Agrupación no supervisada de textos cortos en redes sociales

JEFFREY ORIHUELA, HERLESS ALVARADO, and JUAN VARGAS, Universidad de Ingeniería y Tecnología, Peru

LINK DEL REPOSITORIO EN GITHUB:
https://github.com/juan4056/BigDataEmbeddingWords

CCS Concepts: • **Clustering**; • **Deep Learning**; • **Non Supervised Learning**;

ACM Reference Format:
Jeffrey Orihuela, Herless Alvarado, and Juan Vargas. 2021. Agrupación no supervisada de textos cortos en redes sociales. 1, 1 (December 2021), 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCCIÓN

Redes sociales, correos electrónicos y blogs proporcionan bastante información donde el text clustering o agrupación de textos nos permite agrupar esos datos en grupos con características similares. Los grupos de texto han demostrado ser de ayuda en muchas aplicaciones, incluyendo análisis de sentimientos, sistemas de preguntas y respuestas automáticas, filtración de spams, detección y monitoreo de temas entre otros.

Actualmente, la agrupación automática de textos cortos proveniente del internet se está convirtiendo en un campo de investigación interesante. Esto se debe al auge de las redes sociales, como Twitter e Instagram, así también como los foros en línea, StackOverflow y Reddit entre otros más. A su vez, estas plataformas han hecho que el estudio para la agrupación de textos cortos tenga una mayor relevancia en el tema de procesamiento de lenguaje natural.

Sin embargo, la agrupación de textos cortos presenta desafíos adicionales a comparación de la agrupación de textos largos; debido a que tradicionalmente los textos son representados como una bolsa de palabras o mediante la técnica de término de frecuencia y frecuencia inversa de documentos (TF-IDF) vectorizando el texto.

Como resultado de la corta longitud de estos textos, sus representaciones vectoriales tienen alta dimensionalidad y diversidad semántica produciendo vectores característicos poco densos. Recientemente, representaciones de baja dimensionalidad han demostrado potencial para resolver el problema de vectores característicos poco densos de textos cortos como Skipgram[5] o GloVe[6].

Con la ayuda de las *word embeddings* las redes neuronales tienen un buen desempeño para la construcción de modelos gramaticales como las redes neuronales recurrentes. Las cuales son apropiadas para datos que vienen en un orden definido como en el lenguaje natural: oral o escrito. En estas redes, la recurrencia es la entrada de datos desde una iteración anterior produciendo que las características aprendidas del pasado se mantengan presentes en el contexto del modelo. Por otra parte, las redes neuronales convolucionales aplican sus filtros convolucionales para capturar características locales y luego optimizarlas utilizando la operación pooling. Después dichas características aprendidas pueden representar de mejor manera a las *word embeddings*. Por lo tanto, el resultado facilita las tareas de agrupamiento de textos cortos.

A pesar de estos avances, el campo de clasificación no supervisado para textos cortos tiene aspectos por mejorar al momento de incorporar características semánticas a los modelos de entrenamiento no supervisado, principalmente para conversaciones o comentarios provenientes de redes sociales donde se genera información cada segundo.

El siguiente trabajo explora la agrupación de textos cortos en redes sociales a través de un modelo no supervisado basado en un autoencoder utilizando redes neuronales convolucionales para abstraer más las *word embeddings*. En los próximos subcapítulos se describe más a detalle el problema, nuestra justificación y objetivo. La siguiente sección describe los trabajos realizados en el campo de clasificación de textos cortos así como una revisión del estado del arte de los modelos más influyentes. La tercera sección define conceptos claves para que el lector pueda sentirse cómodo con el tema de investigación. La sección 4 y 5 se refieren a la propuesta y los resultados experimentales del trabajo realizado.

1.1 Motivación y Contexto

La motivación para este trabajo resalta en el requerimiento de entrenar a modelos de inteligencia artificial con un conjunto de datos previamente clasificados. En este contexto nos vemos forzados a crear nuestro propio *dataset* de entrenamiento clasificando manualmente las muestras en caso no exista alguna fuente de información trabajada anteriormente y sea actual. Un ejemplo son las redes sociales ya que debido a la interacción entre sus usuarios se genera nueva información de forma muy acelerada lo cuál provoca una fuente de información no clasificada. Por lo tanto, la motivación para desarrollar este trabajo es poder otorgar una método de clasificación de textos no supervisados para textos cortos que pueda ser aplicado de forma rápida y sea de utilidad para el entrenamiento de modelos de Procesamiento de Lenguaje Natural.

