Universidad Nacional de Rosario

Facultad de Ciencias Exactas, Ingeniería y Agrimensura

Procesamiento del Lenguaje Natural

Trabajo Práctico Nº1 Parte 2

Rodríguez y Barros, Francisco: R-4559/4

Ejercicio 2

Introducción

Para esta parte del trabajo me enfoque en aplicar técnicas de búsqueda semántica sobre un texto, en este caso el manual del juego Cascadia.

Para esta parte del trabajo elegí el texto correspondiente al manual del juego https://github.com/franciscoryb1/NLP-2025/blob/dd00a3beed5318603c9b834dd63b6514e9bf35f8/Tp1%20Parte%202/datos/informaci%C3%B3n/cascadia_manual.txt
porque es un texto extenso con mucho contenido, útil para aplicar técnicas de búsqueda semántica.

Limpieza de datos

Inicialmente antes de desarrollar la resolución de la consigna incluí un paso previo dedicado a la limpieza del contenido del archivo cascadia_manual.txt porque tenía muchos caracteres, números, palabras, símbolos, etc. sueltos y generaban un conflicto al hacer los chunks para luego generar los embeddings. La limpieza del archivo la realice con un script

de python haciendo principalmente de la librería re para poder implementar el uso de expresiones regulares.

Archivo resultante:
https://github.com/franciscoryb1/NLP-2025/blob/909278e814eb3586002909e78a02ef189f5a8b9b/Tp1%20Parte%202/datos/informaci%C3%B3n/cascadia_manual_limpio.txt9

Implementación

Entre los diferentes modelos presentados en la unidad 2 como TF-IDF, Word2Vec, Doc2Vec y Sentence-BERT decidi por implementar el uso de **Sentece-BERT** porque es un modelo específicamente entrenado para comparar frases completas de forma semántica, también incluyendo entre múltiples idiomas. Se utilizó el modelo paraphrase-multilingual-MiniLM-L12-v2.

Resultados

Los resultados fueron bastante buenos. En consultas concretas como “qué pasa al final del juego” o “cómo se colocan las fichas de hábitat”, el modelo encontró respuestas claras y precisas. En otras más generales o que no tenían una respuesta tan literal en el texto las coincidencias fueron un poco más deficientes.

En conclusión, con **Sentence-BERT** logré aplicar una búsqueda semántica bastante efectiva sobre textos largos y complejos, incluso en otro idioma. Pero me quedo con que, si bien es una herramienta muy potente, considero que es muy importante una buena preparación previa de los textos a usar.

Ejercicio 3

Introducción

Para este ejercicio decidí utilizar nuevamente el texto cascadia_manual.txt porque es extenso, completo y, como se comentó anteriormente en el ejercicio 2, se le aplicó una limpieza de palabras sueltas, caracteres, líneas en blanco, etc.

Implementación

Lo primero que se hizo fue dividir todo el texto en fragmentos de 60 palabras. Este parámetro lo fui variando en un rango desde 40 a 100 palabras a lo largo de diferentes pruebas, idealmente encontré un balance en 60 palabras, resultando en que el fragmento tenga suficiente contenido y a la vez no sea muy extenso.

Con POS se identificaron los sustantivos de cada fragmento y luego se aplicó un reconocimiento de entidades nombradas (NER) para clasificar algunos de estos sustantivos dentro de categorías como lugares, productos, personas, etc.
Esto es muy útil para aportar al análisis semántico un plus a la hora de analizar los sustantivos ya que no todos los sustantivos tienen el mismo valor informativo dependiendo del contexto.

El siguiente paso fue representar cada fragmento usando los sustantivos extraídos mediante el uso de la técnica TF-IDF que nos permite convertir un texto en un vector numérico reflejando la importancia de cada palabra dentro de la frase y no únicamente cuantas veces aparece.

