Universidad Nacional de Rosario
Facultad de Ciencias Exactas, Ingeniería y Agrimensura
T.U.I.A

Procesamiento del Lenguaje Natural

Trabajo Práctico 1

Clasificador de Recomendaciones Recreativas utilizando Procesamiento de Lenguaje Natural (NLP)

Docentes:
Geary, Alan
Masón, Juan Pablo

Integrantes:
Flaibani, Marcela
Masciangelo, Lucía

2024

Índice

Objetivo

Desarrollar un sistema de recomendación de contenidos que proporcione sugerencias de películas, juegos de mesa y libros en función del estado de ánimo y las preferencias específicas del usuario.

Descripción General del Sistema

El sistema utiliza técnicas de procesamiento de lenguaje natural (NLP) y Machine Learning para analizar el estado de ánimo del usuario, expresado en frases o palabras, y sus preferencias de contenido. En base a ésto, sugiere los títulos de películas, juegos de mesa o libros que podrían ser de su interés.

Estructura

En el repositorio de GitHub (NLP/TP_1 at main · flaibani/NLP) se presentan dos programas y dos carpetas:

Programas

- TP_1_NLP.ipynb
  En este código se procesan o descargan las fuentes de datos.
  Se prueban diferentes técnicas, tanto para la detección del estado de ánimo como para hacer las recomendaciones de películas, juegos y libros, en función de las preferencias del usuario y de su estado de ánimo.
  Se seleccionan los modelos con mejores métricas y se guardan para su posterior uso.
- Ejecutable.ipynb
  El programa interactúa con el usuario. El usuario ingresa dos frases: la primera describe cómo se siente y la segunda cuáles son sus preferencias. En base a esta información se recomiendan películas, juegos y libros (tres opciones por ítem).
  En este código se utilizan los modelos guardados, para evitar el consumo de tiempo que implica el entrenamiento y mejorar la experiencia del usuario.

Carpetas

- Archivos_CSV
  Contiene los siguientes archivos:
  - bgg_database.csv: dataset de juegos.
  - gutenberg_books_detailed.csv: dataset de libros (generado por scrapping).
  - IMDB-Movie-Data.csv: dataset de películas.
- Modelos
  Contiene los modelos entrenados seleccionados:
  - modelo_LR_Em.joblib: modelo regresión logística para la clasificación del estado de ánimo.
  - game_embeddings.pt: modelo de embedding para juegos.
  - book_embeddings.pt: modelo de embedding para libros.
  - movie_embeddings.pt: modelo de embedding para películas.

Programa TP_1_NLP.ipynb - Pasos del desarrollo

1) **Web scraping**:
   Se realiza un web scrapping en la página https://www.gutenberg.org/browse/scores/top1000.php#books-last1, para construir una base de datos de los 1000 libros con mayor cantidad de descargas.
   El proceso consiste en enviar una solicitud a la página principal y almacenar los enlaces de cada libro en el listado de los más descargados. Luego, se accede a cada enlace de forma individual para extraer y almacenar información clave, como el título, autor, resumen, idioma, categorías, fecha de publicación, cantidad de descargas y el enlace directo de descarga.

2) **Detección de estado de ánimo:**
   Se clasifica el estado de ánimo en tres categorías: alegre, neutral o triste. Para ésto, se crea un dataset con 100 frases de cada estado de ánimo, con su correspondiente etiqueta: 0 para alegre, 1 para neutral o 2 para triste, para luego poder entrenar nuestro modelo de Machine Learning.

   - **TF-IDF con Regresión Logística:** Este modelo vectoriza las frases del usuario utilizando TF-IDF y entrena una Regresión Logística para la clasificación. Previamente, se aplica un preprocesamiento que incluye: eliminación de stopwords, puntuación, acentuación, y se transforma el texto a minúsculas.

