Clasificación de textos digitales con *machine learning*

Ing. Eduardo Maldonado Cárdenas, M.I. David Maloof Flores, M.A. Olanda Prieto Ordaz y el M.I.D. Miguel Ángel López Santillán

Universidad Autónoma de Chihuahua / Facultad de Ingeniería
FINGUACH Año 6, Núm. 20, junio - agosto del 2019

Desde el inicio de la civilización, los seres humanos han visto la necesidad de almacenar información (en monumentos, papiros, libros, bases de datos, entre otros) de todo tipo. Con el paso del tiempo el hombre también se dio cuenta de que la recuperación de la información era algo crucial, así que comenzó a compilar grandes cantidades de información en bibliotecas organizadas por secciones para hacer la recuperación de esta de manera más práctica.

La clasificación de documentos en línea es una tarea de gran importancia. Ante el crecimiento geométrico del *Internet* y los datos que éste almacena, se ha convertido en una necesidad imperante el contar con herramientas de clasificación automática, de tal forma que la información almacenada pueda ser de provecho. Los grandes repositorios de información actualmente esparcidos digitalmente solo son de utilidad si se les extrae provecho por medio de técnicas como *Information Retrival. Google* y otras compañías apuestan enormemente por técnicas de inteligencia artificial que permitan una mejor explotación del inmenso volumen de datos disponibles.

Pero la clasificación de la información no es el único problema presente: la extracción de la información es también algo de que ocuparse, ya que el lenguaje natural puede llegar a ser ambiguo, convirtiendo la extracción de la información en algo complejo. El Procesamiento de Lenguaje Natural (NLP por sus siglas en inglés) es un conjunto de herramientas que nos ayudan a procesar el texto para hacer más fácil la comprensión del lenguaje humano hacia la máquina y que ésta pueda entonces hacer un trabajo de clasificación certero.

Una solución a este problema pueden ser los clasificadores lineales, que también son un buen punto de partida para la clasificación de texto, donde destaca su simplicidad y su gran potencial para trabajar con grandes cantidades de texto.

En este trabajo se presenta una propuesta de un diseño y desarrollo de una herramienta de *software* que permite clasificar textos digitales según el tópico principal del mismo. Para lograrlo se vectorizaron dos *datasets* y se entrenaron diversos algoritmos de *machine learning* para que eventualmente se pudieran clasificar textos nuevos (nunca antes vistos por el algoritmo). Mediante la implementación de diversos algoritmos de clasificación se dio robustez al *software* para que pudiera realizar la categorización con el mejor conjunto de parámetros para el algoritmo que mejor generalizó.

Se utilizaron *datasets* de gran relevancia para esta investigación, como lo es el *dataset Reuters*-21578, una colección de noticias de 1987, siendo constantemente corregido y mejorado significativamente; así como el *dataset* de IMDB, con más de 50 mil documentos con reseñas de películas.

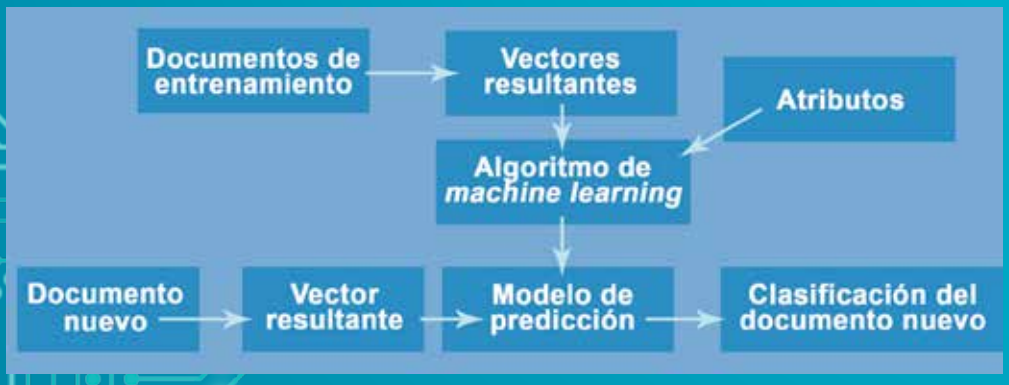

Figura 1. Propuesta de solución.

