Detección de usuarios propensos a la transmisión de desinformación o 'Fake News'

Estudio para PAN at CLEF 2020

Ignacio de Toro Ruiz

Master Big Data Analytics UPV
igdeto1@masters.upv.es

Abstract En este informe se presenta una solución a la tarea de la detección de usuarios de twitter asiduos a la propagación de 'Fake News' mediante arquitecturas de Deep Learning. Para ello, se ha hecho uso de los métodos de entrenamiento SVM con Kernel Lineal y Random Forest para tratar el problema según el idioma del conjunto de datos: Español e Inglés respectivamente. De este modo hemos llegado a una precisión mayor del 80% para el caso Español y mayor a 73% para el caso Inglés.

1 Introducción y descripción de la Tarea.

El término anglosajon 'Fake News' se usa para denominar noticias inventadas o propaganda con contenido desinformativo. Los motivos que dan lugar a la creación de estas 'noticias falsas' son para dañar la reputación de alguien, para adoctrinar al lector o para causar sensacionalismo. Este tipo de noticias se suelen retransmitir tanto por los medios de comunicación tradicionales (televisión, radio, prensa...) como por los más modernos: redes sociales como Facebook o Twitter, siendo este último el objetivo de nuestro estudio.

Por ello el objetivo de esta tarea es poder identificar a posibles divulgadores de desinformación en Twitter, partiendo de la siguiente información:

• Dos bases de datos (una correspondiente a tweets en inglés y otra en español) cada una con 300 archivos XML correspondientes a un usuario de twitter cada uno.

• Cada archivo XML contiene 100 tweets del autor.

• Cada base de datos contiene un archivo 'truth.txt' con dos columnas: la primera contiene la 'ID' del autor y la segunda un factor que determina si el autor es propenso a la transmisión de desinformación.

2 Funciones utilizadas en el código.

Para procesar el conjunto de datos para el entrenamiento usaremos dos funciones auxiliares:

• GenerateVocabulary: Dado el conjunto de datos, preprocesa los textos contenidos (teniendo en cuenta o no las minúsculas, los signos de puntuación, los espacios en blanco, una lista de palabras, o los números) en un número N dado de tweets por autor y obtiene las 'n' palabras más frecuentes.

• GenerateBow: Dado el conjunto de datos, y el vocabulario con las frecuencias de las palabras obtenido mediante la anterior función, realiza un análisis de sentimiento (positivo a negativo) y de emoción (amor, alegría, enfado, tristeza y sorpresa) mediante la API de Symanto, y además se le ha añadido un análisis de legibilidad del texto mediante el paquete de R de análisis cuantitativo de textos Quanteda. Para llevar a cabo este análisis de legibilidad, se usa la función '$textstat\_readability(text, measure = "Flesch")$' que mide la legibilidad de un texto mediante la prueba de nivel de facilidad de lectura de Flesch, de modo que un resultado cerca de 0 indicaría una muy difícil lectura y uno de cerca de 100, una muy fácil lectura.

3 Resultados.

Para el lenguaje español hemos usado la siguiente configuración de parámetros:

• Los parámetros antes mencionados $N = 10$, $n = 10000$.

• La función GenerateVocabulary con los parámetros por defecto, añadiendo una lista de palabras a no tener en cuenta, que son: "hashtag", "user" y "url".

• La función GenerateBow con los parámetros de análisis de legibilidad, de emoción y de sentimiento activados.

• Y por último, obviamente: Lenguaje="es".

Así entrenamos un modelo SVM con Kernel lineal mediante validación cruzada con 10 pliegues y 3 repeticiones, de manera que obtenemos el siguiente resultado:

Support Vector Machines with Linear Kernel
300 samples
10444 predictors
2 classes: 'NO', 'YES'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)

Summary of sample sizes: 270, 270, 270, 270, 270, 270, ...
Resampling results:
Accuracy Kappa
0.8077778 0.6155556

Tuning parameter 'C' was held constant at a value of 1.

Es decir, un 80.78% de precisión.

Para el lenguaje inglés hemos usado la siguiente configuración de parámetros:

• Los parámetros antes mencionados $N = 10$, $n = 10000$.

• La función GenerateVocabulary con los parámetros por defecto, sin añadir ninguna lista de palabras adicional.

• La función GenerateBow con los parámetros de análisis de emoción y de sentimiento activados, esta vez no se han usado los de legibilidad porque apenas surtian efecto en la precisión.

• Y por último: Lenguaje="en".

En este caso hemos usado el método del 'Random Forest', que nos resulta en una mayor precisión, de esta manera entrenamos el modelo mediante validación cruzada con 10 pliegues y 3 repeticiones, de forma que obtenemos el siguiente resultado:

Random Forest
300 samples
1022 predictors
2 classes: 'NO', 'YES'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times) Summary of sample sizes: 270, 270, 270, 270, 270, 270, ...
Resampling results across tuning parameters:
mtry Accuracy Kappa
2 0.7311111 0.4622222
45 0.7322222 0.4644444
1022 0.7211111 0.4422222

Accuracy was used to select the optimal model using the largest value. The final value used for the model was mtry = 45.

Es decir, un 73.22% de precisión.

4 Bibliografía

References

1. Text Mining en Social Media, Máster Big Data Analytics.
   Paolo Rosso and Francisco Rangel (2020) - UPV and SYMANTO AI.
2. Fake News Detection: A Deep Learning Approach
   A. Thota, P. Tilak, S. Ahluwalia, N. Lohia (2018) - SMU Data Science Review, Vol. 1 No. 3 Art. 10
3. A new readability yardstick.
   Flesch R (1948) - Journal of Applied Psychology. 32 (3): 221-233
4. Quanteda: An R package for the quantitative analysis of textual data.
   K. Benoit, K. Watanabe, H. Wang, P. Nulty, A. Obeng, S. Müller and A. Matsuool. (2018) - Journal of Open Source Software, 3 No. 33 Pages 776