ESCUELA POLITÉCNICA NACIONAL

RECUPERACION DE INFORMACIÓN

Proyecto Bimestral: Sistema de Recuperación de Información basado en Reuters-21578

Elaborado por:
Dilan Andrade
David Calahorrano
Andrés Zambrano

Profesor:
Ing. Carrera Izurieta Ivan Marcelo

Sistema de Recuperación de Información (SRI)

1. **Introducción**

   El Sistema de Recuperación de Información (SRI) desarrollado tiene como objetivo permitir la búsqueda eficiente de documentos relevantes en el corpus Reuters-21578. Este proyecto abarca varias etapas, incluyendo la adquisición, preprocesamiento, representación vectorial, indexación, búsqueda y evaluación del sistema.

   A lo largo de cada fase se tomaron decisiones para maximizar la calidad de los resultados y optimizar los tiempos de procesamiento. A continuación, se detallan los procesos, decisiones tomadas y resultados obtenidos en cada etapa del desarrollo del sistema.

2. **Fases del Proyecto**

   2.1. Adquisición de Datos

   Objetivo: Unificar los textos del corpus Reuters-21578 en un formato estructurado y fácil de manipular.

   Proceso:

   1. **Identificación del corpus:** El corpus estaba organizado en archivos de texto distribuidos en las carpetas data/test y data/training.
   2. **Extracción de datos:** Se utilizó Python para recorrer cada archivo, leer su contenido y almacenar información clave:
      - **ID del documento:** Basado en el nombre del archivo.
      - **Contenido textual del documento.**
      - **Subcarpeta de origen:** Indicando si el documento pertenece a test o training.
   3. **Estructuración en DataFrame:** Los datos extraídos se almacenaron en un DataFrame de Pandas.
   4. **Exportación:** El DataFrame se guardó en un archivo CSV llamado corpus_sin_procesar.csv.

   Decisiones tomadas:

   - **Uso de Pandas para almacenamiento:** Se decidió utilizar Pandas debido a su facilidad para manejar grandes volúmenes de datos y su flexibilidad para manipular estructuras tabulares.
   - **Estrategia de lectura:** Los errores en la lectura de archivos se gestionaron utilizando el parámetro errors='ignore'.

   Resultados:

   - Un archivo corpus_sin_procesar.csv con todas las noticias del corpus unificadas y categorizadas.

   2.2. Preprocesamiento

   Objetivo: Transformar los textos en un formato homogéneo, eliminando ruido y redundancias.

   Procesos realizados:

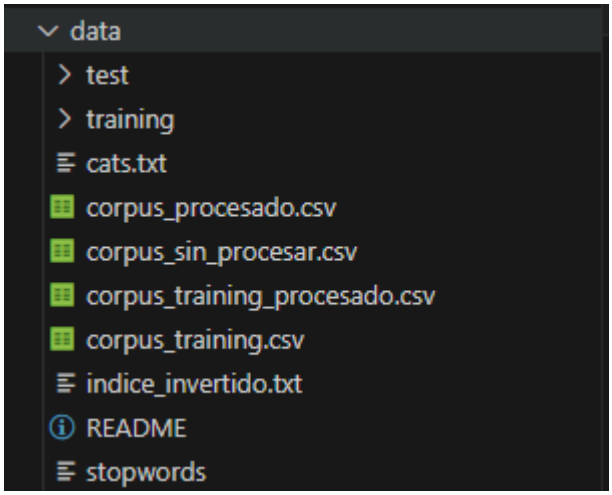
   1. **Conversión a minúsculas:** Unificar el formato de las palabras.
   2. **Eliminación de caracteres no deseados:** Uso de expresiones regulares para eliminar números, signos de puntuación y espacios redundantes.
   3. **Tokenización:** Dividir los textos en palabras individuales o tokens.
   4. **Eliminación de *stop words* y lematización:** Se creó una lista personalizada de *stop words* y se implementó lematización usando el lematizador de NLTK para reducir las palabras a su forma base.
   5. **Pos-tagging (Etiquetado gramatical):** Se aplicó un etiquetado gramatical POS (Part-Of-Speech) para identificar la categoría gramatical de cada palabra, lo que mejoró la precisión de la lematización.

   Decisiones tomadas:

- **Uso de NLTK:** Escogido por su robustez y soporte para múltiples tareas de procesamiento de lenguaje natural.
- **Filtrado de stop words:** Se decidió cargar los stop words desde un archivo externo para garantizar que se usara un conjunto adecuado y fácil de modificar.

Resultados:

- Un archivo corpus_procesado.csv con los textos preprocesados listos para su análisis y un nuevo archivo corpus_training_procesado.csv.



2.3. Representación de Datos en Espacio Vectorial

Objetivo: Convertir los textos en vectores numéricos para facilitar su comparación matemática.

Procesos realizados:

Se implementaron tres métodos principales:

1. **Bag of Words (BoW):** Representa los textos contando la frecuencia de cada palabra.
2. **TF-IDF:** Calcula la importancia relativa de las palabras considerando su frecuencia en un documento y en el corpus completo.
3. **Word2Vec:** Utiliza un modelo preentrenado para asignar vectores semánticos a las palabras.

