IMPACTO DEL TAMAÑO DE SECUENCIAS DE TEXTO EN EL RENDIMIENTO DE MODELOS DE APRENDIZAJE PROFUNDO PARA CLASIFICACIÓN DE SENTIMIENTO.

PIÑEIRO A., YÉPEZ F., GUERRA A. & LUNA A.

RESUMEN

La investigación tiene como objetivo desarrollar un modelo de aprendizaje profundo que clasifique los sentimientos de reseñas de películas entre positivas y negativas. Se utilizó una base de datos pública que contiene las reseñas en texto plano y la clasificación de cada reseña, a favor o en contra. Durante el proceso de modelado de la solución se realizó una comparación entre distintos procesos de embedding y distintos modelos para evaluar la diferencia obtenida respecto a la precisión de cada uno. Nuestra investigación revela que el tamaño de cada reseña de película tiene un impacto en la exactitud de los modelos de aprendizaje profundo así como la decisión de entrenar o utilizar vectores pre-entrenados para la capa de embedding.

DESCRIPCIÓN DEL PROBLEMA

El tamaño de las cadenas de texto puede tener un impacto significativo en la eficiencia computacional de los modelos de aprendizaje automático [17]. Si las cadenas son demasiado largas, aumenta la carga computacional necesaria para procesarlas y almacenarlas. Por otro lado, si las cadenas son demasiado cortas, se puede perder información importante. Una investigación sobre el tamaño ideal nos ayuda a determinar la longitud óptima que maximice la eficiencia computacional sin comprometer el rendimiento del modelo. Para determinar el tamaño ideal decidimos compara tres tamaños diferentes de cadenas de texto con dos tipos de embedding (uno propio y el Word2Vec de Google) y tres tipos de modelos (DNN, CNN, LSTM).

METODOLOGÍA

Preprocesamiento de Datos

La base de datos cuenta con información suficiente para las pruebas que realizamos. Sin embargo, fue importante preparar los datos de mejor manera antes de comenzar a construir el modelo. Con el objetivo de que el entrenamiento sea más rápido y mejorar la efectividad del modelo, los pasos del pre-procesamiento de datos son los siguientes:

- Eliminar ruido (tags de HTML, símbolos).
- Eliminar stopwords.
- Eliminar puntuación.
- Eliminar cadenas de texto con un carácter.
- Eliminar múltiples espacios en blanco.
- Convertir mayúsculas a minúsculas.

Hiper parámetros para cada modelo

Para cada tipo de red neuronal usamos diferentes hiperparámetros. Los hiperparámetros de la capa de Embedding son los únicos que cambian para las pruebas. Por el lado de los modelos los parámetros se quedan fijos en todas la pruebas y fueron seleccionados con base en recomendaciones de articularlos de investigación actuales. Los modelos que generaron fueron los siguientes.

- Deep Neural Network (DNN)
- Convolutional Neural Network (CNN)
- Long Short Term Memory (LSTM)

Parámetros para embedding

En nuestra investigación realizamos el embedding de nuestro dataset utilizando un método propio y utilizando transfer learning con el modelo de Google Word2Vec. Los dos estilos de embedding usaron la misma configuración para poder ser comparados:

- *Tamaño de Vocabulario*: 22,175 equivalente al 10% de tokens de todo nuestro data set.
- *Tamaño de Dimensiones*: 300D debido a que es el mismo que utiliza el Word2Vec de Google y de esta froma se podra realizar una compración.
- *Tamaño de cadenas de texto*: Esta variable es la que probamos en nuestra investigación y hablaremos de esta en el siguiente inciso.

# PRUEBA	TAMAÑO DE RESEÑAS	TIPO DE EMBEDDING	MODELO
01	89 palabras	Embedding Propio	DNN
02		Word2Vec	
03	120 palabras	Embedding Propio	
04		Word2Vec	
05	145 palabras	Embedding Propio	
06		Word2Vec	
07	89 palabras	Embedding Propio	CNN
08		Word2Vec	
09	120 palabras	Embedding Propio	
10		Word2Vec	
11	145 palabras	Embedding Propio	
12		Word2Vec	
13	89 palabras	Embedding Propio	LSTM
14		Word2Vec	
15	120 palabras	Embedding Propio	
16		Word2Vec	
17	145 palabras	Embedding Propio	
18		Word2Vec	

