Asignatura Text Mining en Social Media
Máster Big Data Analytics

20 de junio de 2022

Equipo Kminos
Lorena Ponce Ruiz, Sergio Langarita Benítez, Luis Miguel Bartolin Arnau
lorena.ponru@gmail.com
sergiolangaritabenitez@gmail.com
luismibartolinarnau@gmail.com
Acceso al código en GitHub: Kminos-PAN-2022

Abstract

En este artículo se presenta la clasificación de tweets en dos estados, irónico o no irónico, de los tweets que conforman el dataset proporcionado. Para ello, se han utilizado diferentes modelos, donde un modelo Random Forest entrenado con 1.000 palabras ha obtenido los mejores resultados. Como el dataset dado contenía tweets de diferente índole con distintos caracteres, previamente se ha realizado un tratamiento de los datos antes de utilizarlo para el entrenamiento del modelo. Finalmente, tras testear las predicciones obtenidas en el dataset test se obtuvo, en la mejor predicción, un valor de *Accuracy* de 0,667.

1 Introducción

Las técnicas de Text Mining así como las herramientas del procesado del lenguaje natural (PLN) tienen un papel fundamental en campos como la analítica de sentimientos, clasificación de personas para estudios de población o para fines no tan éticos, como campañas políticas o como se aprecia en la actualidad, para intentar enfrentar a la población construyendo "Fake news" personalizadas a diferentes sectores de la población.

En este proyecto se estudian diferentes modelos de clasificación para un dataset de tweets en bruto con caracteres especiales, emoticonos y repetición de letras para hacer énfasis en la frase. Al tratarse de un dataset con tweets tal cual han sido escritos por los autores es necesario un primer tratamiento de los datos antes de entrenar los diferentes modelos que serán los encargados de clasificar si el tweet de cada autor es irónico o si se trata de un tweet que no muestra ningún tipo de ironía.

El resto del artículo se organiza como se indica a continuación: Sección II se explica el dataset de tweets dado. Sección III se presenta detalladamente la propuesta implementada para realizar una correcta predicción. En la Sección IV se presentan los resultados experimentales obtenidos tras el entrenamiento del modelo utilizado. Sección V se detallan las conclusiones obtenidas, así como los futuros trabajos que se podrían haber desarrollado.

2 Dataset

Para el entrenamiento del modelo de clasificación se ha proporcionado un dataset con 200 tweets en inglés, en formato xml, para un total de 420 autores distintos, haciendo un total de 84.000 tweets de entrenamiento. Previamente el dataset había sido preprocesado para anonimizar los nombres de los autores y otros elementos como los hashtags, las menciones a otros usuarios y las urls. Además se dispone de un archivo 'truth.txt', con formato csv, que incluye la etiqueta para cada uno de los autores indicando si es irónico o no. Por otro lado, se dispone de un dataset de testeo que contiene 200 tweets para 180 autores que vienen sin etiquetar.

Dado el dataset completo de entrenamiento, y ya que se desconoce el resultado del dataset de test, se dispone a dividirlo en dos partes: training, con el 70% de los datos, y test, con el 30% restante. De esta manera, se emplea el set de training para entrenar los diferentes modelos y el set de test para validar el modelo, calculando la precisión (*Accuracy*) en la predicción de si el autor es o no irónico. Para ello se ha empleado la función `parititon()` que permite balancear las particiones de acuerdo al valor de una columna, para que ambos factores de predicción estén equilibrados.

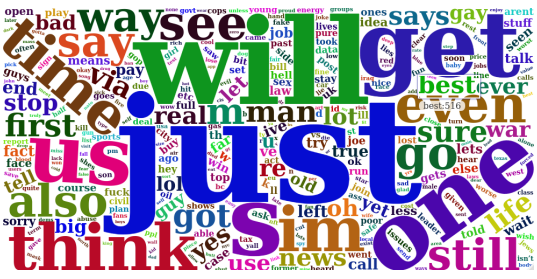
Previa a la fase de modelización se lleva a cabo una exploración del dataset completo. En primera instacia revisamos que el dataset esté equilibrado, apreciando que hay el mismo número de autores irónicos (N=210) como no irónicos (N=210), por lo que se descarta aplicar medidas de balanceo durante el preprocesamiento de los datos.

Seguidamente comprobamos el número medio de caracteres por tweet y autor. Esta medida es relevante dado que Twitter permite un máximo de 280 caracteres, por lo que *a priori*, podría considerarse que este factor puede ser relevante en la distinción entre irónicos y no irnónicos. Para ello, se calculó si la diferencia entre el número medio de caracteres non-stop; es decir, de palabras con significado (Tabla 1), para cada grupo, era significativo (**p.valor inferior a 0,001**). Esta significancia nos lleva a concluir que los irónicos de media, son más escuetos en el uso de palabras con significado.

