Tecnicatura Universitaria en Inteligencia Artificial

Trabajo Práctico 2 NLP - Pradera

Autor: Caballero Franco

Asignatura: Procesamiento del Lenguaje Natural

Profesores: Juan Pablo Manson, Alan Geary, Constantino Ferrucci y Dolores Sollberger

Fecha de entrega: 21/05/2025

ÍNDICE

1. Resumen ..... pág. 2

2. Introducción ..... pág. 3-4

3. Metodología ..... pág. 4-7

4. Resultados ..... pág. 8-12

5. Conclusiones ..... pág. 12-13

6. Referencias ..... pág. 13-14

7. Anexos ..... pág. 14

RESUMEN

En esta segunda parte del trabajo práctico, se continuó con el análisis de datos obtenidos mediante web scraping de un sitio web de juegos de mesa, en particular centrado en el juego **Pradera**, previamente estructurado en un repositorio con carpetas organizadas por juego y temática. A partir de los textos extensos almacenados en la sección de información, se aplicaron técnicas de procesamiento de lenguaje natural (NLP) para su fragmentación y posterior vectorización utilizando modelos de embeddings semánticos estudiados en clase. Se realizaron análisis de similitud textual comparando diferentes métricas de distancia, evaluando su desempeño y seleccionando aquella que resultó más adecuada para búsquedas semánticas.

En otro de los ejercicios, se trabajó con técnicas de extracción de sustantivos (POS tagging) y reconocimiento de entidades nombradas (NER), combinadas con búsquedas semánticas restringidas, evaluando nuevamente distintas métricas de similitud. Posteriormente, se implementó un sistema de detección automática de idioma, clasificando y organizando los textos por lenguaje en un dataframe.

En el análisis de reseñas de usuarios, se aplicó un modelo pre-entrenado de análisis de sentimientos, integrando los resultados a un sistema de recuperación de información basado en similitud semántica con opción de filtrado por sentimiento. Finalmente, se construyó un dataset de más de 300 consultas categorizadas según su correspondencia con las fuentes de datos: estadísticas, información o relaciones. Se entrenó un modelo de regresión logística para predecir la categoría a partir de la consulta vectorizada, evaluando su rendimiento mediante distintas métricas.

INTRODUCCIÓN

En la actualidad, el volumen de información disponible en línea exige el desarrollo de herramientas que permitan organizar, procesar y extraer valor de grandes cantidades de texto no estructurado. El procesamiento de lenguaje natural (NLP) y los modelos semánticos han demostrado ser útiles para abordar estos desafíos, permitiendo realizar análisis automatizados que faciliten la recuperación de conocimiento relevante a partir de textos complejos.

Este trabajo se enmarca en ese contexto, como una continuación de una primera etapa donde se aplicaron técnicas de *web scraping* para extraer información del juego de mesa *Pradera*. A partir de esa base, se busca aplicar técnicas de NLP para profundizar en el análisis semántico de los textos recopilados, con el fin de explorar similitudes entre fragmentos, identificar entidades clave, clasificar información y facilitar búsquedas por contenido y por sentimiento.

La elección de este enfoque permite no sólo ejercitar conceptos teóricos abordados en clases sino también poner en práctica herramientas modernas que son ampliamente utilizadas en la industria, como modelos de *embeddings*, clasificación automática y visualización de datos. Asimismo, se justifica el uso de un único juego como caso de estudio, ya que permite concentrar los esfuerzos analíticos en una fuente acotada, pero rica en contenido textual y relaciones semánticas.

Además, el análisis semántico y la estructuración de la información realizada en este trabajo pueden servir como base para desarrollar en el futuro un sistema de búsqueda avanzada o incluso un chatbot específico para el juego *Pradera*. Este chatbot permitiría a los usuarios consultar directamente sobre aspectos del juego y obtener respuestas precisas basadas en la información extraída y procesada, facilitando así el acceso a conocimiento detallado y personalizado sin necesidad de revisar manualmente grandes volúmenes de texto.

