

Facultad de Ciencias Exactas, Ingeniería y Agrimensura

# Tecnicatura en Inteligencia Artificial

## Procesamiento de Lenguaje Natural

TRABAJO PRÁCTICO 1 - Parte 2

Oviedo María Victoria

**EJERCICIO 1**

Se carga el repositorio asignado (Tiny Towns).

Información sobre el repositorio:  
Github: [https://github.com/Augusto-Rabbia/NLP\_TP1/](https://github.com/Augusto-Rabbia/NLP_TP1/tree/main)

De los créditos obtuvimos los distintos nombres alternativos del juego, año de lanzamiento, los mecanismos que incluye el juego y su descripción, los tipo de familia de juego en los que pertenece este juego y su descripción, y también obtuvimos las descripciones de los diseñadores, los publicadores, los artistas y los juegos en los que están asociados.

Link: <https://boardgamegeek.com/boardgame/265736/tiny-towns/credits>

Para obtener reglas, variantes, estrategia y críticas del juego, hicimos web scraping de los foros de BGG, iterando por los 10 posts con más rating en las respectivas secciones de los foros. De esta forma, asumimos que obtendremos la información de mayor calidad que se puede encontrar en los foros para un usuario.

Link: <https://boardgamegeek.com/boardgame/265736/tiny-towns>

Para datos tabulares obtuvimos distintos valores numéricos que figuran en la página de boardgamegeek.com correspondiente al juego. Incluye valores como la cantidad de jugadores apropiada, la duración esperada de una partida, la cantidad de aficionados al juego, precios a los que se puede obtener en distintas tiendas, etc.

Link: <https://boardgamegeek.com/boardgame/265736/tiny-towns/stats>

En cuanto a vídeos, utilizamos algunos de los vídeos que figuran en la página de boardgamegeek.com, y obtuvimos sus transcripciones. Estos vídeos incluyen reviews y tutoriales, en idiomas como el inglés, el español y el neerlandés.

Links : [<https://www.youtube.com/watch?v=Ko6waVAt1CU>

<https://www.youtube.com/watch?v=WGhZ83B4MoQ>

<https://www.youtube.com/watch?v=yF--uYWp0zw>

<https://www.youtube.com/watch?v=OyBCApqyUHg>]

Obtuvimos también texto extrayendolo de un documento en formato PDF. Este documento es el manual oficial del juego, en inglés, e incluye información como las reglas, las piezas que vienen con el juego, algún trasfondo del universo en el que se ambienta y los roles que cumplen los distintos jugadores.

Link: <https://www.alderac.com/wp-content/uploads/2025/03/TinyTowns_Rulebook.pdf>

Además extrajimos de una página web una extensa reseña en español que puede proporcionar interesantes justificaciones a sobre las opiniones de las distintas mecánicas del juego, y cuenta con una estructura dividida en secciones y subsecciones.

Link: <https://misutmeeple.com/2020/03/resena-tiny-towns/>

**EJERCICIO 2**

Para este ejercicio se cargaron de la sección de Información los textos que estaban en español ([review\_bgg.txt](https://github.com/Augusto-Rabbia/NLP_TP1/blob/main/datos/informacion/review_bgg.txt), [review\_externa.txt](https://github.com/Augusto-Rabbia/NLP_TP1/blob/main/datos/informacion/review_externa.txt)) y se concatenaron en un solo texto para que sea más extenso.

Se hace una **limpieza** ligera de los datos ya que luego vamos a utilizar S-BERT y al eliminar puntuación o normalizar demasiado, se corre el riesgo de quitarle contexto al modelo.

Para la **segmentación** se separa el texto por saltos de línea dobles, lo que permite identificar bloques temáticamente coherentes. Luego, se aplica una segmentación "consciente del contenido" que conserva la estructura semántica de manera eficiente y mantiene el contexto. Finalmente, se agrupan las oraciones para evitar fragmentos demasiado cortos o ambiguos, lo cual mejora la calidad de los embeddings generados por modelos como S-BERT.

Luego se verifican algunos fragmentos y se verifican la cantidad de tokens por fragmento.

Rango de Tokens Cantidad de Fragmentos

0–10 0

11–20 5

21–30 11

31–40 20

41–50 16

51–60 11

61–80 20

81–100 15

101–128 8

129+ 4

Para analizar la similitud semántica entre diferentes fragmentos de texto, se **vectorizó** con Sentence-BERT (SBERT).

A partir de estos embeddings, se calculó la similitud entre todos los pares de fragmentos usando la métrica de similitud coseno, que mide el ángulo entre los vectores en el espacio de características. Se identificaron los pares con mayor similitud para destacar las oraciones que comparten significados o temas más cercanos.

Los resultados se presentan ordenados en una tabla con los diez pares más similares, mostrando sus textos originales y la puntuación de similitud coseno correspondiente.

