Espacio Vectorial:
Práctica.
Informe técnico: Análisis de Similitud entre Documentos utilizando TF-IDF y Similitud del Coseno.

1. Introducción.

En el presente informe se describe un programa desarrollado en Python que permite calcular el grado de similitud entre documentos textuales mediante técnicas del modelo de espacio vectorial. Se emplea la biblioteca ***scikit-learn*** para transformar el texto en vectores numéricos a través del esquema **TF-IDF** (*Term Frequency - Inverse Document Frequency*), y para calcular la **similitud del coseno** entre dichos vectores. Finalmente, los resultados son visualizados mediante una matriz gráfica usando ***matplotlib***.

2. Objetivo del Programa.

El objetivo principal del programa es:

- Convertir documentos de texto en representaciones numéricas vectoriales.
- Medir cuán similares son entre sí utilizando métricas matemáticas.
- Representar visualmente los resultados para facilitar su interpretación.

Este tipo de análisis es común en tareas de minería de texto, recuperación de información, clasificación de documentos y sistemas de recomendación.

3. Descripción del Funcionamiento.

A. Carga de documentos.

Se definen tres documentos representados por cadenas de texto:

```
docs = {
   "doc1": "El veloz zorro marrón salta sobre el perro perezoso.",
   "doc2": "Un perro marrón persiguió al zorro.",
   "doc3": "El perro es perezoso."
}
```

B. Vectorización TF-IDF

Espacio Vectorial:
Práctica.

Se utiliza TfidfVectorizer de sklearn.feature_extraction.text para transformar los documentos en vectores numéricos que reflejan la importancia relativa de cada palabra:

vectorizador = TfidfVectorizer()

tfidf_matrix = vectorizador.fit_transform(textos)

Este paso convierte los textos en una matriz dispersa donde cada fila representa un documento y cada columna representa un término.

C. Cálculo de similitud del coseno

Se aplica la función cosine_similarity de sklearn.metrics.pairwise para obtener una matriz cuadrada que indica el grado de similitud (entre 0 y 1) entre cada par de documentos:

sim_matrix = cosine_similarity(tfidf_matrix)

Un valor cercano a 1 indica alta similitud; valores cercanos a 0 indican baja similitud.

D. Visualización con matplotlib

La matriz de similitud se representa visualmente mediante una **matriz de calor** (matshow), donde los colores y los valores numéricos permiten identificar rápidamente qué documentos son más similares entre sí:

fig, ax = plt.subplots()

cax = ax.matshow(sim_matrix, cmap='Blues')

plt.show()

4. Resultados esperados

El programa genera una matriz de 3x3 en la que cada celda [i][j] representa la similitud entre el documento i y j. Por ejemplo:

	doc1	doc2	doc3
doc1	1.00	0.44	0.51
doc2	0.44	1.00	0.24
doc3	0.51	0.24	1.00

Esto indica, por ejemplo, que doc1 y doc3 son moderadamente similares (0.51), mientras que doc2 y doc3 son menos similares (0.24).

5. Requisitos del sistema

- Python 3.x
- Bibliotecas necesarias:
    - scikit-learn (pip install scikit-learn)
    - matplotlib (pip install matplotlib)

6. Conclusión

El programa implementado permite realizar un análisis de similitud entre documentos utilizando técnicas ampliamente validadas en el campo de la lingüística computacional. Gracias a la combinación de TF-IDF y similitud del coseno, es posible transformar contenido textual en información cuantificable y visual, facilitando la comparación automatizada entre textos.

Este tipo de solución es escalable y puede adaptarse para trabajar con grandes volúmenes de documentos, como artículos, correos electrónicos, publicaciones en redes sociales, entre otros.