Table 1: Descriptivos para el número de caracteres non-stop por clase

Clase	Media	Desv.típica
Irónicos	13,95	7,92
No irónicos	16,18	8,24
Total	15,06	8,15

(a) Palabras más frecuentes en el dataset train



(b) Palabras más frecuentes en el dataset test

Figure 1: Palabras representadas en base a su frecuencia en el dataset de train (a) y test (b).

Finalmente comparamos las palabras más comunes entre el dataset completo de train y el dataset de test. Extrayendo las 100 palabras más frecuentes, al menos un 92% de los caracteres eran coincidentes en ambos datasets. Este porcentaje disminuye a un 90,8% al aumentar el número de palabras a 1.000. A pesar de ello, gran parte de las palabras empleadas son coincidentes lo que parece indicar que, el empleo de la frecuencia de las palabras y aproximaciones derivadas pueden ser una buena aproximación para el entrenamiento de modelos.

Remarcar que previo a estas estadísticas descriptivas los datos se preprocesaron de acuerdo a la descripción detallada en la Sección III.

3 Propuesta del alumno

El primer paso, previo a la modelización, es la fase de preprocesado y curación de los datos. En esta primera fase se ha empleado la función predefinida `GenerateVocabulary()` que, con sus diferentes opciones, permite convertir todas las palabras a minúsculas y eliminar números, signos de puntuación, espacios en blanco y stop-words, lo cual se aplicó a ambos datasets.

Seguidamente, y tras particionar el training set en train y test respectivamente, se aplican las diferentes aproximaciones:

1. (M1) La primera de ellas fue la aplicación más sencilla, partiendo de un Bag of Words con la frecuencia relativa de cada una de las palabras del set de training.

2. (M2) En la siguiente aproximación se parte del Bag of Words anterior y se genera otro dataframe dibujando un Árbol de decisión y seleccionando las variables más relevantes, *Variable importance*. En cada iteración se muestran aproximadamente unas 12 palabras como más importantes, de las cuales unas 4 ó 5 poseen el peso de la partición del árbol de decisión.

3. (M3) La tercera aproximación fue generando un tercer dataframe con una matriz de correlación de las variables entre sí y seleccionando las variables con más correlación respecto a variable de predicción. El primer paso para ello es transformar el factor irónico

y no irónico a numérico y posteriormente calcular las correlaciones con la función de R `corr()`. Finlamente seleccionamos aquellas que tenían un peso absoluto superior a 0.2, observando máximos de 0.4.

4. (M4) Finalmente, se creó una matriz con la importancia relativa de cada una de las palabras, calulando el TF-IDF (*Term frequency - Inverse document frequency*), también conocido como bolsa de palabras ponderada.

Las tres primeras matrices o dataframes anteriores se aplicaron los modelos de Support Vector Machine (SVM), General Linear Model (GLM), Linear Regression (LR), Arbol de decisión (TREE) y Random Forest (RF).Todos los modelos aplicados se llevaron a cabo con el paquete de R `caret`. De forma excepcional se aplicó en el cuarto dataset una red neuronal.

Entre los diferentes parámetros de variabilidad, se modificaron el número de palabras (n) en el vocabulario, siendo 100 el mínimo, pasando por 1.000, 10.000 y subiendo a un máximo de 15.000. También se aumentaron el número de folds (k), con valores comprendidos entre 1, 3, 5 y 10, y repeticiones en el cross-validations (r), probando 1, 3 y 5 repeticiones.

4 Resultados experimentales

A continuación se muestra una tabla con los mejores resultados, de acuerdo al parámetro de *Accuracy*, de todos los modelos y aproximaciones empleadas:

Table 2: Métricas obtenidas tras la ejecución de cada modelo, donde se indica el modelo, los parámetros aplicados, los valores de *Accuracy* y Kappa para el set de train (negro) y el de test (azul).

Modelo	Parámetros	Accuracy	Kappa
RF-1	n=1.000;k=1;r=1;m=1	0,954/0,950	0,883/0,9
RF-2	n=1.000;k=10;r=3;m=1	0,931/0,976	0,862/0,952
SVM	n=1.000;k=10;r=3;m=1	0,879/0,905	0,758/0,843
TREE	n=1.000;k=10;r=1;m=1	0,871/0,841	0,743/0,683
GLM	n=1.000;k=10;r=3;m=2	0,827/0,825	0,654/0,651

En una primera fase del entrenamiento se obtuvo que el mejor modelo era Random Forest (Tabla 2; RF-1), con valores de validación de 0.95 en *Accuracy* y 0.9 en Kappa como se muestra en la Tabla 2. En una siguiente fase se obtuvieron valores similares, aunque con métricas más acortadas (Tabla 2; RF-2).