1.2 Descripción del problema

La agrupación de textos cortos a diferencia de la agrupación normal de textos tiene el problema de la escasez. Es decir, la ocurrencia de las palabras en los textos cortos es mínima dentro un extenso vocabulario. Como consecuencia la representación de vectores característicos no aporta mucha información para los modelos de agrupación tradicional.

Sin embargo, las nuevas representaciones para textos cortos no requieren una dimensión del tamaño del corpus o vocabulario. Al contrario, son de baja dimensión porque utilizan otras métricas para representar las palabras. Por lo tanto, el diseño y optimización

Authors' address: Jeffrey Orihuela; Herless Alvarado; Juan Vargas, jeffrey.orihuela@utec.edu.pe, Universidad de Ingeniería y Tecnología, Lima, Peru.

XXXX-XXXX/2021/12-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

de nuevos modelos para la agrupación de textos cortos utilizando *word embeddings* y modelos como redes neuronales es un área por investigar.

1.3 Justificación

Debido al gran volumen de información generado por redes sociales y servicios de mensajería, la agrupación de textos por atributos nominales es una tarea constante para el preprocesamiento de este tipo de información. La mayoría de modelos de *clustering* requieren un conjunto de datos previamente etiquetados para lograr un entrenamiento correcto del modelo. Por consiguiente la creación de un *dataset* implica una participación humana en la tarea del etiquetado.

Por otro lado, modelos de *clustering* no supervisados para textos cortos es un campo de investigación creciente. Muchos de estos modelos pueden incluirse dentro de un sistema de aprendizaje no supervisado más sofisticado para tareas de procesamiento de lenguaje natural.

1.4 Objetivo

La presente investigación pretende efectuar un modelo no supervisado de manera que agrupe eficientemente textos cortos provenientes de servicios de mensajería o redes sociales con una precisión mayor al 85% reduciendo el tiempo de preprocesamiento en un proyecto de ciencia de datos.

1.5 Aporte

Presentar una herramienta API de código abierto para la comunidad de *data science* cuyo propósito sea resolver la agrupación de textos cortos no categorizados en un conjunto de datos.

2 TRABAJOS RELACIONADOS

Los trabajos de investigación acerca del agrupamiento de textos cortos pueden ser vistos desde dos alcances distintos. El primer enfoque pone atención a las *word embeddings* más adecuadas para el algoritmo de agrupamiento K-means. El segundo enfoque se relaciona más con el aspecto del aprendizaje no supervisado para la extracción de características profundas de las *word embeddings*.

El trabajo de Evangelos Milios y Magdalena Jankowska titulado *Enhancement of Short Text Clustering by Iterative Classification*[7] se basa en remover resultados atípicos o *outliers* de los algoritmos de agrupamiento para luego asignarlos en sus apropiados cluster mejorando la precisión del agrupamiento. Para este propósito, se entrena un modelo clasificador usando las etiquetas proporcionadas por los resultados del K-means con los *outliers* reagrupados. Luego se clasifican los *outliers* mediante el modelo entrenado a sus respectivos grupos para una siguiente iteración. Como resultado se entrega una mejor agrupación de muestras de textos cortos.

El trabajo de investigación sobre *word embeddings* y su mejor representación para la afinidad con K-means llamado *Representation Learning for Short Text Clustering* publicado por Yin[12] propone un *autoencoder* basado en una red neuronal grafal como encoder para fusionar las características de las *word embeddings* pre entrenadas con las características obtenidas del *encoder*. La agrupación la consigue mediante un K-means que ordena las *word embeddings*. Se obtienen métricas con los métodos de *Normalized Mutual Information* y *Clustering Accuracy*. Demostrando que su modelo obtiene mejor precisión con las *word embeddings* producidas por BERT a diferencia de otros métodos como *SkipGram* o *Bag of Words*.

Otro trabajo sobre métodos de representaciones profundas no supervisada llamado *An Overview of Unsupervised Deep Feature Representation for Text Categorization* por Wang[10] hace un *survey* de los resultados de precisión de K-means con un grupo de *datasets* donde sus vectores característicos vienen definidos. No enfocándose en la creación de las *word embeddings* sino utilizando estas representaciones como entrada a un *autoencoder* para luego usar los resultados del *autoencoder* en K-means. Como aporte se puede observar el comportamiento de diferentes autoencoders en los datasets.

Xu[11] publicó su trabajo de investigación llamado *Self-taught Convolutional Neural Networks for Short Text Clustering* exponiendo un modelo supervisado que obtiene el conjunto de validación de su propio conjunto de entrenamiento con el próposito de conseguir menores dimensiones y agrupar las muestras. El conjunto de validación es creado al transformar las *word embeddings* a través de técnicas de reducción de la dimensionalidad logrando ser representadas de forma más compacta. Con este conjunto se calcula el error a la red neuronal para que luego de su optimización se utilice la capa de salida donde se encuentran las características profundas y se proceda con el algoritmo de K-means. Como resultado de la investigación se muestra que el modelo obtiene los mejores resultados de precisión de agrupamiento a comparación de otras técnicas.

3 MARCO TEÓRICO

3.1 Word Embedding

Existen dos tipos de representaciones para las palabras: representaciones basadas en conteo y representaciones basadas en predicciones. Actualmente, los métodos basados en predicciones son los que abarcan un rango más grande de aplicaciones con respecto a los métodos basados en conteo pero a nivel de funcionalidades básicas los dos métodos cumplen con lo esperado.

Asimismo, estos modelos basados en predicciones como *Skipgram* y *Continuous Bag-Of-Words* (CBOW)[4] proponen una arquitectura de una sola capa basada en el producto punto de dos palabras en forma de vectores. Su objetivo es predecir el contexto de la palabra dada la misma como entrada a través de de una evaluación de analogía de palabras. Como resultado estos modelos han demostrado la capacidad de aprender patrones lingüísticos al crear relaciones entre vectores característicos. También existe un modelo llamado *Global Vectors*(GloVe)[6] donde las palabras tienen también una representación vectorial y sus distancias entre ellas se miden a través de su similaridad semántica. El modelo es entrenado con la función de costo Mínimos Cuadrados usando como métrica el conteo estadístico de la coocurrencia de dos términos. Finalmente, este modelo define una elegante estructura vectorial en el espacio para una óptima precisión en casos de analogías de palabras y reconocimiento.

3.2 Redes Neuronales Convolucionales

Uno de los tipos de redes neuronales más potentes para el procesamiento de datos dentro del campo de *deep learning* son las redes neuronales convolucionales. Inicialmente, la idea se inspiró en una

parte del cerebro llamada corteza visual. En este espacio los campos receptivos de las neuronas reciben información sensorial que se propaga alrededor de las neuronas de activación para conseguir un mejor entedimiento de la información.

Dicho comportamiento neuronal se replica en redes neuronales convolucionales[3] mediante capas ocultas representadas en forma de matrices. Estas capas contienen información relevante que se conecta entre sus propias celdas similar a las neuronas de activación. Mediante una operación llamada convolución se pueden extraer características importantes de las capas ocultas.

Esta operación se encarga de crear una siguiente capa inicializada con datos provenientes del producto entre la capa original y una matriz de pesos llamada *kernel*. Pueden existir varias capas ocultas para aplicar convoluciones y en diferentes dimensiones según sea la información con la que se esté trabajando.

También existe otra capa muy utilizada en este modelo llamada *pooling*. Funciona de la misma manera que una operación convolucional a diferencia de que su propósito no es aprendizaje sino mejorar lo aprendido. Esto se logra reduciendo la matriz producto de una operación convolucional donde se encuentran las características de las previas capas. Durante este proceso no se trabaja con un *kernel* sino que se opera mediante, en la mayoría de casos, el promedio de los valores de una submatriz de la capa oculta. Finalmente, la optimización de estos modelos así como su entrenamiento es utilizando métodos como la gradiente descendiente y *backpropagation*. Se usan estas técnicas para encontrar la gradiente de la función costo con respecto al *kernel*.

