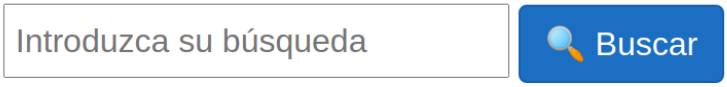
Informe Prácticas de Programación Primer Año

David Michel García Batista

Julio de 2023



Abstract

¿En que consiste este proyecto? ¿Qué es Moogle! Moogle! es una aplicación web (cuyo nombre desborda originalidad) que constituye un modelo de búsqueda vectorial con la finalidad de encontrar en un conjunto de documentos aquellos más relevantes y relacionados con la información deseada. Ésta información será un conjunto de palabras introducidas por el usuario en lo que denominaremos Query.

Para poder llevar a cabo dicha tarea es necesario de alguna manera precisar o mejor dicho, cuantificar que tan relevante es cada documento con respecto a esta Query ingresada para luego poder compararlos y determinar aquellos mas importantes. Pues para esto utilizaremos dos procesos fundamentales, el TF-IDF (Term Frequency-Inverse Document Frequency) y la similitud de cosenos. Ahora procedemos a describir como funciona esto.

Lo primero es comprender en que consiste el TF-IDF y la similitud de cosenos:

- **TF**: No es más que la frequencia con la que aparece una palabra en un texto dependiendo de la cantidad de veces que aparece (Si una palabra aparece 5 veces en un texto de 100 palabras su TF=0,05)
  Fórmula:
  $$TF = \frac{t}{td}$$
  Donde:
  - **t** representa la cantidad de apariciones del término en el documento.
  - **td** la cantidad de términos que tiene el documento.
- **IDF**: Su función es básicamente determinar la rareza de una palabra, o sea, analiza la aparición de una palabra en todos los textos por lo que mientras más común sea un término mas baja será su puntuación, de esta manera se logra despreciar aquellas palabras que se repiten mucho como las preposiciones, dándole prioridad a aquellas palabras "raras" que aparecen pocas veces y se sobrentiende que sean más importantes.
  Fórmula:
  $$IDF = \log(\frac{N}{DF})$$
  Donde:
  - **N** representa el número total de documentos en la colección.
  - **DF** es el número de documentos en los que aparece un término específico.

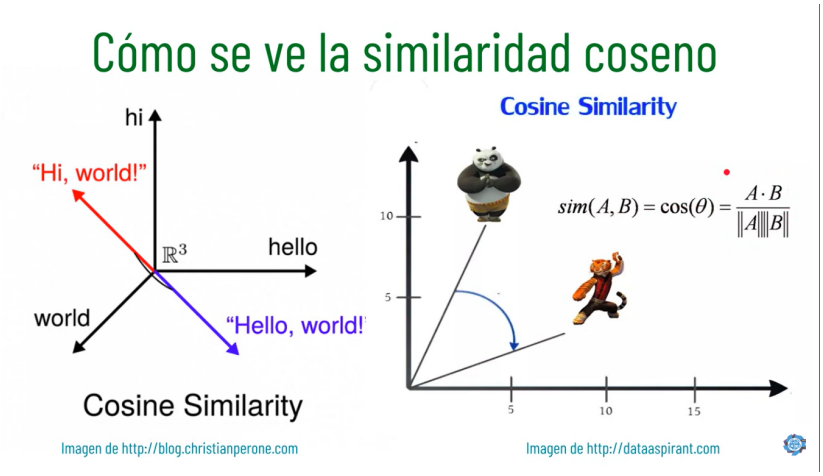
- **Similitud de coseno**: Es un concepto medianamente abstracto, pues permite analizar vectores en el espacio mediante una formúla para determinar el ángulo entre ellos, mientras más pequeño sea el ángulo más cercano estará el valor de su coseno a 1 y por ende mayor similitud tendrá. Ésto es lo que usaremos para comparar cada documento con la Query y así determinar la relevancia. Fórmula:

$$sim(A, B) = cos(\theta) = \frac{A \bigodot B}{\|A\| * \|B\|}$$

  Donde:

  - **A** y **B** son dos vectores.
  - $(A \bigodot B)$ representa el producto escalar entre los vectores A y B, es el producto de los valores correspondientes de los componentes sumados entre sí.
  - $\|A\|$ y $\|B\|$ representan las normas de los vectores A y B. La norma de un vector es la magnitud o longitud del vector y se calcula como la raíz cuadrada de la suma de los cuadrados de sus componentes.

    En la siguiente figura se ve una representación gráfica de esta similitud de coseno para una mejor comprensión



El principio del funcionamiento del proyecto se encuentra dentro de **Moogle Engine** el cual contiene las distintas clases que realizarán las funciones y procesos del programa. La idea es procesar la información de cada documento que se encuentre dentro de la carpeta Content y crear un diccionario para cada documento que contenga una relacion de cada palabra del texto y su correspondiente **TF-IDF**, luego construir una Base de Datos que contenga a todos los documentos y sus relevancias con respecto a la **Query**. Ahora, para determinar el valor de dicha relevancia será necesario procesar la Query de la misma manera que un documento para calcular su TF-IDF, una vez logrado ésto podremos aplicar la fórmula de la similitud de cosenos para calcular estos valores y luego compararlos para dar como resultado de la búsqueda aquellos con mayor relevancia. Como no sería eficiente imprimir todo el documento como resultado, lo mejor es asignarle el título y un fragmento del texto llamado **Snippet** que estará relacionado con la Query para ser imprimidos en pantalla.

Todo este funcionamiento estará respaldado por varias clases que ejecutarán sus funciones.

- **Clase Document** Procesa la información de los documentos individualmente utilizando:
  - Directory.GetCurrentDirectory: Obtener las localizaciones de los archivos .txt
  - Directory.GetFiles: Para conseguir el contenido de dichos archivos.
  - Método Split Text: Para separar los textos por palabas y llevarlas a minúscula usando.ToLower.
  - Método Get Title: Para conseguir el título de los txt. Támbién esta clase servirá para construir el diccionario con la relación palabra y TF-IDF de cada documento.
- **Clase TF-IDF** Estará encaminada específicamente al cálculo de este parámetro.
- **Clase Database** Siguiendo la línea de originalidad en los nombres será la encargada de construir la base de datos con la información de todos los documentos y la Query para poder aplicar la similitud de cosenos y determinar las relevancias, finalmente teniendo todo lo necesario para dar los resultados de la búsqueda.