Universidad Nacional Autónoma de México
Facultad de Ciencias

Reconocimiento de patrones y aprendizaje automatizado

Proyecto Final

Hernández Norberto Dana Berenice 317163027
Martínez Calzada Diego 318275457

1. Introducción

La información compartida por medio de la expresión escrita es un medio de comunicación muy importante que a lo largo del tiempo se ha ido digitalizando, requiriendo ser compactado para lograr obtener un mejor entendimiento de parte de los seres humanos para con el mismo. Para ello, se han desarrollado una gran cantidad de recursos como los resumenes o sintesis, al ser de forma digital y masiva se convierte en un trabajo que requiere automatizarse y estandarizarse.
Así mismo identificación de temas en cualquier tipo de expresión oral o escrita nos permite formarnos una idea general de la información que podemos encontrar en el mismo, de igual forma se pueden asociar ideas generales a cada uno de estos temas. Así las aplicaciones de procesar un discurso y realizar clasificarlas pueden ser vastas, ya que esta clase de datos se encuentran desde artículos escritos, libros, canciones, podcast, conversaciones telefónicas y más, esto ayudaría a generar nueva información o contenido que sirva como una representación más concisa de ella. De esta manera se han creado diversas metodologías que nos han permitido identificar el tema y secuencia de la información escrita, entre ellas se encuentra el aprendizaje automatizado conjunto con el llamado procesamiento de lenguaje natural.

2. Objetivo

Este proyecto tiene como objetivo el crear un algoritmo que nos permita realizar una clasificación de noticias, extracción de subtemas y con ello generar recomendaciones de lectura a los usuarios. Es un objetivo interesante debido al interés que pueden llegar a tener los usuarios en tal o cual tema, de esta manera podemos obtener mejores resultados en cuanto al flujo de información.

3. Método y materiales:

Para la implementación de un modelo que nos permita realizar la asignación de subtemas y clasificación de la noticias seleccionamos una base de datos de noticias recolectadas por Kishan Yadav de la aplicación inshorts, con articulos de tecnología, deportes, noticias mundiales, politica, entretenimiento, automoviles y ciencia, con un total de 12120 noticias clasificadas.

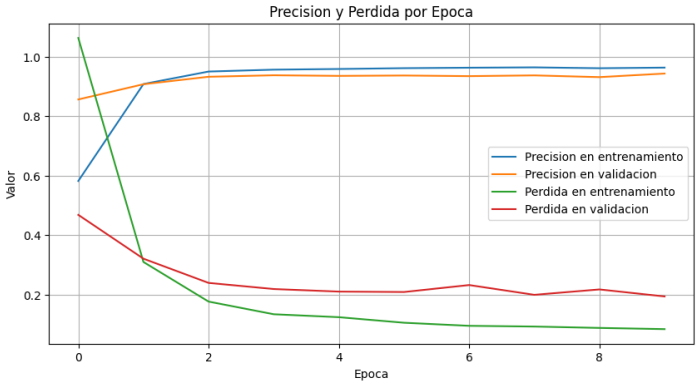
- **Procesamiento de texto:** Para poder analizar textos con un modelo de aprendizaje de maquina primero es necesario procesarlo, para ello tenemos que eliminar caracteres especiales, eliminar espacios en blanco, signos de puntuación, urls y convertir todas las letras correspondientes al texto a letras minusculas. De la misma manera, es necesario segmentar el texto con el que queremos alimentar a nuestro modelo, esto lo realizaremos por medio de la tokenización, nombre que hace referencia la proceso de dividir un texto en pequeños fragmentos del mismo, es posible realizarlo por frases, palabras o simplemente por una cantidad determinada de fragmentos de texto.