Finalmente calcule la similitud entre fragmentos usando tres técnicas diferentes: **distancia de coseno**, **Jaccard** y **Dice**, estas dos últimas se basan en si las palabras coinciden o no, sin importar cuántas veces aparecen ni cuán importantes son. Por otro lado, la distancia de coseno compara el ángulo entre dos vectores, y tiene en cuenta tanto la presencia como la importancia de cada término.

Resultados

Los resultados fueron bastantes similares con las tres técnicas aplicadas, especialmente en los fragmentos que repiten palabras o de contenidos casi idénticos.

En fragmentos que no son idénticos pero están conceptualmente relacionados, la distancia de coseno funciono un poco mejor porque pudo captar similitud incluso cuando el vocabulario no era exactamente el mismo, mientras que Jaccard y Dice se vieron un poco más limitadas.

Los resultados fueron en general buenos, pero considero que para tareas como esta donde buscamos encontrar similitud semántica entre textos que no necesariamente son exactamente iguales, la distancia del coseno es mejor opción porque permite captar mejor el contexto general del fragmento y no solo su coincidencia literal.

Ejercicio 4

Introducción

El objetivo de este ejercicio es detectar el idioma de los diferentes archivos de texto presentes en la carpeta “Información” de mi set de datos.

Implementación

Para hacer la detección de idiomas utilice la librería langdetect ideal para detectar idiomas en textos y que no requiere entrenamiento.

Fui recorriendo cada archivo de la carpeta “Información” de mi dataset, leyendo y detectando el idioma presente en cada uno para finalmente construir un dataframe de pandas con el nombre de cada archivo y su idioma detectado.

Resultados

En mi dataset todos los archivos de texto están en el idioma inglés por lo que el resultado obtenido es más que correcto.

	archivo	idioma_detectado
0	cascadia_manual.txt	en
1	whatboardgame_review.txt	en
2	oneboardfamily_review.txt	en
3	flatout_games.txt	en
4	boardgamereview_review.txt	en
5	board_game_co_uk_guide.txt	en
6	bluehighwaygames_description.txt	en
7	foro_reviews_cascadia.txt	en
8	cascadia_manual_limpio.txt	en

Ejercicio 5

Introducción

En este ejercicio trabajé con las reseñas de usuarios sobre el juego Cascadia que fueron extraídas del foro. El objetivo fue analizar el sentimiento de cada reseña y elaborar un sistema de búsqueda por similitud semántica que permita además filtrar por sentimiento.

Implementación

No utilice los archivos de reseñas presentes en la carpeta “Información” del set de datos debido a que, analizándolos, me encontré con que tenían muchas palabras sueltas, espacios en blanco, contenido como menúes de página, etc. que no hacían a las reseñas en específico.

Realice una nueva extracción de reseñas utilizando el script presente en el archivo https://github.com/franciscoryb1/NLP-2025/blob/dd00a3beed5318603c9b834dd63b6514e9bf35f8/Tp1%20Parte%202/Extraccion%20reviews%20cascadia.ipynb
el cual forma parte de mi primer entrega de este trabajo, simplemente cambie el link al del juego Cascadia y añadir una linea de codigo para que, al guardar cada reseña en el archivo .txt, se agregue un separador "===RESEÑA===\n" permitiendo posteriormente parsear de forma más eficiente cada reseña.

Aplique un análisis de sentimiento sobre cada reseña utilizando un modelo multilingüe y que ya sabe cómo interpretar frases y asignarles un valor sentimental de BERT porque mis textos están en el idioma inglés. Este modelo devuelve una escala de 1 a 5 estrellas.

Volqué todas las reseñas junto con su puntuación (en estrellas) y nivel de confianza en un dataframe de pandas para una mejor visualización de los resultados obtenidos.

Luego agrupe en tres categorías: negativo, neutro y positivo usando la función **clasificar_sentimiento**, que básicamente recibe la puntuación y devuelve su categoría

correspondiente, utilizando el criterio de que, 1 y 2 estrellas hacen a la categoría “negativo”, 3 estrellas, “neutro” y más de 4 estrellas “positivo”.