Objetivos específicos

- Fragmentar y vectorizar textos extensos extraídos del juego *Pradera* utilizando modelos de embeddings semánticos.

- Comparar distintas técnicas de medición de similitud textual para identificar la más adecuada en búsquedas semánticas.

- Realizar extracción de sustantivos y reconocimiento de entidades nombradas para enriquecer la búsqueda semántica.

- Implementar un sistema de detección automática de idioma para clasificar los textos según su lengua.

- Aplicar análisis de sentimientos a reseñas de usuarios y desarrollar un sistema de búsqueda que permita filtrar resultados según el sentimiento.

- Construir y categorizar un conjunto de consultas frecuentes, entrenar un modelo de clasificación para predecir la categoría de consulta a partir del texto ingresado.

- Evaluar y comparar diferentes modelos y métricas para optimizar la precisión y utilidad de las búsquedas semánticas realizadas.

Breve descripción de la estructura del informe

El informe se organiza en varias secciones para facilitar la comprensión del trabajo realizado. Tras esta introducción, se presenta la metodología empleada, donde se detallan las fuentes de datos, técnicas de procesamiento y herramientas utilizadas. A continuación, en la sección de desarrollo o implementación, se describen los procedimientos aplicados para fragmentar, vectorizar y analizar los textos, junto con explicaciones de los algoritmos y modelos usados.

La sección de resultados muestra los principales hallazgos, apoyados en gráficos, tablas y análisis críticos que permiten evaluar el desempeño de las distintas técnicas de similitud y clasificación. Finalmente, se presentan las conclusiones, que resumen los aportes y limitaciones del trabajo, y se sugieren posibles líneas para futuras investigaciones o aplicaciones prácticas. Además, se incluyen referencias bibliográficas y anexos con material complementario.

METODOLOGÍA

Fuentes de datos:
Se utilizaron los textos extensos extraídos mediante web scraping del sitio web BoardGameGeek - Meadow. Estos textos se encuentran organizados en la carpeta "Información" del repositorio, en formato .txt, y constituyen la base para los análisis semánticos realizados.

Técnicas y métodos aplicados
Se aplicaron técnicas de procesamiento de lenguaje natural (NLP) para fragmentar los textos en partes manejables, las cuales luego fueron vectorizadas usando modelos de embeddings semánticos estudiados en clase. Para este trabajo, se eligió el modelo **DistilUSE-base-multilingual-cased-v1** de Sentence-Transformers. La elección de este modelo se basó en varias razones:

- Su capacidad **multilingüe**, ideal para textos en español e inglés.
- Está basado en **DistilBERT**, lo que proporciona eficiencia en memoria y tiempo sin comprometer la calidad de los embeddings.
- Fue pre-entrenado específicamente para tareas de **similitud semántica**, lo que lo hace especialmente adecuado para comparar fragmentos de texto extraídos de documentos largos.

Para medir la similitud entre fragmentos, se compararon varias métricas: Jaccard, Levenshtein, Dice, Jaro-Winkler y similitud del coseno. Se seleccionó la similitud del coseno debido a:

- Su **invarianza respecto a la longitud** de los vectores, lo cual es crucial para textos de diferente extensión.
- Su efectividad en **espacios de alta dimensión**, donde otras métricas como la distancia euclidiana pueden perder significado.
- Su **eficiencia computacional** y claridad interpretativa, al entregar valores entre 0 y 1 que indican el grado de similitud semántica.

En el caso específico del análisis de listas de sustantivos extraídos mediante *POS tagging*, los términos fueron previamente **lematizados** para reducir las variaciones morfológicas. Aunque métricas como Jaccard o Dice funcionan con coincidencias exactas, su desempeño se ve limitado si no hay coincidencias literales. Por ello, se seleccionó la **distancia Jaro-Winkler**, que permite comparar cada sustantivo con todos los de la lista opuesta, capturando similitudes fonéticas y morfológicas, incluso ante leves diferencias de escritura

Se implementó también, un sistema de detección automática de idioma con la librería **langdetect**, clasificando y organizando los textos por idioma en un dataframe para facilitar análisis posteriores.

