Universidad de la Habana, MATCOM
Sistemas de Recuperación de Información (2022)

Informe de Entrega Parcial

Samuel David Suárez Rodríguez C512
Enmanuel Verdesia Suárez C511
Gabriel Fernando Martín Fernández C511

1. Descripción General

1.1. Resumen

Se pretende desarrollar un Sistema de Recuperación de Información (SRI) con el apoyo de técnicas de Inteligencia Artificial (IA). Dado un conjunto de varios documentos de diversos temas se desarrollará una aplicación de búsqueda que obtenga resultados relevantes y acordes a una consulta presentada por el usuario. Además se emplearán métodos para clasificar los documentos y extraer sus características más relevantes usando algoritmos para la clasificación y selección de características. También se desarrollarán herramientas para generar clústeres de documentos y agrupar los que tengan contenidos afines en un mismo grupo. Todos estos procesos de clasificación y agrupamiento se apoyarán en el uso de algoritmos de machine learning (ML) para lograr su objetivo.

1.2. Estructura del proyecto

El proyecto presenta tres componentes principales: el backend o componente lógica de la aplicación, junto con la API, y el frontend en forma de un servicio web.

Está desarrollado en GitHub bajo la organización Gologle: https://github.com/Gologle.

Ahí pueden ser encontrados los dos repositorios principales. Además están hospedados el cliente del sistema y la API en internet.

- Web: https://gologle.vercel.app
- API: https://gologle-api.herokuapp.com/

2. Detalles de implementación

2.1. Procesamiento de los documentos

Cada set de datos empleado tiene un parser asociado para extraer la información de estos y poder delimitar e identificar cada documento que contiene de forma única por un ID. Esto permite poder guardar el texto de los documentos en una base de datos SQLite y obtener un menor tiempo de respuesta ante un pedido de parte del cliente.

Dado que ha sido implementado el modelo vectorial, haciendo uso de la biblioteca `sklearn` nos ajustamos a la forma que esta requiere para representar los documentos. Dicha biblioteca se usó para tokenizar los documentos y extraer como *features* los términos que estos contienen. El texto de los documentos es llevado a minúscula, además los términos de un simple carácter son eliminados. Así se obtiene cada documento como un vector de términos (palabras) con una dimensión determinada.

A partir del conjunto de documentos se obtiene una matriz esparcida que representa la cantidad de veces que aparece el término i en el documento j. Sobre esta matriz nos apoyamos para calcular la frecuencia de documentos inversa (idf) para cada término. Luego haciendo uso de la clase

`sklearn.feature_extraction.text.TfidfTransformer` calculamos los pesos para cada término en cada documento. Hecho esto, se tiene la matriz esparcida de los pesos, la cual es guardada en la base de datos para agilizar el tiempo de respuesta.

Resumiendo el proceso antes descrito, la base de datos es considerada el índice de nuestro sistema. Almacena los documentos, los términos, *idf* para cada término y pesos de los términos en cada documento. Su objetivo es agilizar el tiempo de respuesta.

2.2. Recuperación de Información

Cuando es recibida una consulta esta es procesada de forma similar a como se procesa un documento. Se calcula el peso para cada término y la similitud entre el vector obtenido y los documentos de la base de datos usando como valor el coseno del ángulo entre estos. Los documentos con mayor similitud son dados como resultado.

3. Próximos Pasos

- Implementar métricas para valorar la efectividad de un modelo.
- Implementar modelos más efectivos y comparar resultados.
- Añadir más sets de datos.
- Mejorar el tiempo de respuesta.