Puntos clave para la presentación del *poster*

- Presentación
    - Buenas noches, compañeros, soy Ma. Luisa Argaez y el día de hoy les expondré del análisis de sentimientos aplicando aprendizaje profundo
- Introducción
    - Contexto
        - Hoy en día la generación de datos ha crecido exponencialmente gracias al internet de las cosas que se encarga de conectarse e intercambiar datos con otros dispositivos y sistemas a través de internet, sin embargo, también es posible intercambiar datos utilizando otros medios.
        - La gran cantidad de datos que se recaban es importante analizarlos y obtener beneficios a través de su recolección, almacenamiento, preprocesamiento, y análisis para poder tomar mejores decisiones
        - Entre los diferentes tipos de datos que se pueden obtener se encuentran los datos relacionales y no relacionales, de este último uno muy popular es el texto y para analizarlo eficazmente, las técnicas del NLP nos sirven mucho.
        - Uno de los casos donde se puede utilizar el NLP para analizar el texto, es en el análisis de sentimientos, el cual tiene como objetivo la evaluación de emociones, actitudes y opiniones mediante el análisis de texto.
        - Entre las múltiples formas de texto, como son los libros, noticias, reseñas, entre otros. Especificamente de las reseñas que expresan opiniones de algún objeto o situación.
        - Amazon, vende millones de productos que tienen reseñas y calificaciones. En este proyecto se trabaja con una muestra de 1 millón de reseñas con su etiqueta de si es buena 1 o mala 0.
    - Problemática
        - Actualmente las empresas encuentran importante estudiar las reseñas de los productos y servicios que venden, pero al ser tantos los datos no es posible hacerlo de una manera rápida y eficaz
    - Objetivo
        - Principal – general: Limpiar, preprocesar y analizar las reseñas para obtener una exactitud de al menos del 70% en el conjunto de entrenamiento.
        - Secundarios:
            - Limpiar la BD removiendo todos aquellos caracteres que no sean alfanuméricos, pasar todo a minúsculas y eliminar *stopwords*
            - Realizar dos preprocesamientos, *stemming* y lematización para analizar cual preprocesamiento brinda mejores resultados
            - Realizar una LSTM que no se sobreajuste.
    - Modelo LSTM
        - Es un tipo de RNN, una arquitectura de una NN usada en DL.
        - Se destaca por capturar dependencias de largo plazo haciéndola ideal para tareas de predicción de secuencias, como el texto.
        - Una de sus fortalezas es que resuelve el problema del desvanecimiento del gradiente, el cual es con cada iteracion de la propagación hacia atrás, el gradiente se vuelve cada vez más pequeño hasta ser casi cero.
- Metodología
    - Análisis exploratorio de datos
        - Se analiza una comparación entre la cantidad de palabras de antes y después de eliminar las stopwords. En el primero, se observa la cantidad de palabras, en el segundo la longitud de la reseña

y en el tercero la media de las palabras. Es importante ver que la media en todos los casos disminuye haciendo el análisis mas consistente.

- Limpieza de datos
  - Primero se convierte todo a minúsculas, después se eliminan todos los caracteres que no son digito, letra o espacio en blanco y después se quitaron las stopwords
- Procesamiento de datos
  - Se establece que todas las secuencias de texto tendrán una longitud máxima de 10 palabras.
  - Se creo un tokenizador con la capacidad de manejar hasta 100 palabras distintas. Se especifica el número máximo de palabras a mantener según la frecuencia de palabras. En este caso, se están considerando las 100 palabras más frecuentes en el conjunto de datos.
  - El tokenizador se ajusta al conjunto de textos `X`. Esto implica que el tokenizer analiza el texto y construye un índice de palabras basado en la frecuencia de ocurrencia de cada palabra en el conjunto de datos.
  - Después se transforma cada texto en una secuencia de números enteros. Cada palabra en el texto se asigna a un número entero según el índice construido por el tokenizer durante el paso anterior.
  - Se aplica relleno (padding) a las secuencias para que todas tengan la misma longitud. Todas las secuencias se rellenarán o truncarán para tener una longitud de 10. Si una secuencia es más corta que 10, se rellenará con ceros al principio, y si es más larga, se truncará.
- Modelo LSTM
  - El modelo se construyó utilizando la librería de “Keras. en Python. El modelo tuvo la siguiente arquitectura: primero, se agregó una capa de incrustación (“Embedding”), posteriormente se apilaron dos capas “LSTM” de 50 y 10 unidades cada una. Después se agregó una capa de abandono (“Dropout”) con una tasa del 50 %. Luego se agregó una capa de ´ 8 neuronas y finalmente, una capa de salida sigmoidal con una neurona. El modelo se compilo utilizando el optimizador “Adam”, la función de ´ perdida “Entropía cruzada binaria” (“Binary Crossentropy”), se evaluó el rendimiento del modelo con ´ la métrica de exactitud. El modelo se entrenó durante 20 ´ épocas con un lote de 32 muestras con datos ´ de entrenamiento y se monitoreo con la métrica de pérdida y exactitud con un conjunto de validación. ´ Se utilizo una técnica que consiste en si la pérdida en el conjunto de validación no mejora durante 3 ´ épocas consecutivas, la tasa de aprendizaje se reduce a la mitad (3) (4). Esta metodología se realizó dos veces, una donde los datos eran preprocesados utilizando “Stemming” y con lematización.

- Resultados
  - Lematizacion
    - Se logro una exactitud en el conjunto de entrenamiento y prueba de 75.64 uy 73.66% respectivamente
  - Stemming
    - Se logro una exactitud en el conjunto de entrenamiento y prueba de 76 uy 74.34% respectivamente
  - Ambos preprocesamientos lograron resultados similares, sin embargo, el de stemming fue mejor por 1.3% en exactitud. En la función de perdida se observa que a partir de la época 10 la perdida en el conjunto de prueba ya no decreció, si no que ya oscilo entre 0.51 y 0.52.
- Conclusiones
  - Se logro alcanzar los objetivos generales y secundarios satisfactoriamente. En este caso, el procesamiento de datos a la raíz resulto como mejor metodología. También fue posible observar que al aumentar el tamaño del diccionario en la tokenizacion mejorar los resultados.
- Trabajo a futuro
  - Hacer este ejercicio con mayor poder de cómputo para incluir los 4 millones de muestras y aumentar el tamaño del diccionario. Incluir ingeniería de características para el texto para reducirlo más.