Para el análisis de sentimientos sobre reseñas de usuarios, se utilizó un modelo preentrenado de HuggingFace: **nlptown/bert-base-multilingual-uncased-sentiment**. Este modelo fue elegido por:

- Estar **preentrenado específicamente en reseñas**, lo cual mejora su precisión para este tipo de texto.
- Su capacidad **multilingüe**, útil ante reseñas que incluyen fragmentos en inglés y otros idiomas.
- Su base en BERT, lo que garantiza un rendimiento robusto en clasificación de sentimientos.

En la creación del sistema de **búsqueda semántica filtrada por sentimiento**, se optó por un modelo **asimétrico**, adecuado para escenarios en los que se cuenta con consultas cortas

(como preguntas o frases clave) que deben buscarse en fragmentos más largos del corpus textual.

Clasificación de consultas

Se construyó un dataset con más de **300 consultas**, categorizadas en tres clases: **“Relación”, “Estadística” e “Información”**. Estas consultas fueron vectorizadas usando **TF-IDF**, una técnica útil para resaltar los términos más representativos y distintivos de cada categoría. Esto facilita la clasificación ya que TF-IDF asigna más peso a esos términos más distintivos. Algunos ejemplos de términos frecuentes por categoría incluyen:

Categoría	Palabras comunes
Relación	"autor", "editorial", "colaborado", "diseñó", "ilustró", "comparten"
Estadística	"puntaje", "ranking", "cantidad", "jugado", "popularidad", "reseñas"
Información	"tablero", "reglas", "componentes", "instrucciones", "objetivo", "duración"

Posteriormente, utilicé **LabelEncoder** para codificar las categorías porque es una opción simple y eficiente en este caso. Al tratarse de pocas clases, evita aumentar la dimensionalidad innecesariamente, como ocurriría con **OneHotEncoder**.

Elegí **Regresión Logística** como modelo de clasificación porque es un modelo lineal simple, eficiente y adecuado para problemas de clasificación multiclase con un conjunto de datos pequeño y bien definido. Ofrece interpretabilidad, buen rendimiento en tareas linealmente separables y no requiere gran poder de cómputo ni ajustes complejos.

Herramientas y tecnologías

El desarrollo se realizó principalmente en **Python**, utilizando las siguientes librerías especializadas:

- **spaCy** para procesamiento lingüístico y *POS tagging*.

- **sentence-transformers** para generación de embeddings.
- **scikit-learn** para clasificación, vectorización y métricas.
- **pandas** para manipulación de datos.
- **langdetect** para detección de idioma.
- **transformers** de HuggingFace para el modelo de análisis de sentimientos.

