Sistemas de recomendación

Modelos basados en contenido

Universidad de La Laguna
Grado en Ingeniería Informática
Gestión del Conocimiento en las Organizaciones
Óscar Navarro Mesa (alu0101504094@ull.edu.es)
Alejandro Javier Aguiar Pérez (alu0101487168@ull.edu.es)
Leandro Samuel Armas Torres (alu0101464305@ull.edu.es)

ÍNDICE

Introducción **2**
Objetivo de la Práctica **2**
Herramientas y Dependencias **2**
Estructura del Código **3**
Ejecución y Ejemplo de Uso **4**
Argumentos 4
Ejecución 5
Ejemplo de Uso 5
Resultados Obtenidos **7**
Análisis de los Valores TF-IDF 7
Matriz de Similitud Coseno 7
Observaciones Generales 7
Conclusiones **8**

Introducción

En esta práctica se ha desarrollado un sistema de recomendación basado en el contenido utilizando Python. Estos sistemas tienen como objetivo personalizar recomendaciones en función de las características de los propios documentos, sin requerir la interacción directa de los usuarios, lo que los hace especialmente útiles en entornos donde se cuenta con un conjunto de documentos y se busca recomendar aquellos que mejor se relacionan entre sí o con un documento en particular.

El sistema implementado procesa documentos de texto, elimina términos irrelevantes (palabras de parada o "stop words") y genera una matriz de términos representada mediante valores TF-IDF. Además, se calcula la similitud coseno entre pares de documentos para medir la relación entre ellos en términos de contenido.

Objetivo de la práctica

El objetivo de esta práctica fue desarrollar un software que hiciera lo siguiente:

- Procesar un conjunto de documentos en formato de texto plano y eliminar palabras poco informativas utilizando un archivo de palabras de parada.
- Lematizar los términos mediante un archivo de lematización, para mejorar la precisión de los términos.
- Calcular el valor TF-IDF (Term Frequency-Inverse Document Frequency) para cada término en cada documento.
- Determinar la similitud coseno entre cada par de documentos y devolver una matriz de similitudes.

Herramientas y dependencias

Para el desarrollo de esta práctica se usó Python aprovechando varias de sus librerías estándar y otras herramientas de visualización de datos. Las que utilizamos fueron:

- **argparse**: Permite crear una interfaz de línea de comandos para gestionar las entradas del programa.
- **json**: Facilita la lectura y manejo de archivos JSON, utilizado en esta práctica para cargar el archivo de lematización.
- **math**: Proporciona funciones matemáticas como logaritmos y raíces cuadradas necesarias para los cálculos de similitud coseno y TF-IDF.
- **re**: Librería de expresiones regulares para la manipulación de texto, utilizada en el preprocesamiento de documentos para limpiar y normalizar los términos.

- **collections.Counter y collections.defaultdict**: Utilidades que permiten crear conteos automáticos de términos y diccionarios con valores por defecto, facilitando el conteo de palabras y la organización de datos.
- **matplotlib.pyplot**: Librería de visualización utilizada para crear gráficos y representar visualmente las relaciones entre documentos, como matrices de similitud.
- **seaborn**: Librería de visualización avanzada basada en matplotlib, que permite mejorar la apariencia y claridad de los gráficos generados para la matriz de similitud.

Ejecución y Ejemplo de Uso

El programa desarrollado se ejecuta desde la línea de comandos, utilizando una serie de argumentos que permiten especificar las rutas de los archivos de entrada y salida. A continuación, se describe cómo utilizar estos argumentos y un ejemplo práctico de ejecución.

Argumentos

El programa acepta los siguientes argumentos:

- -d o --documents: Ruta al archivo de documentos en formato .txt, donde cada línea representa un documento individual.
- -s o --stopwords: Ruta al archivo de palabras de parada (stop words) en formato .txt, utilizadas para eliminar términos poco informativos.
- -l o --lemmatization: Ruta al archivo de lematización en formato .json, que permite normalizar los términos a su forma básica.
- -o o --output: Ruta del archivo de salida en el que se guardarán los resultados, incluyendo los valores TF, IDF, TF-IDF de cada término y la matriz de similitud coseno entre los documentos.
- -g o --graph: Ruta del archivo de imagen donde se guardará el gráfico de la matriz de similitud coseno, que se genera como un mapa de calor para facilitar la interpretación visual.