- **Clasificación de noticias:** Para clasificar las noticias de nuestra base de datos utilizamos un modelo de aprendizaje supervisado, que nos permita determinar el tema al que hace referencia el texto, para ello utilizamos la biblioteca keras de python. Posteriormente pasamos a convertir estos fragmentos de texto a números, rellenando con ceros a los fragmentos que no alcancen la longitud establecida, con el fin de que todos los fragmentos y secuencias de números sean ahora de la misma longitud.Ahora que tenemos los textos en un formato que sean legibles para un modelo de aprendizaje no supervisado, implementamos la arquitectura de la red.
  - **Capa de entrada:** En esta capa de la red es de donde ingresan los datos de los cuales se alimenta, definimos una cantidad de 100 neuronas para esta capa.
  - **Capa de embedding:** Los embeddings son una técnica utilizada en procesamiento de lenguaje natural que nos permite convertir lenguaje humano a vectores matematicos, transformando los índices de tal forma que palabras con un significado similar, se encuentren dimensionalmente cerca, es decir, estos vectores capturan relaciones semanticas entre palabras, permitiendonos obtener mejores resultados en nuestro objetivo.
  - **Capa de LSTM:** Las siglas LSTM hacen referencia a su nombre en inglés "Long Short-Term Memory", es decir "Memoria de largo a corto plazo". Es un tipo de red neuronal recurrente, utilizada de frecuentemente en análisis de procesamiento de lenguaje natural, series de tiempo y reconocimiento de patrones. Les redes LSTM tienen la capacidad de retener información durante secuencias largas por medio de celdas de memoria, en el caso del procesamiento de lenguaje resulta importante el uso de la misma debido a que es común encontrarnos palabras iguales que pueden ser utilizadas en diferentes contextos, para ello es necesario revisar palabras anteriores a la palabra de estudio, es decir, es necesario revisar el contexto en el que se presenta el mismo. Las LSTM tienen una arquitectura definida por tres compuertas las cuales nos permiten regular el flujo de información dentro y fuera de la celda de memoria:
    - **Célula de memoria:** Es la estructura que nos permite mantener la información por largos periodos de tiempo.
    - **Compuerta de entrada:** Controla el flujo de información que retiene la celula, compuesta tipicamente por una capa de activación sigmoide que decide qué valores actualizamos, y una capa tanh, la cual crea un vector de nuevos valores candidatos para agregar al estado de la celda.
    - **Compuerta de olvido:** Determina qué información descartar del estado de la celula. Toma información (paso de tiempo actual y estado oculto anterior) y produce un número entre 0 y 1 para cada número en el estado de la celda. Si obtiene un número 1, la información se conserva, si obtiene un 0, la información se desecha.
    - **Compuerta de salida:** Decide qué parte de la información en la célula de memoria se usará para la salida, o si se mantiene oculto. Determina el siguiente estado oculto en función del estado de la celda actualizado, además de filtrar la información que generará el LSTM en función del estado actualizado de la celda.
  - **Capa de attention:** La atención es un mecanismo que permite a la red neuronal enfocarse en las partes más relevantes de un texto, en este caso, asignandole un score por medio del producto vectorial, para posteriormente aplicar la función softmax, esta nos permite obtener los valores escalados de los score obtenidos y por lo tanto, obtener la combinación lineal de esos valores.
  - **Capas de salida:** Es la capa en la que se clasifica el tema del texto con el que se alimenta la red neuronal, en este caso puede tomar un valor de 0 a 6, dependiendo del índice al que hace referencia. Nuestra primer capa de salida promedia los vectores que regresa la capa de attention (regresa un vector por palabra), así esta capa regresa un solo vector. La segunda capa reduce dimensiones y con una función ReLU quita la linealidad quitando valores cercanos iguales o menores a cero. La última capa toma el vector de salida de la capa anterior y usando su matriz de pesos lo transforma en un vector con 7 valores, uno por categoría, finalmente usando una función softmax transforma cada valor a una probabilidad para cada categoría.
- **Generación de recomendaciones:** Realizamos una función que nos permite tomar un conjunto aleatorio de noticias de otra base da datos, con las cuales la red no ha tenido interacción, de esta forma, dependiendo de la categoría favorita del usuario, regresará las noticias que pueden ser de su interés.