Una vez obtenidas las emociones asociadas a cada reseña, pase a construir el sistema de búsqueda. Utilice técnicas de similitud semántica mediante el uso de embeddings generados con el modelo all-MiniLM-L6-v2 de la librería sentence-transformers.

Tanto las reseñas como la consulta del usuario se transforman en vectores, y luego usó la distancia coseno para calcular sus similitudes.

Resultados

A pesar de que los valores de confianza al asignar una valoración de sentimiento no fueron altos, considero que sirvieron como base para realizar un filtro inicial de las reseñas y focalizar un poco mejor los resultados en relación al sentimiento buscado.

A partir de este filtro inicial el sistema de embeddings y búsqueda de similitudes funcionó de forma correcta buscando entre la primera selección de reseñas acorde al sentimiento especificado.

Ejercicio 6

Introducción

En este ejercicio el objetivo es construir un modelo para clasificar preguntas del usuario en tres categorías: información, estadísticas y relaciones. Teniendo como objetivo que el sistema pueda determinar de forma automática a qué fuente de datos debería ir a buscar la respuesta.

Implementación

Primero se generó un dataset con 300 preguntas y su respectiva categorización. Inclui 100 preguntas por cada categoría contemplando que mi set de datos se encuentre balanceado.

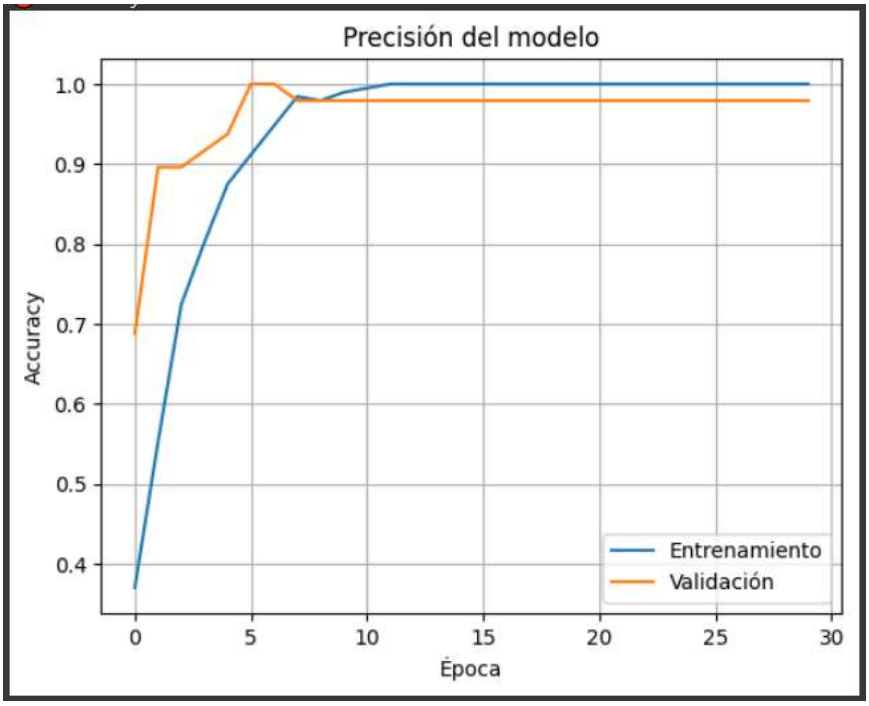
Dividí mi set de datos en X e y. En X se incluyeron los embeddings de todas las preguntas generados con TF-IDF, lo que me permite mantener la importancia de las palabras en la frase. Por otro lado en y se incluyeron las tres categorías (información, estadísticas y relaciones) pasadas a valor numérico (0, 1 y 2) respectivamente.

Se generaron los embedding para cada pregunta del set de datos utilizando TF-IDF

Se implementaron dos modelos diferentes, uno más simple y fácil de interpretar, Regresión Logística y por otro lado una Red Neuronal Densa, un modelo con una arquitectura más compleja.