### **Análisis de similitud de texto:**

Se definieron un conjunto de consultas representativas para evaluar la capacidad del modelo de identificar fragmentos relevantes dentro del corpus de texto. Cada consulta se transformó en un embedding vectorial utilizando el modelo Sentence-BERT. Posteriormente, se calculó la similitud coseno entre el embedding de cada consulta y los embeddings de todos los fragmentos, lo que permitió medir qué tan cercanos están semánticamente. Para cada consulta, se seleccionaron los dos fragmentos con mayor puntuación de similitud.

Para realizar la búsqueda semántica entre fragmentos de texto, se probaron distintas métricas de Semejanza:

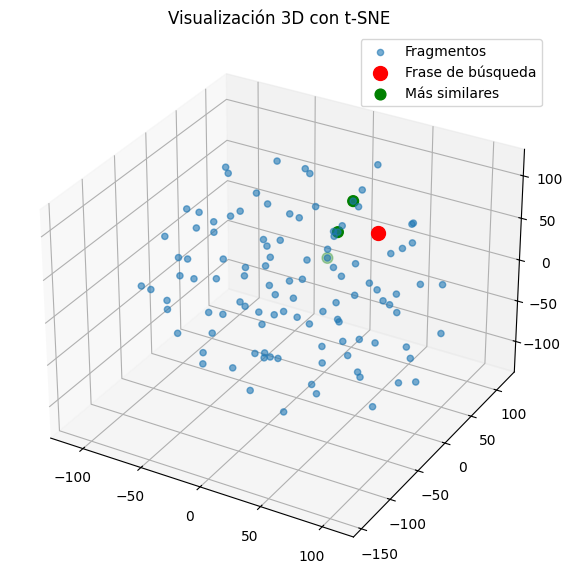
**Similitud del coseno**, **Distancia de Jaccard** y **Similitud de Dice**.

Otras métricas como Levenshtein o Jaro-Winkler se descartaron porque miden diferencias a nivel de caracteres y no son adecuadas para evaluar similitud semántica entre textos largos o fragmentados.

**La Similitud del coseno fue la que mejor reflejó la relación semántica entre la consulta y los fragmentos, por lo que se eligió como métrica principal para el análisis.**

Por último se generó una visualización son T-SNE para ver la ubicación de los fragmentos y la query ingresada vectorizada en el espacio.

Si bien en el caso de esta consulta los fragmentos más similares están cerca graficamente de la frase de búsqueda en otros casos no fue así.



**EJERCICIO 3**

A partir de los fragmentos previamente generados del archivo manual.txt, se aplicaron dos técnicas de procesamiento de lenguaje natural:

Extracción de sustantivos con spaCy:  
 Se utilizó el modelo de spaCy en inglés para procesar cada fragmento y extraer palabras etiquetadas como sustantivos comunes (NOUN) o propios (PROPN).

Y luego reconocimiento de entidades personalizadas con GLINER:  
Se aplicó el modelo multilingüe GLINER. En este caso, se definió una lista personalizada de etiquetas relacionadas con el dominio del juego (por ejemplo, GameComponent, Role, RuleName, board game, entre otras). Para cada fragmento, el modelo identificó menciones relevantes asociadas a estas categorías, permitiendo una extracción semántica más específica y alineada al contexto del juego.

**Búsqueda de similitudes entre palabras**

Se implementaron funciones para cada métrica de distancia y luego una función general que compara una consulta con un conjunto de palabras usando diferentes métodos de similitud. El objetivo fue ver qué palabras son más parecidas a lo que se está buscando, tanto en forma como en significado.

Luego de analizar las métricas se pudo ver que para una “similitud textual” (si se parecen las palabras de forma escrita) las métricas como Jaccard, Levenshtein, Dice, Jaro-Winkler son adecuadas porque miden la similitud basada en caracteres o tokens. Entre estas métricas, Jaro-Winkler suele ser la más efectiva porque pondera los prefijos comunes y corrige bien errores de tipeo al inicio de las palabras.

Pero si pensamos en “similitud semántica” (relación de significado o contexto) similitud por coseno es la mejor opción.

**EJERCICIO 4**

Para este paso se utilizó la librería langdetect, que permite detectar automáticamente el idioma de un texto. Se recorrieron las carpetas que contenían los archivos del trabajo (estadísticas, información y relaciones) y se detectó el idioma de cada archivo .txt o .csv.

Los resultados se guardaron en un DataFrame y, luego, todos los archivos fueron copiados a nuevas carpetas separadas por idioma.

archivo idioma

0 /content/NLP\_TP1/datos/relaciones/creditos\_rel... en

1 /content/NLP\_TP1/datos/informacion/video1.txt nl

2 /content/NLP\_TP1/datos/informacion/video2.txt en

3 /content/NLP\_TP1/datos/informacion/video3.txt en

4 /content/NLP\_TP1/datos/informacion/manual.txt en

5 /content/NLP\_TP1/datos/informacion/review\_bgg.txt es

6 /content/NLP\_TP1/datos/informacion/df\_foros\_bg... en

...