Resultados:

- Matrices de vectores para cada técnica, listas para ser utilizadas en búsquedas y análisis.

2.4. Indexación

Objetivo:

Facilitar búsquedas rápidas mediante un índice invertido que mapea palabras clave a los documentos donde aparecen.

Procesos realizados:

1. **Construcción del índice:** Se creó un diccionario donde cada palabra clave apunta a los documentos que la contienen, junto con su frecuencia.
2. **Exportación del índice:** Se guardó en un archivo de texto plano (indice_invertido.txt) para facilitar su uso futuro.

Decisiones tomadas:

- **Filtrado de términos:** Se limitó el índice a palabras con tres o más caracteres para evitar que se incluyeran términos irrelevantes o demasiado frecuentes.

Resultados:

- Un archivo indice_invertido.txt fue creado, conteniendo un índice invertido que mapea cada término a los documentos en los que aparece, junto con su frecuencia.

2.5. Diseño del Motor de Búsqueda

Objetivo: Implementar la funcionalidad de búsqueda.

Procesos realizados:

1. **Carga de datos y recursos:**
   - Lectura del corpus preprocesado desde corpus_procesado.csv y carga del índice invertido desde indice_invertido.txt.
   - Configuración de stopwords y signos de puntuación mediante funciones auxiliares.
2. **Vectorización del corpus:**
   Se aplicaron tres técnicas para representar los textos:
   - **Bag of Words (BoW)**: Conteo de la frecuencia de palabras.
   - **TF-IDF**: Cálculo de la relevancia de palabras considerando todo el corpus.
   - **Word2Vec**: Uso del modelo preentrenado word2vec-google-news-300 para vectores semánticos.
3. **Procesamiento de consultas:**
   - Las consultas se preprocesan para eliminar ruido y normalizar términos.
   - Dependiendo del método elegido, la consulta es vectorizada para comparación.
4. **Búsqueda y ranking:**
   - **Similitud coseno** utilizada para comparar la consulta vectorizada con los documentos.
   - Para el índice invertido, se devuelve la lista de documentos donde aparece el término.
5. **Interfaz interactiva:**
   - Menú para seleccionar entre métodos de búsqueda: TF-IDF, BoW, Word2Vec o índice invertido.
   - Muestra los resultados relevantes ordenados por relevancia.

Decisiones tomadas:

- **Múltiples métodos de búsqueda:** Se permiten opciones flexibles para comparar distintas técnicas de recuperación.
- **Similitud coseno:** Elegida por su eficacia para medir la relevancia entre vectores.
- **Índice invertido para términos únicos:** Ideal para búsquedas rápidas y eficientes por palabras clave.

Resultados:

- Recuperación efectiva y ordenada de documentos relevantes.
- Flexibilidad en la selección del método de búsqueda según las necesidades del usuario.
- Preprocesamiento consistente que garantiza precisión en las búsquedas.

Representación de motor de búsqueda:

```
|============== BUSCADOR DE PRUEBA ==============|

1 --> Utilizar TF-IDF para búsqueda
2 --> Utilizar Bag Of Words para búsqueda
3 --> Utilizar Word2Vec para búsqueda
4 --> Utilizar índice invertido
0 --> Salir del buscador

Nota --> Indice invertido debe recibir una única palabra para buscar
Si no es el caso, se usará la primera palabra válida de la oración

Escribe una de las opciones para comenzar:   indice_invertido
```

```
A continuación, escriba su consulta:   trading
Documento con ID: 5376:
TRADE INTERESTS READY FOR FIGHT IN U.S. CONGRESS
  U.S. lawmakers are gearing up for a
  showdown between protectionists and free traders as a major
  trade bill winds its way through committees to a vote by the
  full House of Representatives in late April.
      In a move to toughen U.S. enforcement of trade laws, a key
  House subcommittee last week approved a toned down version of
  legislation to require President Reagan to retaliate against
  foreign countries that follow unfair trade practices.
      This bill will be the cornerstone of congressional efforts
  to restore competitiveness of American industries and turn
  around last year's record 169 billion dlrs trade deficit.
      Several lawmakers have argued the new trade bill made too
  many concessions to Reagan and said they intend to back
  amendments to "get tough" with countries that violate trade
  agreements or keep out U.S. products.
       On the other hand, congressmen known for their allegiance
  to free trade, said the bill ties Reagan's hands too much in
  trade disputes and they will seek to restore his negotiating
  powers.
      Republican Bill Frenzel of Michigan said the subcommittee's
  bill was not one "that a free trader like me could endorse in
  all respects," but he emphasized there was a consensus among
  trade lawmakers to work toward a bill Reagan and Republicans
  would ultimately endorse.
```

2.6. Evaluación del Sistema

Objetivo: Medir la efectividad del sistema.

Decisiones Tomadas:

- **Métricas seleccionadas:**
  - **Precisión:** Mide qué porcentaje de los resultados recuperados es relevante.
  - **Recall:** Evalúa qué porcentaje de documentos relevantes fue recuperado.
  - **F1-Score:** Combina precisión y recall para proporcionar una métrica equilibrada.
- **Conjunto de prueba:** Se utilizó una muestra del corpus_training_preprocesado para evaluar las métricas.