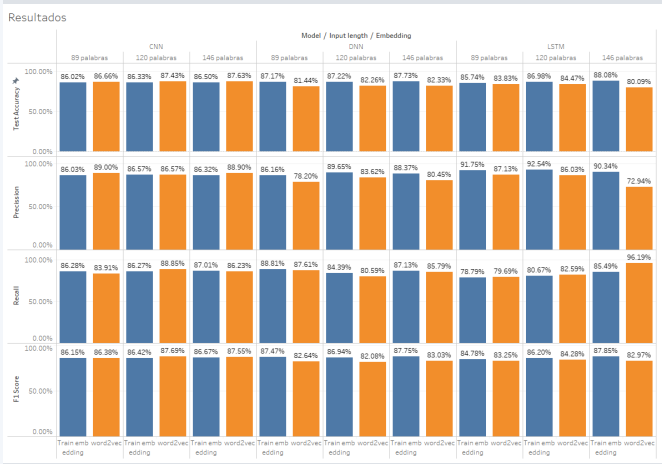
Pruebas

Se decidió utilizar tres versiones diferentes de nuestro dataset para alimentar a nuestras dos versiones de embedding y posteriormente a nuestros tres modelos de aprendizaje automático. Las tres versiones del dataset varían en la cantidad máxima de palabras que tienen:

- *89 palabras*: la mediana del dataset.
- *120 palabras*: el promedio del dataset.
- *145 palabras*: el tercer cuartil del dataset.

RESULTADOS

Evaluamos cada una de las 18 pruebas definidas anteriormente con las siguientes métricas de evaluación: Accuracy, Precision, Recall, F1-score.



- Según incrementa el tamaño de secuencia de texto usada, mejor rendimiento en modelos de aprendizaje profundo (DNN, CNN, LSTM).
- Accuracy y F1-score más altos: LSTM con longitud de cadena 146 palabras entrenando capa de embedding. Accuracy = 88.08%
- Mejor precisión: 92.54% LSTM, cadenas longitud 120 palabras entrenando embedding propio.
- Recall más alto: 96.19% LSTM cadenas longitud 146 palabras usando embedding de Word2Vec.

CONCLUSIÓN

Finalmente, comprobamos con este artículo que la longitud de entrada que usamos para los modelos si tiene efecto en el rendimiento del mismo. Para los 3 modelos y los 2 tipos de embedding resultó ser así ya que entre más grande fue la longitud de las reseñas, mejores fueron los resultados de accuracy obtenidos. Usar los datos del 3er Cuartil tuvo un buen rendimiento, aunque sería de utilidad realizar un análisis posterior en el que se pruebe usando el máximo de palabras como longitud de las reseñas.

REFERENCIAS

[1] Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. Ain Shams Engineering Journal. Doi:10.1016/j.asej.2014.04.011.
[2] Jassim, M. A., Abd, D. H., & Omri, M. N. (2023). Machine learning-based new approach to films review. Social Network Analysis and Mining. Doi:10.1007/s13278-023-01042-7.
[3] Alayba, A. M., Palade, V., England, M., & Iqbal, R. (2018). A combined CNN and LSTM model for arabic sentiment analysis. Doi:10.1007/978-3-319-99740-7_12.
[4] Kaur, G., & Sharma, A. (2023). A deep learning-based model using hybrid feature extraction approach for consumer sentiment analysis. doi:10.1186/s40537-022-00680-6.
[5] Kim, Y. (2014). Convolutional neural networks for sentence classification. doi:10.3115/v1/d14-1181.
[6] Liu, Y., Bi, J., & Fan, Z. (2017) Multi-class sentiment classification: The experimental comparisons of feature selection and machine learning algorithms. doi:10.1016/j.eswa.2017.03.042.
[7] Benrouba, F., & Boudour, R. (2023). Emotional sentiment analysis of social media content for mental health safety. Social Network Analysis and Mining, 13(1) doi:10.1007/s13278-022-01000-9.
[8] Fan, X. (2023). Artificial intelligence technology-based semantic sentiment analysis on network public opinion texts. doi:10.4018/IJITSA.318447.
[9] Taherdoost, H., & Madanchian, M. (2023). Artificial intelligence and sentiment analysis: A review in competitive research. doi:10.3390/computers12020037.
[10] Ullah, K., Rashad, A., Khan, M., Ghadi, Y., Aljuaid, H., & Nawaz, Z. (2022). A deep neural network-based approach for sentiment analysis of movie reviews. Doi:10.1155/2022/5217491.
[11] A Cunha, M. Costa, and M. Pacheco, "Sentiment analysis of youtube video comments using deep neural networks," International Conference on Artificial Intelligence, 2019.
[12] Srivastava, N. (2013). Improving neural networks with dropout. University of Toronto, 182(566), 7.
[13] S. Seo, C. Kim, H. Kim, K. Mo and P. Kang, "Comparative Study of Deep Learning-Based Sentiment Classification," doi: 10.1109/AC-CESS.2019.2963426.
[14] Kingma, D.P., & Ba, J. (2015). Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations (ICLR).
[15] Song, F., Li, Y., Cheng, W., & Dong, L. (2023). An improved dynamic programming tracking-before-detection algorithm based on LSTM network. doi:10.1186/s13634-023-01020-3
[16] An, J. H., Wang, Z., & Joe, I. (2023). A CNN based automatic vulnerability detection. Eurasip Journal on Wireless Communications and Networking. doi:10.1186/s13638-023-02255-2
[17] Mikolov, T., Chen, K., Corrado, G., Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. arXivpreprint arXiv:1301.3781.