**EJERCICIO 5**

En esta etapa se utilizó un modelo BERT pre entrenado en múltiples idiomas para realizar análisis de sentimientos sobre reseñas de usuarios. Se trabajó con una pipeline de transformers que permite aplicar el modelo directamente sobre los textos, devolviendo una puntuación en estrellas (de 1 a 5).

Luego, estas estrellas se convirtieron a categorías más simples: NEGATIVE (1 o 2 estrellas), NEUTRAL (3) y POSITIVE (4 o 5).

Conversation Sentiment\_raw Sentiment

0 Title: Review: Still fun after ten plays? My p... {'label': '3 stars', 'score': 0.3729034960269928} NEUTRAL

1 Title: Tiny review of Tiny Towns\nOriginal pos... {'label': '4 stars', 'score': 0.4030592143535614} POSITIVE

2 Title: Teensy Towns (a Space-Biff! review)\nOr... {'label': '2 stars', 'score': 0.37357813119888... NEGATIVE

3 Title: Clear Eyes, Tiny Towns, Can't Lose: A R... {'label': '4 stars', 'score': 0.43274393677711... POSITIVE

4 Title: Simple, Thinky and Brutal\nOriginal pos... {'label': '4 stars', 'score': 0.3976544439792633} POSITIVE

5 Title: I recommend Tiny Towns\nOriginal post: ... {'label': '4 stars', 'score': 0.6091551184654236} POSITIVE

6 Title: Tiny Towns Review\nOriginal post: Tiny ... {'label': '4 stars', 'score': 0.4687051475048065} POSITIVE

7 Title: Board Game Gumbo: Spice it up with Tiny... {'label': '3 stars', 'score': 0.34646639227867... NEUTRAL

8 Title: Tiny Towns: A delightful city-building ... {'label': '4 stars', 'score': 0.46982190012931... POSITIVE

9 Title: Tiny Towns Review - Played this 8 times... {'label': '5 stars', 'score': 0.7061994075775146} POSITIVE

Luego se desarrolló una función que permite buscar reseñas similares a una consulta dada (query), utilizando embeddings y similitud del coseno. Además, es posible filtrar las reseñas por sentimiento (positivo, negativo o neutro), lo que permite ajustar la búsqueda según el tipo de opinión que se desea encontrar.

La consulta ingresada se transforma en un vector usando el mismo modelo de embeddings aplicado previamente a las reseñas. Luego, se comparan los vectores usando similitud coseno para detectar qué textos son más parecidos. Se devuelven las reseñas más cercanas, junto con su categoría de sentimiento y el puntaje de similitud.

.

**EJERCICIO 6**

Se creó un archivo .csv con 300 preguntas que están categorizadas entre las fuentes de datos que las responden (estadísticas, información y relaciones).

Pregunta Categoría

0 ¿Cuál es la edad recomendada por la comunidad? Estadísticas

1 ¿Cuántas personas quieren cambiarlo? Estadísticas

2 ¿Cuántas veces se ha jugado este mes? Estadísticas

3 ¿Qué roles cumplen los jugadores en Tiny Towns? Información

4 ¿Qué otros juegos ha hecho Matt Paquette & Co.? Relaciones

5 ¿Quién es el editor de Tiny Towns? Relaciones

6 ¿Cuál es la edad recomendada oficialmente para... Estadísticas

7 ¿Cuál es el número oficial de jugadores? Estadísticas

8 ¿Cuántos comentarios tiene Tiny Towns? Estadísticas

...

Se realizó un análisis de clasificación de preguntas en distintas categorías utilizando dos métodos diferentes de representación de texto.   
Primero, se dividió el conjunto de datos en partes de entrenamiento y prueba manteniendo la proporción original de clases. Luego, se aplicaron dos enfoques para convertir las preguntas en representaciones numéricas: TF-IDF y Doc2Vec. Ambos modelos se evaluaron con métricas estándar de desempeño (precisión, recall, F1-score y exactitud), y se realizó una predicción con una pregunta nueva para validar su funcionamiento.

El modelo TF-IDF combinado con regresión logística obtuvo un rendimiento perfecto (100% en precisión, recall y F1-score) sobre el conjunto de prueba. Esto puede deberse a que las categorías del dataset están claramente diferenciadas por palabras clave específicas, lo que facilita la clasificación con TF-IDF.

Por otro lado, el modelo basado en Doc2Vec mostró un desempeño significativamente inferior, con métricas alrededor del 59%, lo que indica que la representación semántica generada por Doc2Vec no fue tan efectiva para esta tarea específica en comparación con TF-IDF.