Resultados de Evaluación:

Consulta: *"problems with trading on united states of america "*

Métrica	BoW	TF-IDF	Word2Vec
Precisión	0.5804	0.5804	0.5729
Recall	0.1385	0.1385	0.7955
F1-Score	0.2236	0.2236	0.6661

Análisis:

- **BoW (Bag of Words**): Es efectivo en términos de precisión. Es adecuado para consultas simples.
- **TF-IDF:** TF-IDF tiene la misma precisión que BoW, es útil para consultas donde la importancia relativa de las palabras es clave.
- **Word2Vec:** Word2Vec destaca por su alto recall, lo que le permite recuperar una mayor proporción de documentos relevantes, su mejor F1-Score lo hace la opción más equilibrada y eficaz, especialmente para consultas semánticamente complejas.

Nota importante: Es importante resaltar que para la obtención de documentos relevantes para realizar las métricas se utilizó los documentos de la carpeta training esto puede afectar las métricas ya que para los documentos que se recuperan continente tanto la carpeta test como training

2.7. Interfaz Web de Usuario

Objetivo: Crear una interfaz para interactuar con el sistema.

1. **Interfaz web:** Página principal con formulario para ingresar consultas y seleccionar métodos de búsqueda.
2. **Procesamiento de consultas:** Normalización de texto con `preprocesar_contenido`.
3. **Búsqueda eficiente:** Implementación de TF-IDF, BoW, Word2Vec y búsquedas en índice invertido.
4. **Visualización de resultados:** Paginación y evaluación de precisión, recall y F1-score.

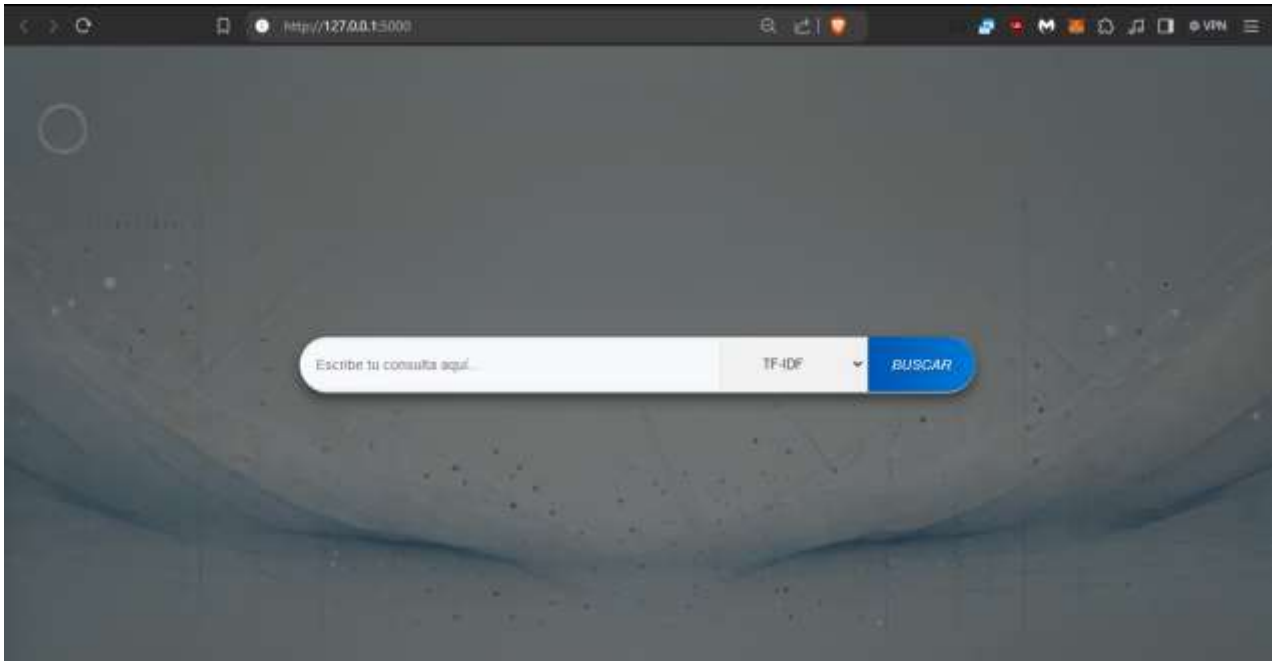
Decisiones tomadas:

- **Uso de Flask:** Framework elegido por su simplicidad y eficiencia en la creación de aplicaciones web.
- **Vectorización flexible:** Se implementaron múltiples métodos de vectorización (BoW, TF-IDF, Word2Vec) para cubrir diferentes necesidades de búsqueda.
- **Paginación de resultados:** Se implementó la funcionalidad para mostrar resultados por página y mejorar la experiencia del usuario.

Resultados:

- Interfaz funcional para búsquedas con resultados paginados.

Ventana de inicio:



Ventana de búsqueda: