Petro-News

Daniel de la Osa Fernandez D.OSA@ESTUDIANTES.MATCOM.UH.CU
Grupo C-511

Jose Luis Alvarez de la Campa J.ALVAREZ@ESTUDIANTES.MATCOM.UH.CU
Grupo C-512

Wendy Diaz Ramires W.DIAZ@ESTUDIANTES.MATCOM.UH.CU
Grupo C-512

Resumen

Se desarrolla una metodología para realizar la tarea de clasificación de noticias usando *SVM* y probando diferentes técnicas para la extraccón de características como conteo de palabras, *word embedings* y *BERT* alcanzando un 98 % de precisión. La tarea de clasificación consiste en determinar para cada artículo si es o no una noticia sobre el petróleo.

Palabras Clave: *Word embedings*, *BERT*, *SVM*, *word2vec*, *DNN*, *transfer learning*.

1. Introducción

Hoy en día la cantidad de información con que cuenta la humanidad es enorme. Esto es en gran medida gracias al uso popular de las nuevas tecnologías como son el Internet. El correcto uso de esta información ha probado en varios casos ser de gran utilidad, ejemplo de esto son los casos del *Brexit* y la elecciones por la presidencia de los *EUA* del 2018 relativos a la ya clausurada empresa de *Cambridge Analytica*. Particularmente, los medios de comunicaciones, como son los periódicos digitales, constituyen una amplia, extensa, 'verídica' y actualizada fuente de información.

Por otra parte en el proceso de toma de decisiones de cualquier empresa es de vital importancia partir de un contexto actualizado con respecto al mercado y la situación político-social del medio en que se desenvuelve. Sería entonces de gran interés para esta empresa un sistema que automatizara el proceso de recuperar y filtrar la información de estos sitios con respecto a un tema en específico; en nuestro caso particular, esto son las noticias relacionadas con el petróleo.

Este proyecto contituye una primera aproximación a la clasificación automática de noticias relativas al petróleo.

2. Representación de textos

Para poder llevar a cabao la tarea de clasificación fue necesario definir el conjunto de características que se debían elegir para poder entrenar un clasificador. Esta fase es sumamente importante ya que elegir estas es crucial para el buen rendimiento del modelo. Para lograr esto se debía buscar una manera de representar las noticias como vectores numéricos que de alguna forma describieran el contenido de este.

2.1 Bolsa de Palabras

Para crear un modelo base la primera estrategia que se siguió fue usar una matrix término-documento, es decir usando una forma clásica para representar doumentos [11, 12], que consiste en que cada documento está representado por un vector de palabras del vocabulario y en cada componente se encuetra la frecuencia con que ocurre esa palabra en ese documento. Esta forma es sencilla pero presenta diferentes desventajas que otro enfoque pudiera mejorar. Esta forma de representar los artículos, tiene la desventaja de que son vectores muy grandes, en este caso legaron a tener 120 mil componentes, es decir eso son muchas características para que el clasificador entrene. Además no mide de ninguna forma la semántica que existe entre las palabras. Por lo que se pudiera mejorar buscando otra representación.

2.2 Word Embedings

Tratando de buscar una mejor representacíon de los artículos, se usó *word embedings* [1, **?**] para esto. Está nueva forma de representación usa modelos de redes neuronales entrenados que logran transformar cada palabra en un vector real de n dimensiones. Existen diferentes algorítmos para esto como son *word2vec*[10, 9] y *GloVe*.

Para nuestro problema en partícular se usó el modelo del lenguaje de *Spacy*, con el cuál se transformaron los artículos a vectores de 96 dimensiones. *Spacy* utiliza *word2vec* para entrenar este modelo.

Esta forma de representación mejora con respecto a la anterior pero, a pesar que logra captar mayor contenido semántico a nivel de oración, a nivel de documento no es así.

2.3 BERT Embedings

Así llegamos al usar el estado del arte en representación vectorial de textos, *BERT* por sus siglas en inglés que signifcan *Bidirectional Encoder Representation of Transformers* presentado por *Google* en el año 2018[3, 6, 7, 8]. Usando este modelo se lograron obtener resultados que representan el estado del arte en 11 tareas de procesamiento de lenguaje natural, entre ellas la clasificación por lo que se ajusta a nuestro problema. *BERT* es un modelo que sigue una arquitectura de una red neuronal profunda bidireccional usando *atención*, lo cual le ha dado gran ventaja sobre las demás representaciones.

