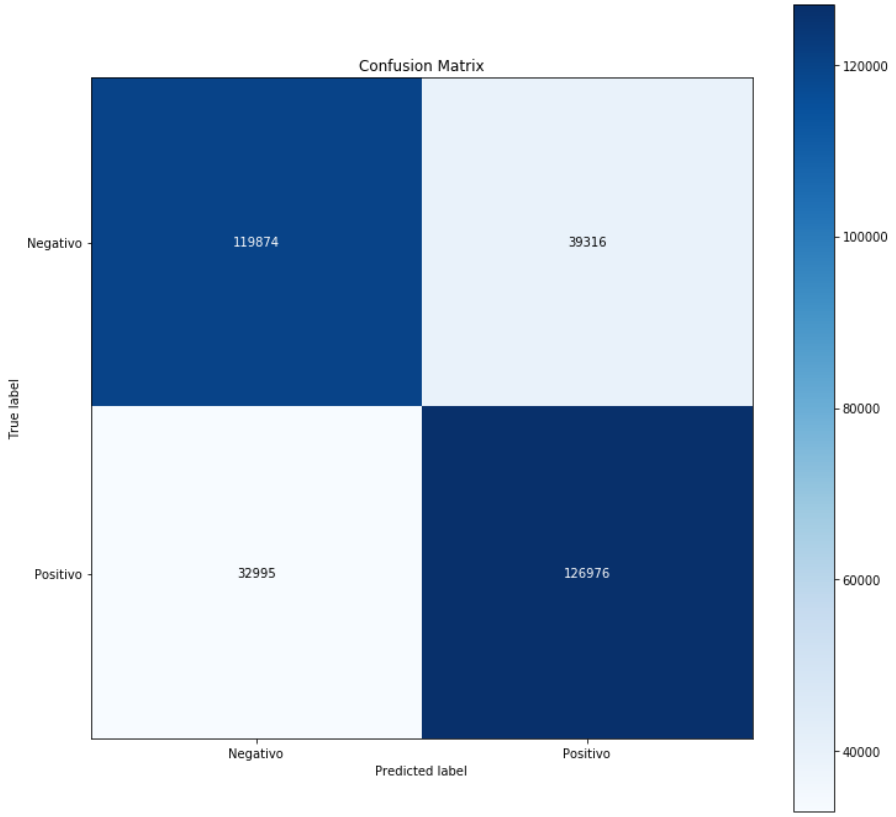
Para la aplicación de este modelo, es necesario realizar una transformación de los datos, esta transformación consistirá en convertir los tuits a vectores TFIDF. Para ello, el primer paso será obtener la frecuencia de cada elemento en cada documento, creando así una matriz n x m siendo n el número de tuits (filas) y m el número de tokens diferentes (columnas). Posteriormente se calcularán los índices TF e IDF y se calculará su valor de TFIDF por cada palabra. Los resultados obtenidos son los siguientes:

	tfidf
carr	0.496322
shoulda	0.440454
third	0.400221
bummer	0.371295
david	0.350437
...	...
gloveshack	0.000000
glovess	0.000000
glovesub	0.000000
glow	0.000000
zzuu	0.000000

Para el primer documento se muestra un vector de n elementos en los que los elementos que no se han usado reciben un valor de 0 dado que el índice de TF es 0. Se observa, que nombres propios como David reciben un valor bastante alto ya que no es un token que aparezca en muchos tuits probablemente.

Una vez realizada esta transformación, ya es posible realizar las predicciones. Los atributos consistirán en los índices TFIDF del tuit y se clasificará como positivo y

negativo. Como resultado se obtuvo una precisión del 77.34% y una matriz de confusión como se muestra a continuación:



Como se observa se ha mejorado en alrededor de un 1% la predicción de NaiveBayes. Se observa en la matriz de confusión que su mejora es respecto los verdaderos positivos empeorando los verdaderos negativos respecto a Naive Bayes.

Hay que tener en cuenta que el procesamiento es más costoso en este caso, ya que se necesita calcular los índices TFIDF para cada documento. En caso de encontrarse en un problema incremental, no resulta muy eficiente este método a diferencia del Naive Bayes ya que sería necesario reentrenar el modelo completo junto a los índices con cada nuevo tuit que llegara.