     Se obtienen las siguientes métricas:

```
Precisión Regresión Logística Train: 0.9796747967479674
Reporte de clasificación Regresión Logística Train:
              precision    recall  f1-score   support

           0       0.96      0.98      0.97        84
           1       0.98      0.99      0.98        81
           2       1.00      0.98      0.99        81

    accuracy                           0.98       246
   macro avg       0.98      0.98      0.98       246
weighted avg       0.98      0.98      0.98       246

Precisión Regresión Logística Test: 0.7903225806451613
Reporte de clasificación Regresión Logística Test:
              precision    recall  f1-score   support

           0       0.93      0.58      0.72        24
           1       0.69      0.90      0.78        20
           2       0.81      0.94      0.87        18

    accuracy                           0.79        62
   macro avg       0.81      0.81      0.79        62
weighted avg       0.82      0.79      0.78        62
```

Comparando las métricas obtenidas en los conjuntos de entrenamiento y testeo, el modelo muestra un sobreajuste en el conjunto de entrenamiento. Por este motivo se prueban otras alternativas.

- **SentenceTransformer y Regresión Logística:** Se emplea un modelo de embeddings semánticos multilingües para vectorizar el texto y un modelo de Regresión Logística para clasificar. En este caso, no es necesario normalizar el texto, ya que el modelo de embeddings es capaz de capturar el contexto sin requerir preprocesamiento adicional.

Se evalúan varios modelos de embeddings, obteniendo las siguientes métricas en entrenamiento y prueba:

- sentence-transformers/all-mpnet-base-v2:

```
Precisión Regresión Logística Train: 0.8373983739837398
Reporte de clasificación Regresión Logística Train:
              precision    recall  f1-score   support

           0       0.82      0.81      0.81        84
           1       0.88      0.81      0.85        81
           2       0.82      0.89      0.85        81

    accuracy                           0.84       246
   macro avg       0.84      0.84      0.84       246
weighted avg       0.84      0.84      0.84       246

Precisión Regresión Logística Test: 0.7741935483870968
Reporte de clasificación Regresión Logística Test:
              precision    recall  f1-score   support

           0       0.86      0.75      0.80        24
           1       0.80      0.80      0.80        20
           2       0.67      0.78      0.72        18

    accuracy                           0.77        62
   macro avg       0.77      0.78      0.77        62
weighted avg       0.78      0.77      0.78        62
```

- msmarco-MiniLM-L-6-v3:

```
Precisión Regresión Logística Train: 0.9512195121951219
Reporte de clasificación Regresión Logística Train:
              precision    recall  f1-score   support

           0       0.96      0.90      0.93        84
           1       0.95      1.00      0.98        81
           2       0.94      0.95      0.94        81

    accuracy                           0.95       246
   macro avg       0.95      0.95      0.95       246
weighted avg       0.95      0.95      0.95       246

Precisión Regresión Logística Test: 0.6612903225806451
Reporte de clasificación Regresión Logística Test:
              precision    recall  f1-score   support

           0       0.75      0.50      0.60        24
           1       0.67      0.70      0.68        20
           2       0.60      0.83      0.70        18

    accuracy                           0.66        62
   macro avg       0.67      0.68      0.66        62
weighted avg       0.68      0.66      0.66        62
```

- intfloat/multilingual-e5-small:

```
Precisión Regresión Logística Train: 0.8699186991869918
Reporte de clasificación Regresión Logística Train:
              precision    recall  f1-score   support

           0       0.89      0.83      0.86        84
           1       0.87      0.85      0.86        81
           2       0.85      0.93      0.89        81

    accuracy                           0.87       246
   macro avg       0.87      0.87      0.87       246
weighted avg       0.87      0.87      0.87       246

Precisión Regresión Logística Test: 0.8870967741935484
Reporte de clasificación Regresión Logística Test:
              precision    recall  f1-score   support

           0       0.95      0.83      0.89        24
           1       0.82      0.90      0.86        20
           2       0.89      0.94      0.92        18

    accuracy                           0.89        62
   macro avg       0.89      0.89      0.89        62
weighted avg       0.89      0.89      0.89        62
```

Se elige esta opción con el modelo **intfloat/multilingual-e5-small**, que muestra las mejores métricas en el conjunto de prueba, presentando además un menor nivel de sobreajuste.

