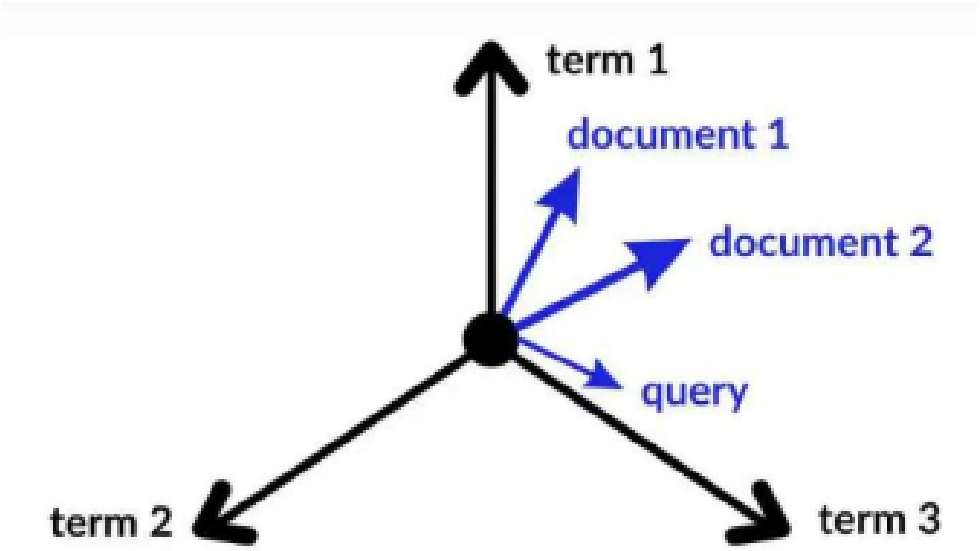
INSTITUTO TECNOLÓGICO BELTRÁN
Centro de Tecnología e Innovación

Informe

Trabajo Práctico N.º 8
Modelo De Espacio Vectorial



Nombre: Coral Tolazzi
Tema: Recuperación de la Información
Profesora: Yanina Ximena Scudero
Cuatrimestre y Año: 1 Cuatrimestre del 2025

Instituto tecnológico Beltrán
Procesamiento del Lenguaje Natural

EJERCICIO:

Crear un programa en Python que calcule y visualice la similitud entre documentos utilizando el modelo de espacio vectorial con la técnica de TF-IDF y similitud del coseno.

Se tienen 3 documentos con información sobre animales:

- **doc1**: "El veloz zorro marrón salta sobre el perro perezoso."
- **doc2**: "Un perro marrón persiguió al zorro."
- **doc3**: "El perro es perezoso."

Pasos a seguir:

1. **Convertir los documentos a vectores numéricos** utilizando la técnica de **TF-IDF (Term Frequency - Inverse Document Frequency)**.

2. **Calcular la similitud del coseno** entre los documentos para medir qué tan parecidos son entre sí.

3. **Visualizar la matriz de similitud** utilizando un mapa de calor (heatmap) que permita interpretar fácilmente los resultados.

Explicacion del Codigo:

1. Importación de librerías
```
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
```

- matplotlib.pyplot: Sirve para crear gráficos.
- seaborn: Librería basada en matplotlib para gráficos estadísticos más atractivos.
- TfidfVectorizer: Convierte texto en vectores numéricos usando el peso TF-IDF.
- cosine_similarity: Calcula la similitud del coseno entre vectores (documentos).

2. Definición de los documentos
```
documents = [
    "El veloz zorro marrón salta sobre el perro perezoso.",
    "Un perro marrón persiguió al zorro.",
    "El perro es perezoso."
]
```

Define los tres documentos de entrada que serán comparados.

3. Conversión a vectores TF-IDF

```
vectorizer = TfidfVectorizer()
tfidf_matrix = vectorizer.fit_transform(documents)
```

- TfidfVectorizer() crea el vectorizador.
- fit_transform(documents) transforma cada documento en un vector numérico donde cada posición representa una palabra, y el valor indica su importancia en el documento (frecuencia relativa ajustada por frecuencia global).

4. Cálculo de la similitud del coseno

```
cosine_sim = cosine_similarity(tfidf_matrix, tfidf_matrix)
```

- Calcula la similitud entre todos los pares de documentos.
- El resultado es una matriz 3x3 donde:
    - Cada celda [i][j] muestra cuán similar es el documento i al documento j.
    - El valor varía entre 0 (nada similar) y 1 (idéntico).

5. Visualización con mapa de calor

```
plt.figure(figsize=(8, 6))
sns.heatmap(cosine_sim, annot=True, cmap="Blues",
        xticklabels=[f"Doc{i+1}" for i in range(len(documents))],
        yticklabels=[f"Doc{i+1}" for i in range(len(documents))])
plt.title("Matriz de Similitud del Coseno")
plt.show()
```

- plt.figure(figsize=(8, 6)): Define el tamaño del gráfico.
- sns.heatmap(...): Crea un mapa de calor con la matriz de similitud:
    - annot=True: Muestra los valores numéricos dentro de las celdas.
    - cmap="Blues": Usa una paleta de colores azul.
    - xticklabels y yticklabels: Etiqueta las filas y columnas como Doc1, Doc2, etc.

- plt.title(...): Agrega un título al gráfico.
- plt.show(): Muestra el gráfico en pantalla.