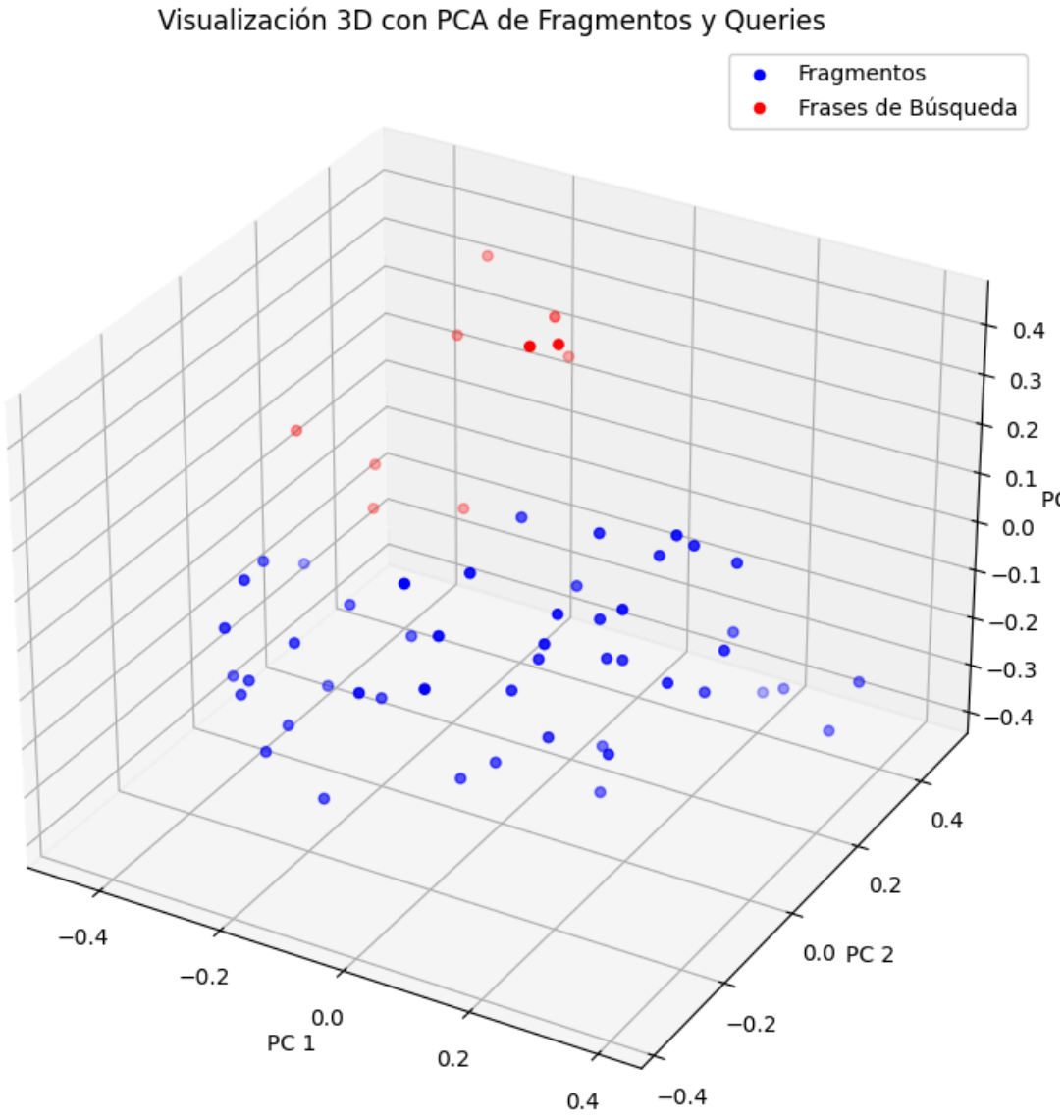
Procedimiento

- Recolección y organización de textos extraídos en el repositorio.
- Fragmentación y vectorización de textos con el modelo DistilUSE.
- Comparación y selección de técnicas de similitud (coseno para texto, Jaro-Winkler para listas de sustantivos).
- Extracción y categorización de sustantivos (POS tagging) y reconocimiento de entidades nombradas (NER).
- Detección automática de idioma y clasificación.
- Análisis de sentimientos y desarrollo del sistema de búsqueda filtrado.
- Creación, vectorización y clasificación supervisada de consultas.
- Evaluación y comparación de resultados para seleccionar las mejores técnicas y modelos.

RESULTADOS

Búsqueda por similitud: Query y Fragmentos

A continuación, se muestra el gráfico 3D obtenido con PCA, donde las *queries* y los fragmentos se distribuyen en el espacio de acuerdo a su similitud semántica



Además, se calcularon diversas métricas de similitud (*coseno, Jaccard, Levenshtein, Dice, Jaro-Winkler*) entre las *queries* y los fragmentos, lo que permitió comparar el desempeño de cada una de ellas. A continuación, se presentan algunos de los resultados obtenidos para estas métricas.