Para poder procesar un *dataset* textual es necesario convertir "los datos a un lenguaje que una computadora pueda comprender". Con esto en mente se optó por una técnica relativamente novedosa como son los vectores de palabras con el algoritmo *word2vec (Skip-gram)* en su implementación en *Gensim* para el lenguaje de programación *Python*. Se entrenaron dos vocabularios (uno para cada *dataset*) y así poder obtener los vectores de cada palabra utilizando los parámetros. Posterior a este paso, fue necesario obtener una representación de los documentos completos a clasificar. Para este proceso se tomó como base el concepto de que un documento puede ser expresado como la adición de sus palabras.

Para obtener un valor de importancia por palabra en cada texto se utilizó el valor estadístico conocido como *tf-idf (term frequency - inverse document frequency)*. Este valor se obtiene al contar la aparición de cada término por documento y por *dataset*. De esta manera, una palabra que aparece en un documento tiene una relevancia para la actividad de clasificar, sin embargo, se debe calcular un valor de desfase *(offset)* para términos que aparecen de manera regular en todo el *dataset* pero que tiene poco poder clasificador (artículos, pronombres, entre otros).

$$\text{tf-idf}_{t,d} = (1 + \log \text{tf}_{t,d}) \cdot \log \frac{N}{\text{df}_t}$$

Figura 2. Fórmula para la obtención de la medida numérica que expresa cuán relevante es una palabra para un documento en una colección.

Una vez obtenidos los vectores para cada documento por medio del método previamente explicado, se alimentaron a cuatro distintos clasificadores: Red Neuronal Recurrente *(LSTM) Support Vector Machine (SVM) Extra Trees (ET)* y *Random Forest (RF)* implementados en la librería de *Machine Learning* para *Python Scikit-Learn*.

En la actualidad, los algoritmos de *machine learning* son evaluados por medio de medidas de rendimiento como *accuracy* o RMSE *(Root Mean Square Error)* dependiendo si el problema es de clasificación o regresión respectivamente. Para obtener estos resultados se dividió el *dataset* en una partición de entrenamiento consistente en el 70 % de los datos y otra partición de prueba formada por el 30 % restante de los mismos. En este trabajo se puso de manifiesto la utilidad y eficacia de estos métodos en tareas elementales del procesamiento de lenguaje natural (NLP) como son clasificación de textos y análisis de opiniones o sentimientos. La era de *machine learning* y en específico de *data science*, está apenas iniciando.

Referencias

"Vicomtech. "procesamiento de lenguaje natural"." http://www.vicomtech.org/t4/e11/procesamiento-del-lenguaje-natural. Accessed: 2018-04-15.

A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," 2016. cite arxiv:1607.01759.

D. D. Lewis, "Reuters-21578." https://archive.ics.uci.edu/ml/datasets/reuters-21578+text+categorization+collection.

R. R ˇ ehu ˚ r ˇ ek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, (Valletta, Malta), pp. 45–50, ELRA, May 2010. http://is.muni.cz/publication/884893/en.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.*

Resultados para el *dataset* de noticias de la agencia *Reuters*				
	LSTM	SVM	ET	RF
Params	Dropout=0.3	C=1, degree=2, gamma=1, kernel=poly, coef0=0	Criterion=gini, max_features=sqrt, n_estimators=200, min_samples_leaf=6	Criterion=entropy, max_features=sqrt, n_estimators=200, min_samples_leaf=4
Accuracy	**89.54**	81.64	81.54	82.15
Resultados para el dataset de análisis de opiniones de IMDB				
	LSTM	SVM	ET	RF
Params	Dropout=0.2	C=10, gree=2, gamma=1, kernel=poly, coef0=0	N_estimators=200, criterion=entropy, max_features=none, n_jobs=-1	N_estimators=200, criterion=entropy, max_features=log2, n_jobs=-1
Accuracy	70.71	49.56	75.71	74.76

Figura 3. Tabla con resultados de la experimentación en cuestión.