Ejecución

Para ejecutar el programa, el siguiente comando muestra la estructura general en la que deben incluirse las rutas específicas de cada archivo:

```bash
python3 src/Sistemas-Recomendacion-Basado-Contenido.py -d
<data/examples_documents/path_to_documents.txt> -s <data/stop-words/path_to_stopwords.txt> -l
<data/corpus/path_to_corpus.json> -o <outputs/path_to_output.txt> -g
<outputs/path_to_graph.png>
```

Ejemplo de Uso

A continuación, se muestra un ejemplo concreto de ejecución con archivos específicos:

```bash
python3 src/Sistemas-Recomendacion-Basado-Contenido.py -d data/examples-documents/documents-01.txt -s data/stop-words/stop-words-en.txt -l data/corpus/corpus-en.json -o outputs/resultado.txt -g outputs/similarity_matrix.png
```

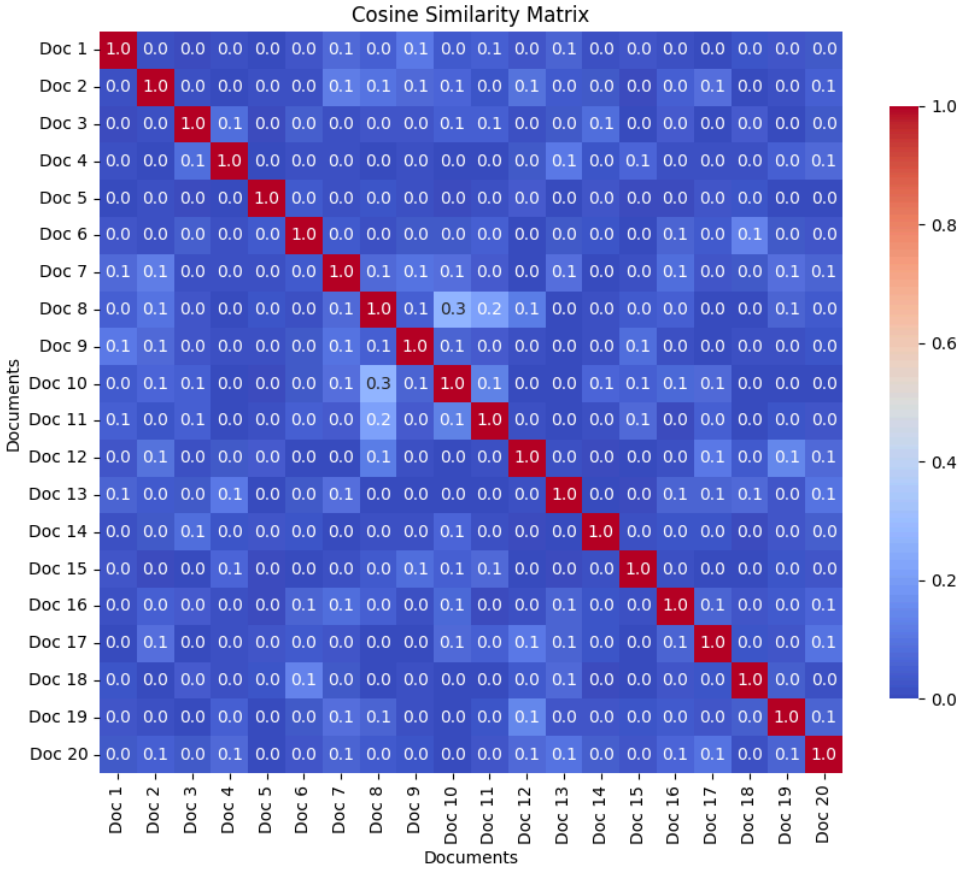
En este ejemplo, el programa procesa el archivo documents-01.txt, eliminando las palabras de parada indicadas en stop-words-en.txt y lematizando los términos de acuerdo con el diccionario en corpus-en.json. Los resultados con las tablas de TF, IDF, TF-IDF y la matriz de similitud coseno, se guardarán en resultado.txt.