3.3 Autoencoders

Las redes neuronales pueden aplicarse en conjunto para hacer modelos generativos de información como los *Autoencoders*.Este modelo se caracteriza por tener el número de neuronas de entrada igual al número de neuronas de salida. Por lo tanto, en el caso de una imagen como dato de entrada la salida que produce debe ser la misma imagen de entrada o muy similar. El mecanismo de regenerar datos de entrada para tener una salida igual al dato original se produce a través de dos redes neuronales llamadas: *encoder* y *decoder*. La primera parte empieza con una dimensión igual a la dimensión de los datos de entrada y a través de las capas ocultas completamente conectadas entre sí se va reduciendo las dimensiones hasta que en la capa de salida se obtiene un vector muy compacto. Dicho vector reducido se utiliza como dato de entrada para el *decoder*. El cuál hace un proceso inverso a la primera red neuronal obteniendo un tensor con las mismas dimensiones del original. El modelo se entrena calculando el error de la función costo usando la imagen original con la imagen producida por el *decoder*, luego se ajusta los pesos de las capas correspondientes para reducir el error de la función costo mediante una gradiente descendiente similar a la manera que se trabaja en las redes neuronales.

4 PROPUESTA

El método trabajo que se plantea para poder resolver la agrupación de textos cortos sin categorizar es a través del uso de *word embeddings* generadas por un modelo entrenado como BERT para representar de mejor manera los textos en vectores poco esparsos y de baja dimensionalidad. A su vez, se consigue mejor representación semántica de los textos gracias al mecanismo de atención[9] que utilizan los transformers como BERT. Sin embargo, el modelo BERT no entrega una *word embeddings* que represente toda la palabra. La representación de una palabra es mediante *substrings* y cada una de ellas tiene su propio *embedding word*. Este aspecto se toma en cuenta para poder representar un texto como el promedio total de cada *word embeddings* y su subtexto[1]. Como consecuencia de esto la representación de oraciones cortas requiere que se combine de la mejor forma las *embedding words* aumentando su costo computacional. El modelo SBERT[8] nos ofrece una herrramienta para la representación de textos cortos más eficiente manteniendo una afinidad con *word embeddings* de BERT sin perder similitud. Por lo tanto, usaremos esa técnica para la representación de nuestro *dataset* de textos cortos.

Luego de obtener la representación de los textos cortos mediante *word embeddings* se pasan como datos de entrada a un autoencoder basado en redes neuronales convolucionales. Con el objetivo de aplicar operaciones convolucionales mediante filtros para sacar las características profundas de los datos. Esta parte se basa en la efectiva clasificación de imágenes que hacen las redes neuronales convolucionales. Por lo tanto, se pretende hacer lo mismo con los textos cortos para obtener un representación oculta en medio de las capas del *autoencoder*.

La última parte del proceso es juntar todas las representaciones ocultas de los textos cortos en un nuevo archivo. Luego son utilizadas por un algoritmo de *clustering* como K-means para agrupar estos vectores.

En la figura 1, se puede observar el modelo BERT(amarillo) el inicio del sistema que se propone y que retorna un vector característica de una oración de nuestro *dataset*. Este vector(rojo) ingresa al *autoencoder*(celeste) como dato de entrada a la red neuronal que va ajustando los pesos de las capas ocultas hasta producir la representación de los datos de forma más compacta(verde). Por último, las representaciones son agrupadas mediante un K-means implementado en MrJob.

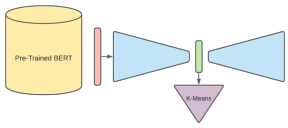

Fig. 1. Arquitectura del modelo propuesto

Una comparación interesante es utilizar *word embeddings* de otra fuente como Fasttext[2] que esta basado en técnicas como BoW. Donde podemos armar un matriz compuesta por filas que representan las palabras y el número de columnas igual al tamaño del *word embedding* correspondiente. Finalmente, se utilizaría el mismo esquema propuesto un *autoencoder* y un K-means para comparar la precisión de los resultados.