Para esta tarea en específico se usó el modelo preentrenado multilenguaje de *BERT*. Este genera por cada artículo un vector de 768 dimensiones que va ha ser nuestro vector de características para entrenar el clasificador. La tarea de transformar el corpus de entrenamiento que contiene 22 mil ejemplos entrenantes mediante *BERT* duró 6 horas en una computadora portátil con $16gb$ de RAM , una CPU $i7$ de 7^{ma} generación y una GPU *Nvidia* 1050. Es necesario al menos $12gb$ de ram para poder correr el modelo pre-entrenado de *BERT*. El modelo multilenguaje contiene 110 millones de parámetros.

3. Modelos

Para realizar la tarea se probaron como clasificadores, *(*SVM)[citar paper svm viejo] y *DNN-Aquitectures*[4, 13]. Variando la entrada de estos clasificadores usando las diferentes representaciones vistas anteriormente. Ahora se explicará como fue el proceso con ambos modelos.

3.1 Suport Vector Machine

Inicialmente para construir un modelo base se eligió este modelo ya que este con kernel RBF fue originalmente creado para la tarea de clasificación binaria [2], con una primera capa inicial de bolsa de palabras para transformar los artículos de entrada. Este modelo se comportó bien en las primeras pruebas con el 1 % del corpus de entrenamiento por lo que se entrenó con todo y alcanzó un alrededor del 90 % de precisión obteniendo una buena base para los posteriores experimentos.

Antes de cambiar el modelo se repitió el proceso usando *word embedings*, esto agilizó el proceso de entrenamiento ya que las características se redujeron de decenas de miles a solo 96. Para un corpus pequeño (1 %) este modelo aprendió mucho más que el anterior, pero para el corpus completo se comportó de manera similar con alguna mejora pequeña alcanzando valores de precisión del 91 %.

Finalmente se mantuvo *SVM* y se probó de nuevo ahora pasándole como entrada los artículos transformados por *BERT*. Tratando de buscar resultados que representaran el estado del arte. Para solo el 1 % de los datos este modelo sobrepasa a los dos anteriores en más de un 20 % alcanzando poco más del 90 % con evidencia que podía mejorar aún más. Se entrenó entonces con el corpus completo y se alcanzó alrededor de un 97 % superando ampliamente los anteriores, mostrando la capacidad de *BERT* para captar el contenido semántico de los documentos.

3.2 Redes Neuronales

Aunque se obtuvieron ya buenos resultados con *SVM*, se decidió probar una red básica. Por eso usamos el regresor logístico con una función sigmoidal de activación, para ver como se comportaba. Este dió resultados tan buenos como el modelo anterior ambos rondando el 98 %. Esto llevó a tratar de mejorar algo más añadiendo dos nuevas capas ocultas de tamaño 100 y 30 respectivamente ambas con función de activación sigmoidal logrando casi un 99 % en el corpus de prueba. Estas redes fueron entrenadas 20 épocas con el 80 % del corpus.

4. Corpus

No existe un corpus para realizar esta tarea en específico por lo que se debió construir uno a partir de las noticias de diferentes periódicos online1. Esto se logró usando una herramienta llamada *news-please*[5].

Periódico	**URL**
El pais	https://elpais.com ,
Energia16	https://energia16.com ,
Reuters	https://lta.reuters.com ,
Reuters	https://mx.reuters.com ,
EFE	https://www.efe.com ,
Europapress	https://www.europapress.es ,
Prensa Latina	https://www.prensa-latina.cu ,

Cuadro 1: Periódicos utilizados

Para asignarle un label a este corpus que consta de 22 mil artículos se usaron ciertas heurísticas. Se asumió primeramente que las noticias que vinieran de sitios que eran exclusivamente sobre temas de energía, petróleo, etc., se etiquetaron como positivas. También se etiquetaron como positivas aquellas que contenían palabras claves seguras que indicaban la referencia a este tema como lo es petróleo y que posteriormente fueron revisadas a una a una.

Para los casos negativos se tomaron secciones de periódicos que fueran de temas suficientemente aislados como lo son deporte, cultura etc. Mientras que se tomaron como negativas también las que no contenían palabras claras sobre el tema a clasificar en cuestión. Este fue un enfoque semi-supervisado que puede mejorarse luego con retroalimentación de los modelos. Al completar el corpus se obtuvo cerca de 9 mil artículos positivos y 13 mil negativos.