Ejemplo de resultado.txt:

```output.txt
Documento 1:
Index     Term          TF        IDF       TF-IDF
0         aromas        1.0000    0.6931    0.6931
1         include       1.0000    2.3026    2.3026
...
Documento 2:
Index     Term          TF        IDF       TF-IDF
0         ripe          1.0000    1.8971    1.8971
1         fruity        1.0000    2.3026    2.3026
...
Similitudes coseno entre documentos:
Doc 1: 1.0000, 0.0083, 0.0077, ...
Doc 2: 0.0083, 1.0000, 0.0185, ...
```

Además, se generará un gráfico de la matriz de similitud en similarity_matrix.png para su análisis visual.

Ejemplo de similarity_matrix.png:



Resultados Obtenidos

A continuación, se presentan los resultados obtenidos para los Documentos 1 y 2, que incluyen la tabla de valores TF-IDF y la matriz de similitudes coseno.

Análisis de los Valores TF-IDF

Los términos con valores TF-IDF altos en cada documento indican palabras relevantes que ayudan a identificar el tema principal del contenido. Para los documentos procesados, observamos los siguientes puntos clave:

- **Documento 1**: Los términos con mayor valor TF-IDF incluyen:
  - *tropical*, *broom*, *brimstone*, *unripened*, *dried*, y *sage* (con un TF-IDF de 2.9957 cada uno).
  - Estos términos sugieren un perfil descriptivo de aromas y sabores en un contexto gastronómico, utilizando términos especializados que describen sabores y características aromáticas.
- **Documento 2**: En este documento, los términos más destacados son:

- *smooth*, *structured*, *fill*, *freshened*, *drinkable*, y *2016* (con un TF-IDF de 2.9957 cada uno).
- La relevancia de estos términos apunta a una descripción más estructural y de añada de una bebida, probablemente vino, con énfasis en características de suavidad y estructura.

El análisis TF-IDF en estos documentos muestra cómo cada uno resalta ciertos términos únicos, alineados con descripciones propias de catas o reseñas de productos.

Matriz de Similitud Coseno

La matriz de similitud coseno entre documentos muestra que Documentos 1 y 2 tienen un bajo grado de similitud (valor de 0.0083). Esta baja puntuación indica que los documentos comparten pocos términos significativos y, en su mayoría, poseen vocabulario distintivo, lo cual es esperable dada la especificidad de cada uno.

- **Similitudes Notables**:
  - Las similitudes más altas en la matriz incluyen el propio Documento 1 (valor de 1.0000, pues la similitud coseno de un documento consigo mismo es máxima).
  - En los documentos 1 y 2, las mayores similitudes aparecen con términos relacionados con la *acidez* y palabras comunes como *a*, aunque estas palabras tienen un peso menor debido a su menor especificidad en el contexto general.

Observaciones Generales

Estos resultados permiten inferir la diversidad temática y específica de cada documento, donde los documentos 1 y 2 presentan temas complementarios pero no idénticos. La baja similitud entre ambos respalda que cada documento posee un conjunto único de términos representativos.

En conclusión, el sistema de recomendación basado en contenido logra resaltar las diferencias en el uso de términos y calcula la similitud coseno entre documentos, facilitando la identificación de temas y características distintivas entre ellos.

Conclusiones

Esta práctica nos demostró cómo los sistemas de recomendación basados en contenido son usados para analizar y organizar colecciones de documentos. Además, ofrece una base sólida para futuras implementaciones y adaptaciones en sistemas de recomendación más complejos, incluyendo la posibilidad de combinar este enfoque con otros métodos, como los sistemas basados en el comportamiento del usuario o en la colaboración entre usuarios.