5 RESULTADOS PREELIMINARES

Nuestro conjunto de datos para la experimentación proviene de una recolección de datos en inglés realizado por Zhang[13] en su trabajo

Número de oraciones	Tiempo en segundos
1000	3.52
10000	11.42
100000	90.11
1000000	1038.18

de análisis de clasificación de textos. Para esta tarea el autor recolectó de diversas fuentes tales como Quora, Amazon y Yahoo. Nuestros resultados preeliminares usaron el *dataset* de Yahoo-Respuestas donde se encuentran un millón de respuesta por los usuarios sobre 10 tipos de temas.

Durante esta fase de preprocesamiento de datos seleccionamos oraciones que cumplan con el rango de entre 2 a 500 caracteres. Para luego empezar a codificar las oraciones mediante la librería mencionada previamente[8]. Hemos calculado el tiempo de codificación para este trabajo donde observarmos que el algoritmo usado se comporta de manera exponencial.

Con los textos cortos representados mediante *word embedding* de tamaño 384 podemos representarlas como imágenes de 16x24 pixeles. La figura 2 y 3 representan imágenes de los textos codificados en *sentences embeddings* de *"Fédération Internationale de Football Association"* y *"REALMADRID 4 SURE Barcelona is just good those days i want u 2 b sure that their days will b over soon"* respectivamente.

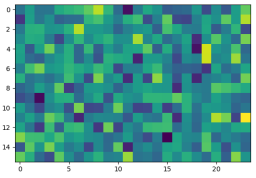

Fig. 2. Representación de una oración de 3 palabras con la etiqueta "deporte"

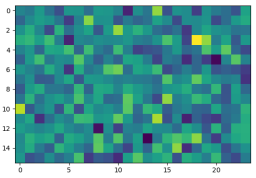

Fig. 3. Representación de una oración de 21 palabras con la etiqueta "deporte"

5.1 Experimentación

Nuestro primer experimento utilizó las *sentence embeddings* provenientes del *dataset* de Yahoo-respuestas[13]. Seleccionamos 5 grupos de 10 oraciones cada una con su respectiva *sentence embeddings* para formar parte del *dataset*. Los grupos son de los siguientes temas: Sociedad y Cultura, Ciencia y Matemáticas, Saluds, Educación y Computadoras. Agrupamos estos datos mediante un Kmeans de la librería *sklearn*. Para medir la precisión del algoritmo con las *sentence embeddings* calculamos la inercia de los datos correctamente clasificados y obtuvimos una suma de distancia euclideanas de las muestras a su centroide más cercano de 45.15 y el Kmeans de *sklearn* indicó una suma de 41.25.

Nuestro segundo experimento utilizó las *sentences embeddings* producidas por el *dataset* de Quora. Seleccionamos un total de 210 oraciones. En este conjunto de datos existen los siguientes temas: Economía(82 oraciones), Idioma Inglés (103 oraciones) y Contraseñas (25 oraciones). Para esta oportunidad implementamos un Kmeans utilizando PySpark consiguiendo una precisión del 100%.

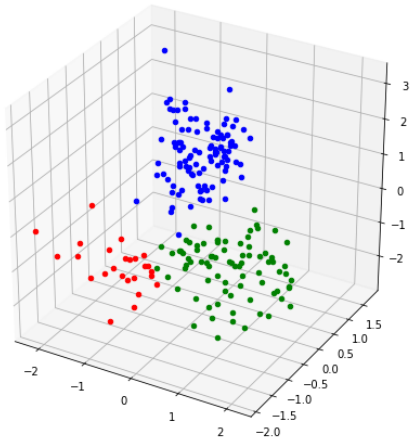

Fig. 4. Resultados de nuestro Kmeans implementado mediante Apache Spark para el segundo experimento.

