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. Todos estos procesos de clasificación 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. 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/

Detalles de implementación

2. Procesamiento de los sets de datos

Para cada set de datos empleado fue implementado un parser para extraer su información. Este proceso no se pudo realizar de forma común pues cada set contenía los documentos de forma particular. Una vez *parseados* los sets de datos se puede delimitar e identificar cada documento que contiene de forma única por un ID y así guardar el texto de los documentos en una base de datos SQLite. Dicha base de datos facilita obtener un menor tiempo de respuesta ante un pedido de parte del cliente.

Los sets de datos con los que se trabajó fueron Cranfield (1400 documentos), Newsgroups (18828 documentos) y Reuters (????? documentos). Se contaba con consultas y resultados relevantes para Cranfield. Que se tiene en Newsgroups y/o Reuters?

3. Modelo Vectorial (TF-IDF)

Este modelo esta basado en la representación de los documentos mediante *bag-of-words*. Es decir, se transforma cada documento a una vector de tamaño fijo que solo contiene información de la cantidad de veces que aparece cada palabra por componente. Este modelo a pesar de ser efectivo tiene como debilidad que se pierde la información respecto al orden de las palabras, además, como no aprende el significado de las palabras, la distancia entre vectores no siempre representa una diferencia en el significado.

3.1. Preprocesamiento

Dado que ha sido implementado el modelo vectorial basado en TF-IDF (del inglés, *term frequency* e *inverse document frequency*), 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.

3.2. Construcción del índice

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.

$$idf_i = \log \frac{N}{n_i}$$

n_i: cantidad de documentos en los que aparece el término i.

Luego haciendo uso de la clase `TfidfTransformer` que se encuentra en el módulo `sklearn.feature_extraction.text` calculamos los pesos para cada término en cada documento, de acuerdo a la expresión:

$$w_{ij} = tf_{ij} * idf_i$$

donde t_{ij} es la frecuencia normalizada del término i en el document j,

$$tf_{ij} = \frac{freq_{ij}}{max_l freq_{lj}}$$

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.

La base de datos creada a partir de los documentos, los términos, la idf para cada término y los pesos es considerada el índice del sistema para satisfacer las consultas de la forma más rápida posible.

3.3. Recuperación de Información

Cuando es recibida una consulta q, esta es procesada de forma similar a como se procesa un documento. Se calcula el peso para cada término (w_{iq}), mediante la siguiente expresión

$$w_{iq} = (\alpha + (1 - \alpha)\frac{freq_{lq}}{max_l freq_{lq}}) * idf_i$$

donde α es un valor de suavizado que permite amortiguar la contribución de la frecuencia de los términos, se le asignó un valor de 0.5.

La similitud entre el vector obtenido y los documentos de la base de datos es calculada usando como valor de referencia el coseno del ángulo entre estos. Los documentos con mayor similitud son dados como resultado de forma ordenada.

4. Modelo Vectorial (Doc2Vec)

4.1. Preprocesamiento

4.2. Construcción del índice

4.3. Recuperación de Información

5. Evaluación de los modelos

Métricas objetivas y subjetivas estudiadas empleando 2 colecciones

Mostrar consultas con los resultados (al menos 1 por colección)

6. Ventajas y Desventajas del Sistema

Aquí creo se puede hablar de todo el sistema en general con un análisis crítico

Recomendaciones

Como mejorar la propuesta?