Maestría en Inteligencia Artificial Aplicada / MNA / Tecnológico de Monterrey
Actividad en Equipos - Semana 5:
Vectores Embebidos
Caso Amazon/Yelp/IMDb

Escuela de Ingeniería y Ciencias

Dr. Luis Eduardo Falcón Morales

Nombre(s): _____

Matrícula(s): _____

En la actividad de esta semana trabajarás en equipos con el modelo de vectores embebidos de OpenAI.
Los actuales LLM pueden generar vectores embebidos de cada token, palabra o enunciado que les proporciones, el coste computacional y de uso de recursos hará la diferencia en el uso de cada uno de ellos.

Una primera manera de trabajar con estos modelos pre-entrenados, es generando un vocabulario a partir de tu conjunto de datos de entrenamiento. Posteriormente, cada palabra de tu vocabulario se sutituye por su correspondiente vector continuo/embebido. De manera predeterminada cuando no existe el vector para una palabra en particular se elige el vector más cercano en similaridad, o bien, se puede eliminar dicha palabra. Existen diversas propuestas para utilizar dichos vectores embebidos como entrada para modelos de aprendizaje automático. Cuando se trabaja con los vectores embebidos de cada palabra/token, una manera de representar a dicho enunciado es sustituyendo el enunciado por el vector promedio de todos los vectores embebidos que lo forman.

En esta actividad vamos a comparar modelos de aprendizaje automático (machine learning) aplicando vectores embebidos a cada palabra, con relación a aplicarlos directamente a los comentarios dados.

1. Descarga los 3 archivos de Canvas y genera un solo DataFrame de Pandas con ellos. En particular, el archivo de datos de IMDb ya no requiere transformarse para obtener sus 1000 registros. Verifica que tienes los 3000 registros con sus respectivas etiquetas en dicho DataFrame. Los archivos los encuentras en Canvas y se llaman: amazon5.txt, imdb5.txt, yelp5.txt.

2. Realiza un proceso de limpieza. Aplica el preprocesamiento que consideres adecuado a todos los comentarios. Llama Xclean a los comentarios procesados y Y a las etiquetas.

3. Realiza una partición aleatoria con los mismos porcentajes de la práctica de ls semana pasada para poder comparar dichos resultados con los de esta actividad, a saber, 70%, 15% y 15%, para entrenamiento, validación y prueba, respectivamente. Verifica que obtienes 2100 registros de entrenamiento y 450 para cada uno de validación y prueba. Usa una semilla para la partición.

4. Construye tu vocabulario a continuación:
    a. Usa el conjunto de entrenamiento para generar tu vocabulario con un tamaño que consideres adecuado. Si lo deseas, puedes filtrar tu vocabulario por la frecuencia mínima de uso de cada palabra, así como por su longitud mínima en caracteres.

b. Indica el tamaño del vocabulario que generaste.
c. ¿Por qué debe usarse solamente el conjunto de entrenamiento para generar el vocabulario?
d. Con el vocabulario generado, filtra los conjuntos de entrenamiento, validación y prueba para que todos los comentarios usen solamente las palabras de este vocabulario.

Hasta este punto básicamente has realizado transformaciones muy análogas a las de la semana pasada y que son válidas para muchos de los procesos dentro del análisis de textos. Procedamos ahora con los vectores embebidos de cada palabra de cada comentario, en lugar de los vectores generados con las matrices Tf-idf.

5. En particular, OpenAI tiene los siguientes modelos de vectores embebidos. Realiza un resumen de las principales características de cada uno de estos modelos:
   - https://platform.openai.com/docs/guides/embeddings
   - https://platform.openai.com/docs/models/text-embedding-3-small
   - https://platform.openai.com/docs/models/text-embedding-3-large
   - https://platform.openai.com/docs/models/text-embedding-ada-002

6. Utiliza alguno de los modelos de OpenAI de vectores embebidos para generar un nuevo diccionario clave-valor (key-value), donde la "clave" será cada palabra de tu vocabulario y el "valor" será su vector embebido de dimensión dada por el modelo seleccionado. Es recomendable que una vez que generes el nuevo vocabulario de vectores embebidos guardes dicho diccionario en un archivo (pickle, npz o el formato que consideres más adecuado). Además, apóyense entre los miembros del equipo para que puedan trabajar con los diferentes modelos de vectores embebidos. Consideren los costos de cada modelo. Indica además la cantidad de tokens de OpenAI utilizados.

7. Una manera de utilizar los vectores embebidos con modelos de aprendizaje automático (machine learning), es asignar a cada comentario el vector embebido de dimensión predeterminada que resulta de promediar todos los vectores embebidos de cada una de sus palabras (tokens). Así, en este ejercicio deberás generar los conjuntos de entrenamiento, validación y prueba de esta manera. Los llamaremos trainEmb, valEmb y testEmb, respectivamente. Es decir, ahora cada comentario es un solo vector de dimensión dada por el modelo de OpenAI seleccionado.

8. Utilizando los nuevos conjuntos embebidos de entrenamiento y validación, obtener los modelos de regresión logística y bosque aleatorio (random forest). Para cada modelo muestra el valor de la exactitud (accuracy) y el reporte de sklearn dado por la función classification_report(). Verifica que no estén sobreentrenados y compara tus resultados con los que obtuviste en la actividad de la semana pasada. Puedes incluir algún otro modelo de machine learning si lo consideras adecuado.

9. Con el mejor modelo y el nuevo conjunto de prueba, obtener la mejor matriz de confusión y el classification_report() de sklearn.

10. Ahora, como segunda parte de esta actividad:
    a. Realiza la transformación a vectores embebidos de todos los 3000 comentarios tal como están dados en los archivos. Selecciona el modelo de vector embebido que consideres más adecuado. Indica la cantidad de tokens de OpenAI utilizados en el proceso.
    b. Realiza una partición en Train-Val-Test del 70%, 15% y 15%, respectivamente. Usa la misma semilla que utilizaste en el ejercicio 3, para la partición.
    c. Utiliza los modelos de regresión logística y bosque aleatorio (random forest) para este problema de clasificación. Para cada modelo muestra el valor de la exactitud (accuracy) y el reporte de sklearn dado por la función classification_report(). Verifica que no estén sobreentrenados y compara tus resultados con los que obtuviste en la primera parte. Puedes incluir algún otro modelo de machine learning si lo consideras adecuado.

11. Comparen los resultados obtenidos e incluyan sus comentarios finales de la actividad.