Ambos se entrenaron con los mismos datos para poder compararlos de forma justa.

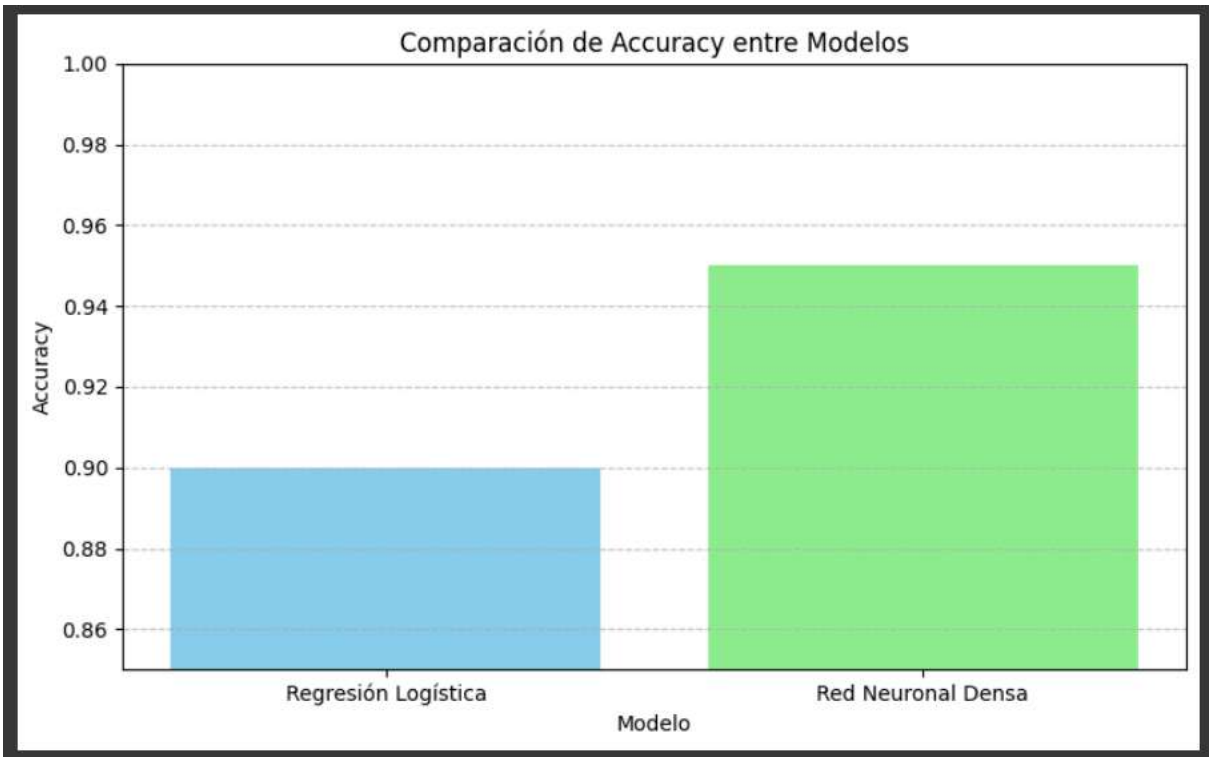
Logrando con la Regresión Logística un accuracy de 0.90, y con la Red Neuronal Densa un accuracy en train de 1 y en validación de 0,97, sin presencia de overfitting ni underfitting.



Resultados

Ambos modelos hicieron un buen trabajo, pero con algunas diferencias clave. La regresión logística logró un accuracy del 90%, un resultado muy bueno teniendo en cuenta que es un modelo simple y rápido de entrenar.

Por otro lado, la red neuronal densa alcanzó un 95% de accuracy, mostrando una mejor capacidad para captar relaciones más complejas entre las palabras y las categorías por su estructura más compleja, no lineal y múltiples capas.



La regresión logística ofrece un rendimiento muy bueno con una implementación más simple y eficiente.