Similitud Coseno	Distancia Jaccard	Distancia Levenshtein	Índice Dice	Similitud Jaro-Winkler
-0.0262	0.9787	265.0000	0.0417	0.2200
0.1754	0.9348	234.0000	0.1224	0.2500
0.0517	0.9538	387.0000	0.0882	0.1600
0.2175	0.9583	249.0000	0.0800	0.2300
0.0038	1.0000	39.0000	0.0000	0.3000
0.1932	0.9706	156.0000	0.0571	0.2400
0.3120	0.9744	1041.0000	0.0500	0.0800
0.2185	0.9333	110.0000	0.1250	0.3300
0.2825	0.9595	625.0000	0.0779	0.1200
0.3514	0.9853	426.0000	0.0290	0.1600
0.3206	0.9688	427.0000	0.0606	0.1600
0.2814	0.9649	356.0000	0.0678	0.1800
0.2395	0.9487	191.0000	0.0976	0.2600
0.1858	0.9722	154.0000	0.0541	0.2700

Extracción y Comparación de sustantivos

Para enriquecer la búsqueda semántica, se realizó la extracción de sustantivos a partir de los fragmentos de texto, utilizando técnicas de **POS tagging**. A continuación, se muestra una muestra de los sustantivos extraídos.

```
print(set(sustantivos_por_fragmento[1]))
✓ 0.0s
{'ilustración', 'mesa', 'responsable', 'mundo', 'juego', '2021', 'incursión', 'versión', 'polaco'}
```

También, se calcularon las similitudes entre los sustantivos extraídos de diferentes fragmentos, utilizando las mismas métricas de similitud mencionadas previamente. Los resultados de estas comparaciones son los siguientes:

```
Similitud Jaccard: 0.0
Similitud Dice: 0.0
Similitud Coseno (BoW): 0.0
Similitud Levenshtein: 0.2895316804407713
Similitud Jaro-Winkler: 0.48727272727272725
```

Detección de Idioma

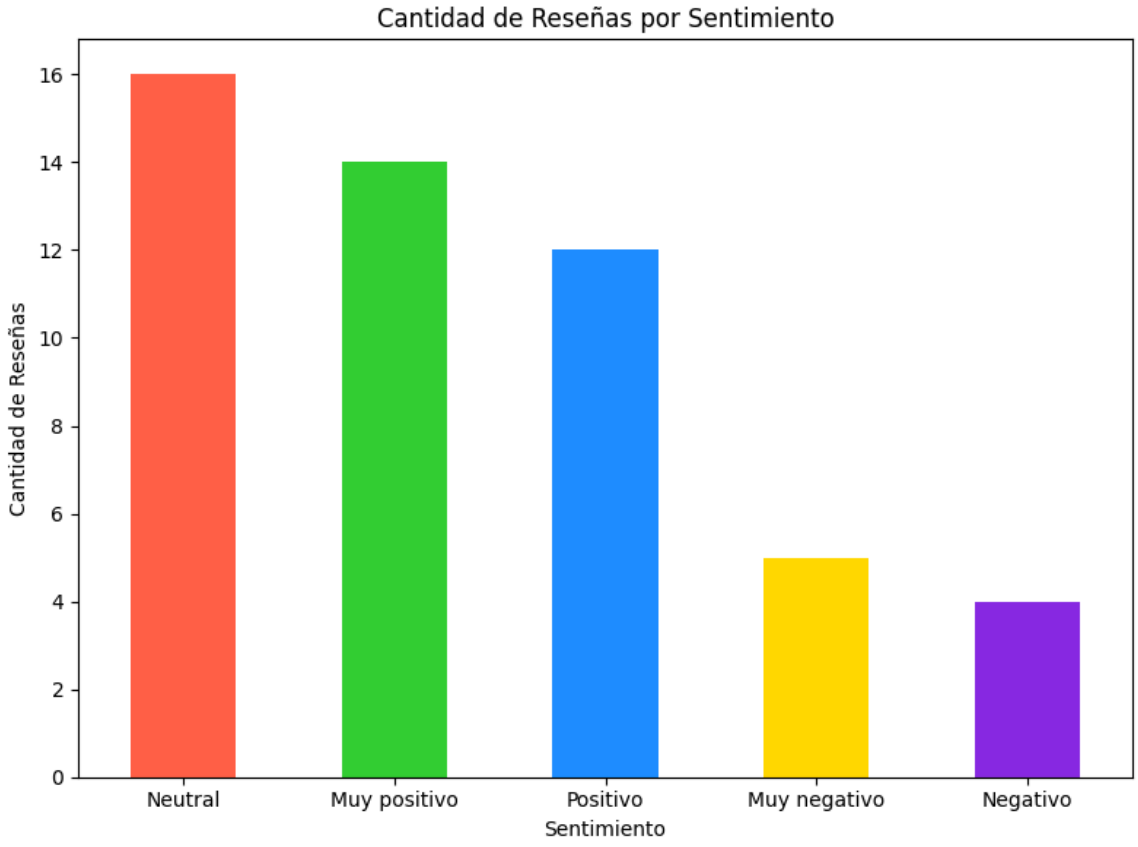
El sistema de detección automática de idioma fue aplicado a los fragmentos de texto, utilizando la librería **langdetect**. Este proceso permitió identificar y clasificar los textos en su idioma correspondiente (español, inglés, etc.).