En base a los diferentes modelos aplicados se extrae que Random Forest es el modelo que presenta las mejores puntuaciones y el único modelo que obtenía valores de *Accuracy* superiores a 0,9. Seguidamente se encuentran los Árboles de Decisiones y Support Vector Machine, aunque estos dependían en gran medida del dataframe de entrenamiento, y finalmente General Linear Model que de media obtenía valores de *Accuracy* en torno a 0,5. Remarcar que en el caso de Random Forest, el efecto al variar los parámetros de cross-validation, al menos en las métricas de validación en el set de entrenamiento, variaban poco y de media se obtenían valores de *Accuracy* superiores a 0,87.

Por otra parte, en la tabla 2 no se muestran los resultados del modelo de regresión lineal, dado que el R^2 en ningún caso fue superior al 30% y disponía de valores de RMSE y MAE elevados, indicando, como se puede intuir ya que estamos ante un problema de clasificación y no de regresión, que no es buen modelo de entrenamiento.

En cuanto al empleo de números de palabras superiores a 1.000 los valores de *Accuracy* no mejoraban, incluso disminuían, y además se encarecía el tiempo de procesado.

A continuación se muestran los valores obtenidos tras procesar los resultados del test con la etiqueta de referencia:

Table 3: Métricas obtenidas en la validación del modelo.

Predicción	Parámetros	Accuracy test
V1	n=1.000;k=1;r=1;m=1	0.5833
V3	n=1.000;k=10;r=1;m=1	0.6389
V4	n=1.000;k=10;r=3;m=1	0.6667

A partir del primer modelo el 44% de los autores de la base de datos de test se clasificaron como irónicos y el 56% restante como no irónicos. En los siguientes modelos entregados se obtuvieron valores de *Accuracy* de 0,93 y 0,94, re-

spectivamente, en el entrenamiento. Todo ello nos indica que en el primer modelo la clasificación podría considerarse un evento debido al azar. Por otro lado, esos valores tan dispares entre entrenamiento y validación pueden estar indicando un efecto de overfitting, donde el modelo se ajusta a los datos de entrenamiento pero este no es capaz de predecir satisfactoriamente los datos del test. Otra de las posibles razones de esta desviación entre la precisión de entrenamiento y test puede deberse a un error el desarrollo del modelo o a la necesidad de aplicar otras estrategias como se comentarán en la Sección V.

5 Conclusiones y trabajo futuro

En base a lo anteriormente visto, podemos concluir de que sería necesario hacer una revisión del código empleado, dado que se han observado que empleando cross-validation se obtienen resultados bastante precisos y es posible que haya errores de ejecución. Por otra parte, dado que los mejores resultados se obtenían con 1.000 palabras los modelos finales se han entrenado con este parámetro, pero referente a la relación e importancia de las palabras con la clasificación entre irónicos y no irónicos puede ser interesante probar a testear en entorno real otras aproximaciones aplicadas a lo largo del proyecto, como el uso de las palabras más importantes obtenidas por Árboles de decisión o a partir de aplicar TF-IDF.

Con respecto al trabajo futuro hay un abanico de posibilidades en la manipulación del dataset, dando más peso a elementos como los emoticonos, hastags, las menciones a otros usuarios y seleccionando diferentes palabras claves. En este punto hacer hincapié que, como se observa en la Figura 1, hay grupos de palabras coincidentes que parecen tener gran relevancia, como serían los términos vinculados al tiempo (now, time, old, last, etc.), algunas expresiones verbales (will, like, can, etc.) o palabras claves como: women, gays, war, entre otras, que podrían requerir de un tratamiento especial o un examen más exhaustivo. Otra posibilidad sería, explotar el número de palabras con sentido como variable predictora.

Además, hay otras técnicas que no se han empleado que pueden resultar interesantes como la clusterización y observar la relevancia por grupo de palabras, el entrenamiento a partir de una reducción de la dimensionalidad o PCA, así como el empleo de n-gramas.

Comentarios

En el repositorio Git estará disponible el código para la obtención de los diferentes resultados y códigos de testeo de las aproximaciones y parámetros descritos en el presente documento.