Fake news detection through Machine Learning

Kerman Sanjuan Malaxechevarria

Euskal Herriko Unibertsitatea/ Universiadad Del Pais Vasco

Objectives

- Clasificar una noticia como verdadera o falsa, leyendo el contenido de esta.
- Utilizar dos clasificadores diferentes.
- Utilizar dos técnicas de vectorización de texto.
- Evaluar y comparar resultados con las combinaciones de los puntos anteriores.

Introduction

Hoy en día, con el extendido uso de internet la cantidad de desinformación que se pública es cada vez mayor. Es por ello que en algunas redes sociales han empezado a implementar sistemas de detección de noticias falsas y/o fraudulentas. En este trabajo se desarrollará un modelo de predicción que clasifique las noticias como falsas o verdaderas.

Figure 1:Pipeline usado en este proyecto

Materials

Los siguientes recursos han sido utilizados para el desarrollo de esta tarea:

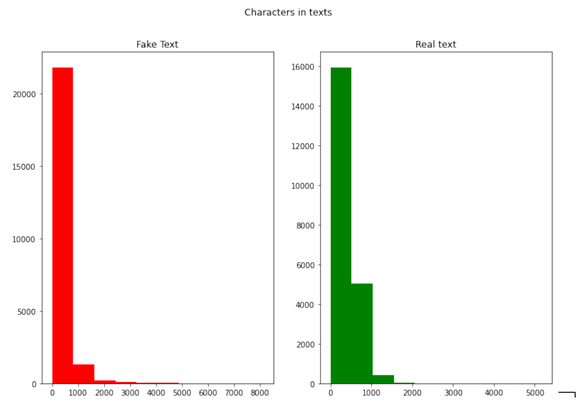
- **Fake News Dataset:** Conjunto de datos a utilizar para desarrollar el modelo de predicción.
- **Gensim/Tensorflow:** Dos frameworks relacionados con aprendizaje supervisado.

Data treatment

- **Pre-processing** Tamaño de los datos.

Pre-process	Docs	Doc Size	Class
Raw	44183	110037	2
TF-IDF (Frequency = 100)	44183	7695	2
TF-IDF (Frequency = 200)	44183	4988	2
Embedding	44183	200/89K	2

- **Exploratory Data Analysis**



Mathematical Section

Vectorización de las frases con **TF-IDF**

$$\mathrm{tf}(t,d) = \frac{\mathrm{f}(t,d)}{\max\{\mathrm{f}(t,d) : t \in d\}} \tag{1}$$

$$\mathrm{idf}(t,D) = \log \frac{|D|}{|\{d \in D : t \in d\}|} \tag{2}$$

$$\mathrm{tfidf}(t,d,D) = \mathrm{tf}(t,d) \times \mathrm{idf}(t,D) \tag{3}$$

El valor tf-idf aumenta proporcionalmente con las apariciones de la palabra en el documento, pero se equilibra con las apariciones en la colección de documentos, lo que permite ver que la aparición de unas palabras es más común que la aparición de otras.

Methods

Se han utilizado cuatro *approach* diferentes, combinando las vectorizaciones con los diferentes modelos.

Treatments	Features
L.Reg + TF-IDF	1760
L.Reg + Embedding	200
NN + TF-IDF	4860
NN + Embedding	89K

Table 1:Tiempo de parametrización y resultado

En el caso de *NN + Embedding* hemos creado un embedding a partir de la libreria Tensorflow usando *Tokenizadores*. De ese modo obteniamos una mayor compatibilidad entre la red neuronal y los datos. Es por ello que la dimensión en este caso es más grande que en casos anteriores.Por otro lado, el conjunto de datos con TF-IDF aplicado se ha mantenido igual en ambos clasificadores.
En ambos clasificadores se ha realizado una hiperparametrización, el cual ha permitido aproximarse a los valores apropiados para el clasificador.

Experimental results

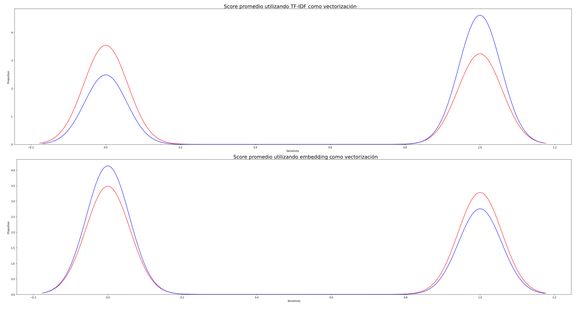


Figure 2:Resultados promedio obtenidos con TF-IDF y embedding (promedio)

Estos han sido los resultados después de de probar los cuatro *approach diferentes*.

Treatments	Train Score	Test Score
L.Reg + TF-IDF	0.751	0.545
L.Reg + Embedding	0.642	0.535
NN + TF-IDF	0.4842	-
NN + Embedding	0.97	0.7316

Table 2:Resultados de los diferentes *approach*

Como se observa en los resultados, el mejor resultado nos lo ha dado la red neuronal combinado con la vectorización *word-embedding*.

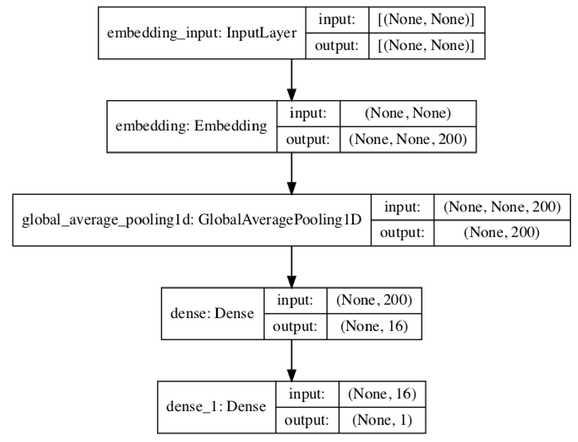


Figure 3:Red Neuronal empleada.

Conclusions

- En nuestro caso de uso, el word-embedding ha funcionado mucho mejor que TF-IDF.
- Un buen preproceso es clave.
- Debido a que la cantidad de valores 0, TF-IDF no es apropiado para redes neuronales.