A continuación, se presenta una visualización de las primeras filas del dataframe resultante, donde se puede observar la detección del idioma junto con una muestra de los textos.

	archivo	idioma	texto
0	comentarios.txt	es	1 Espectacular. Desde que se anunció ya me lla...
1	descripcion_general.txt	en	Meadow is an engaging set collection game with...
2	enlaces_imagenes.txt	fr	https://i0.wp.com/misutmeeple.com/wp-content/u...
3	enlaces_pagina.txt	es	#Colecciones\nhttps://misutmeeple.com/tag/cole...
4	foro_general.txt	en	Missed card packs ?: Hello everyone\n\nAfter c...

Análisis de sentimientos

El modelo **nlptown/bert-base-multilingual-uncased-sentiment** se aplicó sobre más de 50 reseñas de usuarios. Las predicciones fueron mapeadas a una escala de 1 (muy negativa) a 5 (muy positiva). En el siguiente gráfico se observa la cantidad de reseñas por sentimiento:



Se observa que las reseñas con sentimientos 'Muy negativas' y 'Negativas' son las menos frecuentes, mientras que las categorías 'Neutral', 'Muy positivo' y 'Positivo' representan una mayor proporción de las reseñas

Además, se presentan algunas de las reseñas y su clasificación correspondiente, como muestra de cómo el modelo asignó los sentimientos.

	texto	Sentimiento
0	Espectacular. Desde que se anunció ya me llamó...	Muy positivo
1	A mi me tiene encandilado. No deja de ser un j...	Muy positivo
2	Gran reseña, como siempre. Está gustando mucho...	Positivo
3	A mi siempre me deja con ganas de seguir jugan...	Positivo
4	Hola Iván, la verdad es que me has sorprendido...	Neutral

Clasificación de consultas

Se utilizó Regresión Logística para clasificar las consultas en tres categorías: “Estadística”, “Información” y “Relación”. El modelo fue entrenado sobre datos vectorizados con TF-IDF y las etiquetas codificadas mediante **LabelEncodeer**. Los resultados obtenidos se resumen en la siguiente imagen:

	precision	recall	f1-score	support
Estadística	0.90	0.60	0.72	15
Información	0.80	0.93	0.86	30
Relación	0.94	0.94	0.94	17
accuracy			0.85	62
macro avg	0.88	0.82	0.84	62
weighted avg	0.86	0.85	0.85	62

El modelo mostró un rendimiento alto en general, con especial desempeño en la categoría **Relación**, tanto en precisión como en recall. La clase **Estadística** presentó el menor recall, lo cual sugiere que el modelo tiene más dificultad para identificar correctamente este tipo de consultas, posiblemente por su menor representación en el conjunto de datos y su cercanía semántica con otras clases. Aun así, el F1-score de 0.72 es aceptable y se puede mejorar con más datos o afinando las características.