5. Resultados

En la tabla se puede observar una comparación de los modelos en cuanto al *precisión*.

Modelo	Precisión	Precisión
SVM-CountVectorizer	90	87.3
SVM-Spacy-Embedings	91	89
SVM-BERT-Embedings	97	97
DNN-BERT-Embedings	98.9	98.4

Cuadro 2: Resultados

El modelo final elegido fue el de redes neuronales que obtuvo los mejores resultados y en las pruebas de validación también se desempeñó con buenos resultados55.

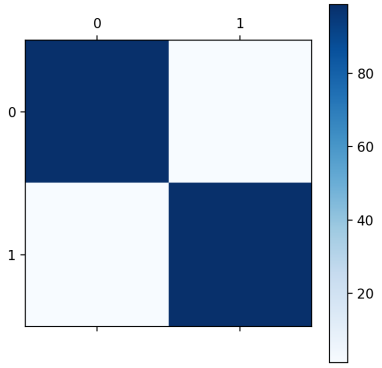

Figura 1: Matriz de Confusión del modelo *DNN*

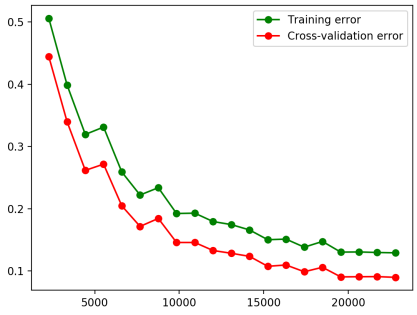

Figura 2: Curva de entrenamiento del modelo *DNN*

6. Conclusiones

Se creó una metodología basada en *BERT* y redes neuronales con las cuales se alcanzaron excelentes resultados en esta tarea, logrando 98.9 % de precisión. La arquitectura del modelo es independiente del tema a clasificar en cuestión, luego con diferentes corpus de entrenamiento se pueden entrenar clasificadores de noticias para diferentes tópicos. Además se mostró como *transfer learning* puede resultar una muy buena estrategía de términos de ahorro de recursos y tiempo en el desarrollo de nuevas herramientas para el procesamiento del lenguaje natural.

7. Recomendaciones

Como posibles tareas futuras se pudiera estudiar la forma de crear un proceso de retroalimentación del modelo a partir de las noticias sobre el petróleo elegidas por el usuario como las más significativas, de esta manera se pudiera reentrenar o ajustar los pesos del modelo logrando mayor especialización en subtemas relacionados que son más interesantes para el usuario.

También se pudiera investigar en usar menos componentes de los que genera *BERT* reduciendo la cantidad de dimensiones y que puede generar mejoras en el desempeño del modelo siguiendo lo descrito por [6].

Referencias

[1] Felipe Almeida and Geraldo Xexéo. Word Embeddings: A Survey. jan 2019.

[2] Marc Claesen, Frank De Smet, Johan A. K. Suykens, and Bart De Moor. Fast Prediction with SVM Models Containing RBF Kernels. mar 2014.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. oct 2018.

[4] Ashwin Geet D'Sa, Irina Illina, and Dominique Fohr. Towards non-toxic landscapes: Automatic toxic comment detection using DNN. nov 2019.

[5] Felix Hamborg, Norman Meuschke, Corinna Breitinger, and Bela Gipp. news-please: A Generic News Crawler and Extractor. In *Proceedings of the 15th International Symposium of Information Science*, pages 218–223. Verlag Werner Hülsbusch, 2017.

[6] Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. Revealing the Dark Secrets of BERT. aug 2019.

[7] Jindřich Libovický, Rudolf Rosa, and Alexander Fraser. How Language-Neutral is Multilingual BERT? nov 2019.

[8] Timothee Mickus, Denis Paperno, Mathieu Constant, and Kees van Deemeter. What do you mean, BERT? Assessing BERT as a Distributional Semantics Model. nov 2019.

[9] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. jan 2013.

[10] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. oct 2013.

[11] G Salton. *The SMART Retrieval System—Experiments in Automatic Document Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1971.

[12] Karen Sparck Jones. *Synonymy and Semantic Classification*. Edinburgh University Press, Edinburgh, Scotland, Scotland, 1986.

[13] Jacques Wainer. Comparison of 14 different families of classification algorithms on 115 binary datasets. jun 2016.