- **Reconocimiento de subtemas:** En este sección del proyecto aplicamos un algoritmo de K-means con el fin de extraer los subtemas de un texto dado. En el contexto de las noticias, resulta útil al momento de requerir una parte de la información en particular, en lugar de tener que leer toda la noticia. Utilizando las funciones anteriormente descritas, como la limpieza y segmentación de texto, utilizando un total de 5 clusters.

4. Resultados:

Para la clasificación de noticias por tema, utilizamos la red neuronal anteriormente descrita, entrenandola con un total de 10 épocas, permitiendonos obtener una precisión de 0.96 para el conjunto de entrenamiento y de 0.94 para el conjunto de prueba. Además de una error de 0.06 para el conjunto de entrenamiento y de 0.19 para el conjunto de validación.



Para la recomendación, logramos generar recomendaciones correctas para los usuarios, logrando de alguna manera nuestro objetivo de mejorar el flujo de información con respecto a los intereses de las personas. De la misma manera, generamos la asignación de subtemas para el texto que se ingrese al algortimo dado, de esta manera, como mencionamos anteriormente, facilita la compresión del texto por parte de los lectores.

5. Conclusiones:

Los resultados que obtuvimos en todos los algoritmos fueron favorables, permitiendonos llegar a nuestro objetivo. De la misma manera, permitiendonos explorar un tema que aún se encuentra siendo estudiado como es el aprendizaje de máquina, el uso de transformadores en redes neuronales y el llamado procesamiento de lenguaje natural.

6. Bibliografia:

- Murzone, F. (2020, 5 de octubre). Procesamiento de Lenguaje Natural. Preprocesado del texto 1: Tokenización. EscuelaDeInteligenciaArtificial. https://medium.com/escueladeinteligenciaartificial/procesamiento-de-texto-para-nlp-1-tokenizacion-4d533f3f6c9b
- Espíndola, G. (2023, 3 de Marzo). ¿Qué son los embeddings y cómo se utilizan en la inteligencia artificial con python? Medium. https://gustavo-espindola.medium.com/que-son-los-embeddings-y-como-se-utilizan-en-la-inteligencia-artificial-con-python-45b751ed86a5
- Hamad, R. (2023, 3 de Diciembre). What is LSTM? Introduction to Long Short-Term Memory. Medium. https://medium.com/@rebeen.jaff/what-is-lstm-introduction-to-long-short-term-memory-66bd3855b9ce
- Gautam, H. (2020, 1 de Marzo) Word Embedding: Basics. Medium. https://medium.com/@hari4om/word-embedding-d816f643140
- Arriola, V. [ArriolaRios] (2023, 27 de Marzo) 2020 12 08 22 13 53 4 Funcionamiento de una LSTM Vol. Youtube. https://youtu.be/FpZWHy1V5uw?si=SKY_iBgvageKgdWL
- Adaloglou, N. (2020, 13 de Septiembre) How Attention works in Deep Learning: understanding the attention mechanism in sequence models. AI Summer. https://theaisummer.com/attention/
- Simth, B. (2024, 9 de Febrero) Self-Attention Explained with Code. Medium. https://towardsdatascience.com/conte transformer-embeddings-using-self-attention-explained-with-diagrams-and-python-code-d7a9f0f4d94e
- Yadav, K. (2021, 21 de Enero). News Classification. Kaggle. https://www.kaggle.com/datasets/kishanyadav/inshort news?select=inshort_news_data-1.csv
- Arunima. (2020, 22 de Junio). Using LSTM Embedding Layer. Kaggle. https://www.kaggle.com/code/arunima24/u lstm-embedding-layer
- Rvk. (2019, 30 de Marzo). LSTM Attention Keras Kaggle. https://www.kaggle.com/code/rahulvks/lstm-attention-kerasBuilding-a-model
- Janakiev, N. (s.f.) Practical Text Classification With Python and Keras. Real Python. https://realpython.com/pyth keras-text-classification/