En la siguiente imagen se puede ver como el modelo clasifica consultas en las categorías correspondientes:

```python
def predecir_categoria(consulta_nueva):
    vector = vectorizador.transform([consulta_nueva])
    pred = modelo.predict(vector)
    return codificador.inverse_transform(pred)[0]

# Ejemplo
print(predecir_categoria("¿Quién ilustró el juego Meadow?"))
print(predecir_categoria("¿Cuál es el puntaje promedio del juego?"))
print(predecir_categoria("¿Qué tipo de tablero utiliza Pradera?"))
```

```
Relación
Estadística
Información
```

CONCLUSIONES

Algunos de los resultados principales fueron:

1. Similitud semántica: La similitud del coseno fue la métrica más efectiva para la comparación de fragmentos de texto, permitiendo un análisis semántico preciso.

2. Extracción de POS y NER: Las técnicas de extracción de sustantivos y entidades nombradas enriquecieron el proceso de búsqueda, con buenos resultados al utilizar la distancia Jaro-Winkler para comparar sustantivos.

3. Detección de Idiomas: El sistema de detección automática de idiomas, identificó correctamente el idioma de los textos en la mayoría de los casos. Sin embargo, se presentaron algunas dificultades al procesar los archivos que contenían enlaces web. En estos casos, la presencia de URLs en los fragmentos de texto causó ciertos errores en la detección del idioma, lo que generó resultados incorrectos para algunos fragmentos. A pesar de estos inconvenientes, el modelo en general mostró una alta precisión en la identificación del idioma y permitió organizar eficientemente los textos en sus respectivos grupos lingüísticos.

4.Análisis de sentimientos: El modelo de análisis de sentimientos mostró que las reseñas Muy negativas y Negativas son menos frecuentes, mientras que Neutral, Muy positivo y

Positivo dominan. Esto refleja una mayor satisfacción general de los usuarios con el juego de mesa.

5. Clasificación de consultas: El modelo de regresión logística logró un buen rendimiento en la clasificación de consultas en las categorías “Estadística”, “Información” y “Relación”, destacando la categoría Relación. La categoría Estadística mostró un menor recall, lo que podría mejorarse con más datos o ajustes en el modelo.

Cumplimiento de los objetivos

Se cumplieron los objetivos propuestos, implementando correctamente las técnicas de clasificación, análisis de sentimientos y búsqueda semántica. El trabajo logró aplicar herramientas de NLP en un contexto real, con buenos resultados a pesar de la limitación de los datos.

Recomendaciones para futuros trabajos

- **Ampliar el conjunto de datos** para mejorar el desempeño, especialmente en categorías menos representadas.
- **Explorar modelos más complejos** para mejorar la clasificación y análisis de sentimientos.
- **Optimizar la búsqueda semántica** utilizando modelos como **BERT** para un análisis más contextual.

En resumen, el trabajo proporciona una base sólida para futuras mejoras en el análisis de textos y la clasificación de datos en el ámbito de NLP.

REFERENCIAS

SOBRE CÓDIGO:

Hugging Face, Inc. (2020). Transformers: State-of-the-art Natural Language Processing for Pytorch and TensorFlow 2.0. *https://huggingface.co/transformers/*

Sennrich, R., Haddow, B., & Birch, A. (2016). Neural Machine Translation of Rare Words with Subword Units. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, 1715-1725. https://aclanthology.org/P16-1162/

Bojanowski, P., Grave, E., Mikolov, T., et al. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5, 135-146. https://doi.org/10.1162/tacl_a_00051

Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is All You Need. *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS 2017)*, 6000-6010. https://arxiv.org/abs/1706.03762

Langdetect, (2025). LangDetect: Language Detection Library. *https://pypi.org/project/langdetect/*

SpaCy, (2025). SpaCy Documentation. *https://spacy.io/*

Sentence-Transformers, (2025). Sentence-Transformers: State-of-the-art Sentence Embeddings. *https://www.sbert.net/*

PAGINA USADA PARA EL WEB SCRAPING

BoardGameGeek. (n.d.). https://boardgamegeek.com/boardgame/314491/meadow

ANEXOS

El código fuente completo, los datos brutos y la documentación adicional están disponibles en el siguiente repositorio de GitHub:

Repositorio en GitHub - Proyecto NLP