Redes neuronales

Las redes neuronales son un tipo de modelo en el que la información fluye a través de nodos conocidos como neuronas. Las neuronas se agrupan en capas con el objetivo de obtener características de los datos y realizar las predicciones.

Antes de hablar sobre el modelo, es necesario comentar un concepto relacionado con esta técnica, el Word embedding.

Word Embedding

El word embedding es una técnica usada en el procesamiento de lenguaje natural. Su principal objetivo es traspasar los tokens a un espacio vectorial multidimensional.

Las redes neuronales no son capaces de trabajar con datos que no sean de tipo numérico, por ello, es necesario transformar los datos para poder introducirlos dentro de la red. Por ejemplo, en el caso de un atributo nominal en el que se nombren calles, se podría transformar de forma numérica de forma que a cada calle distinta se le atribuye un valor numérico. Al trabajar con textos es necesaria realizar esta transformación, sin embargo, existe una diferencia con el caso de los atributos nominales. En un atributo nominal, no existe el concepto de distancia entre los atributos, pero en un token si existe un concepto de distancia o similitud entre ellos. Por ejemplo, el token “good” debería ser más cercano a “amazing” que al token “awful”.

Word embedding es una técnica en la que los tokens son proyectados a un espacio multidimensional en el que los vectores formados por los tokens mantengan distancias cercanas con tokens similares y distancias lejanas con los tokens no similares. Para ello, existen diferentes modelos como Word2Vec [6] o GloVe [7].

Modelo

El modelo de red neuronal que se usará para realizar la predicción será el modelo BERT [8] basado en la estructura de red neuronal llamada Transformer [9]. BERT usa su propio modelo para realizar el word embedding, aunque es posible usarlo con otros como los descritos en el apartado anterior. El objetivo será introducir el modelo más una capa adicional de salida y entrenarlo para realizar un análisis de sentimientos.

```
Train on 477606 samples, validate on 1114417 samples
477606/477606 [==============================] - 2238s 5ms/sample - loss: 0.4798 - acc: 0.7687 - val_loss: 0.4602 - val_acc: 0.7840
<tensorflow.python.keras.callbacks.History at 0x7fc7e8f59f60>
```

Como resultado se obtuvo una precisión del 78.4% en una sola época y entrenando con solo el 30% de los datos. A pesar de que el tiempo de ejecución es mucho mayor a los algoritmos vistos anteriormente, es posible mejorar los modelos anteriormente estudiados.

Conclusiones

Durante el desarrollo del documento se han analizado diferentes conceptos. Finalmente, se han obtenido las siguientes conclusiones:

- El preprocesamiento de los textos posee una parte mecánica, pero también es necesario atender a situaciones muy específicas (entorno).
- El preprocesamiento posee distintas técnicas con distintas propiedades que actualmente conforman un área de estudio.
- Modelos simples con un tiempo muy corto aportan resultados que dependiendo del contexto pueden ser buenos.
- Es necesario una gran cantidad de cómputo para obtener buenos modelos en problemas de minería de texto con modelos de redes neuronales.

Referencias

[1] - Mining Twitter data for crime trend prediction <https://www.researchgate.net/publication/323532548_Mining_Twitter_data_for_crime_trend_prediction>

[2] - Detecting suicidality on Twitter.<https://www.sciencedirect.com/science/article/pii/S2214782915000160>.

[3] - Twitter as a Corpus for Sentiment Analysis and Opinion Mining. <http://www.lrec-conf.org/proceedings/lrec2010/pdf/385_Paper.pdf>.

[4] Kaggle. <https://www.kaggle.com/>.

[5] Twitter Sentiment Classification using Distant Supervision.

<https://cs.stanford.edu/people/alecmgo/papers/TwitterDistantSupervision09.pdf>.

[6] Word2Vec Project. <https://code.google.com/archive/p/word2vec/>.

[7] GloVe Project. <https://nlp.stanford.edu/projects/glove/>.

[8] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding . <https://arxiv.org/abs/1810.04805>.

[9] Attention is all you need. <https://arxiv.org/abs/1706.03762>.