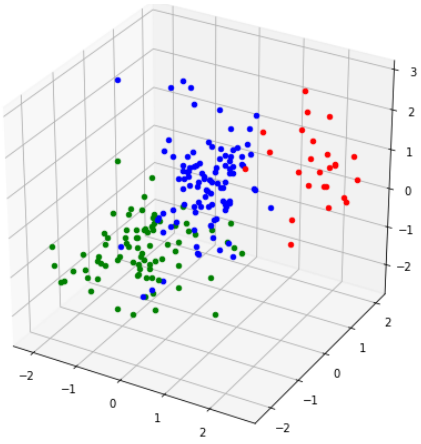

Fig. 5. Resultados de nuestro Kmeans implementado mediante Apache Spark para el segundo experimento utilizando diferentes dimensiones de las *sentences embeddings*.

6 PASOS FUTUROS

Para futuras experimentaciones queda pendiente integrar el autoencoder basado en redes convolucionales usando las *sentences embeddings* representadas en forma de imágenes como entrada. Luego obtener el vector latente entre las dos redes. Para finalmente aplicar Kmeans y analizar los resultados finales.

Hemos notado también la librería *SBERT* da buenas representaciones con textos cortos que no utilicen contracciones del idioma inglés, ni tampoco *emojis* ni lenguaje informal. Sin embargo, para textos más formales, como los de Quora, el rendimiento es óptimo.

REFERENCES

[1] Devlin, Jacob ; Chang, Ming-Wei ; Lee, Kenton ; Toutanova, Kristina: Bert: Pre-training of deep bidirectional transformers for language understanding. In: *arXiv preprint arXiv:1810.04805* (2018)

[2] Joulin, Armand ; Grave, Edouard ; Bojanowski, Piotr ; Douze, Matthijs ; Jégou, Hérve ; Mikolov, Tomas: FastText.zip: Compressing text classification models. In: *arXiv preprint arXiv:1612.03651* (2016)

[3] Krizhevsky, Alex ; Sutskever, Ilya ; Hinton, Geoffrey E.: Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems* 25 (2012), S. 1097–1105

[4] Mikolov, Tomas ; Chen, Kai ; Corrado, Greg ; Dean, Jeffrey: Efficient estimation of word representations in vector space. In: *arXiv preprint arXiv:1301.3781* (2013)

[5] Mikolov, Tomas ; Sutskever, Ilya ; Chen, Kai ; Corrado, Greg S. ; Dean, Jeff: Distributed representations of words and phrases and their compositionality. In: *Advances in neural information processing systems*, 2013, S. 3111–3119

[6] Pennington, Jeffrey ; Socher, Richard ; Manning, Christopher D.: Glove: Global vectors for word representation. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, S. 1532–1543

[7] Rakib, Md Rashadul H. ; Zeh, Norbert ; Jankowska, Magdalena ; Milios, Evangelos: Enhancement of short text clustering by iterative classification. In: *International Conference on Applications of Natural Language to Information Systems* Springer (Veranst.), 2020, S. 105–117

[8] Reimers, Nils ; Gurevych, Iryna: Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 11 2019. – URL https://arxiv.org/abs/1908.10084

[9] Vaswani, Ashish ; Shazeer, Noam ; Parmar, Niki ; Uszkoreit, Jakob ; Jones, Llion ; Gomez, Aidan N. ; Kaiser, Łukasz ; Polosukhin, Illia: Attention is all you need. In: *Advances in neural information processing systems*, 2017, S. 5998–6008

[10] Wang, Shiping ; Cai, Jinyu ; Lin, Qihao ; Guo, Wenzhong: An overview of unsupervised deep feature representation for text categorization. In: *IEEE Transactions on Computational Social Systems* 6 (2019), Nr. 3, S. 504–517

[11] Xu, Jiaming ; Xu, Bo ; Wang, Peng ; Zheng, Suncong ; Tian, Guanhua ; Zhao, Jun: Self-taught convolutional neural networks for short text clustering. In: *Neural Networks* 88 (2017), S. 22–31

[12] Yin, Hui ; Song, Xiangyu ; Yang, Shuiqiao ; Huang, Guangyan ; Li, Jianxin: Representation Learning for Short Text Clustering. In: *arXiv preprint arXiv:2109.09894* (2021)

[13] Zhang, Xiang ; Zhao, Junbo ; LeCun, Yann: Character-level convolutional networks for text classification. In: *Advances in neural information processing systems* 28 (2015), S. 649–657