- **Modelo tweeter:** También se prueba un modelo pre-entrenado con 1.6 millones de tweets etiquetados, utilizando un modelo de Regresión Logística. Sin embargo, el modelo sólo incluye etiquetas para clasificaciones binarias (positivo/negativo), lo cual resulta insuficiente para este trabajo, por lo que fue descartado. Debido al tamaño del archivo "training.1600000.processed.noemoticon.csv", no fue posible subirlo a GitHub, por lo que esta prueba no podrá realizarse.

- **K-Nearest Neighbors (KNN)** Se realiza el mismo procedimiento que en el caso anterior utilizando TF-IDF para vectorizar el texto, y como clasificador utilizamos KNN.

```
Precisión Regresión Logística Train: 0.9796747967479674
Reporte de clasificación Regresión Logística Train:
              precision    recall  f1-score   support

           0       0.96      0.98      0.97        84
           1       0.98      0.99      0.98        81
           2       1.00      0.98      0.99        81

    accuracy                           0.98       246
   macro avg       0.98      0.98      0.98       246
weighted avg       0.98      0.98      0.98       246

Precisión K-Nearest Neighbors: 0.7258064516129032
Reporte de clasificación K-Nearest Neighbors:
              precision    recall  f1-score   support

           0       0.68      0.62      0.65        24
           1       0.68      0.75      0.71        20
           2       0.83      0.83      0.83        18

    accuracy                           0.73        62
   macro avg       0.73      0.74      0.73        62
weighted avg       0.73      0.73      0.72        62
```

3) **Carga y Preprocesamiento de Datasets:**
   El programa trabaja con tres datasets principales: películas, juegos de mesa y libros. Los primeros dos datasets fueron proporcionados, mientras que el de libros fue generado mediante web scraping en Project Gutenberg. Para cada tipo de contenido, seleccionamos las columnas más relevantes:
   - Películas: género, descripción, director y actores.
   - Juegos de Mesa: nombre del juego, descripción y categorías.
   - Libros: título, autor, resumen y categorías.

Cada dataset se procesa de forma independiente, con el fin de tener la información necesaria para responder a las búsquedas del usuario.

4) **Preferencias del usuario:**
   Para generar recomendaciones personalizadas, el sistema convierte las descripciones de cada contenido en vectores de embeddings, utilizando dos enfoques diferentes:
   - Utilizando **TF-IDF:**
     Este enfoque de vectorización convierte las descripciones en vectores de frecuencia de palabras, ajustados por su importancia relativa en el dataset. Se aplicaron varias funciones de preprocesamiento en este método, como eliminación de stopwords, puntuación y acentos, y se transformó todo el texto a minúsculas.
   - Utilizando **SentenceTransformer:**
     Utilizando el modelo 'intfloat/multilingual-e5-small', se transforman las descripciones de contenido en vectores de embeddings, que representan la semántica de cada texto en un espacio multidimensional. No requiere preprocesamiento, ya que los embeddings capturan el contexto.

   Se elige el modelo de Sentence Transformer, luego de testear ambos modelos y observar que éste producía puntuaciones de similitud más altas (por encima del 80%) en comparación con el modelo de TF-IDF.

Recomendaciones y traducciones

El usuario utiliza el idioma español y los datasets están en inglés. Se realiza la traducción de los textos ingresados por el usuario al idioma inglés.

Una vez traducidos, el programa utiliza los modelos de embeddings mencionados anteriormente para comparar con las descripciones, y así poder recomendarle al usuario algo en base a sus preferencias y estado de ánimo.

Finalmente, una vez realizada la recomendación por el programa, se traducen los resultados al español, para mostrarle al usuario las recomendaciones en su idioma original.
Para cada búsqueda, el sistema muestra tres recomendaciones de cada tipo de contenido (películas, juegos de mesa y libros), incluyendo sus títulos y descripciones para facilitar la elección final del usuario.

Conclusión

Este sistema proporciona una herramienta útil para ofrecer recomendaciones de entretenimiento en función de la percepción emocional y preferencias de contenido del usuario. Utiliza enfoques de NLP modernos y modelos de Machine Learning, permitiendo una experiencia de usuario personalizada y precisa.