Tecnológico de Monterrey

Evidencia 1 - Fase 2

Parte A: Diseño Conceptual

Profesores:

Ariel Ortiz Ramírez

Jorge Adolfo Ramírez Uresti

Miguel González Mendoza

Raul Monroy Borja

Equipo 3:

Jorge Alberto Penagos Méndez	A01378450
Luna Abril Gallegos Espíndola	A01751117
Paulina Guadalupe Alva Martinez	A01750624

PARTE A

1. **Descripción del modelo de solución propuesto.**

Procesamiento de los datos de entrada

- Eliminación de palabras repetidas y vacías (y, el, la, los, las, etc)
    - Esto con el fin de reducir el volumen de datos para analizar, en el proceso de normalización de datos tendremos una lista más simplificada y estandarizada para hacer la comparación y así reducir la complejidad computacional.
- Eliminación de signos de puntuación
    - Son datos que son irrelevantes para el análisis de reutilización de texto, esto debido a que los signos de puntuación siempre estarán presentes en un texto y no brindan información relevante para comprobar nuestra búsqueda.
- Conversión a minúsculas.
- Eliminación de preposiciones y conectores para evitar redundancias absurdas.
    - Esto con el fin de evitar errores a futuro y que la información pueda ser trabajada siguiendo la idea de que ambos textos se encuentran bajo las mismas características.
- Tokenización de las palabras
    - Para facilitar el proceso de normalización
- Normalización de las palabras
    - Regresar la forma base de cada palabra

Métodos y herramientas

- Se usarán gramáticas en regex para la eliminación de signos de puntuación
- Se usará la biblioteca NLTK (Natural Language Toolkit) se emplea para realizar stemming, que es el proceso de reducir las palabras a su forma base o raíz.

- Se planea implementar n-gramas para el procesamiento de las palabras del texto. La elección entre unigramas y trigramas dependerá de la cantidad de datos a trabajar. Los unigramas podrían ser más eficientes en términos de procesamiento, mientras que los trigramas pueden capturar mejor el contexto del texto.
- Para identificar el porcentaje de similitud se usará la distancia de cosenos.
- Se implementarán gráficos usando Matplotlib para mostrar los resultados.

2. Listado de funciones o componentes principales (módulos).

Se planean usar las siguientes funciones descritas:

- Función que extrae el texto o textos
    - Esta función se encargará de leer los documentos de texto de entrada y almacenarlos en una estructura de conjunto (set) para evitar la duplicación de palabras.
- Función que procesa el texto o textos
    - Aquí se llevará a cabo la limpieza y normalización del texto, incluyendo la eliminación de palabras vacías y signos de puntuación, la conversión a minúsculas, el stemming y la tokenización.
- Función que calcula la similitud de textos
    - Se realizan n-gramas y distancia del coseno para buscar la similitud entre los textos .
- Función que describe el comportamiento y otorga los resultados a txt
    - Esta función calculará la similitud entre los textos utilizando n-gramas y la distancia del coseno.
- Función que grafica los resultados de la función que calcula la similitud
    - Generará un archivo de texto que contenga el texto original junto con el porcentaje de similitud entre los textos. Además, creará gráficos visuales que representen la similitud entre los textos.

Las librerías planteadas para uso son:

- Re (REGEX): Para el procesamiento de texto basado en patrones.

- Numpy: Para operaciones numéricas.
- Matplotlib: Para la visualización de datos.
- nltk (Natural Language Toolkit): Para realizar stemming y otras tareas de procesamiento de lenguaje natural.
- sklearn (scikit-learn): Para el cálculo de la distancia del coseno.

3. **Descripción de relaciones y dependencias entre- e intra-módulos.**

   Las funciones dependen de si para la transformación de datos y extracción de textos, para poder procesar y transformar los datos, las funciones reciben texto y en caso de querer hacer múltiples comparaciones se transformarán en textos. Por ejemplo, la función de procesamiento de texto alimenta la función de cálculo de similitud, que a su vez proporciona datos a la función de generación de resultados, la cual se encarga de llamar a la función que genera el documento de salida txt.

4. **Descripción de los datos de entrada y datos resultado.**

Descripción de los datos de entrada

Los datos de entrada cumplirán las funciones clave que se nos solicitan, es decir, en primer lugar recibiremos un par de documentos de texto para comparación, en segundo lugar recibiremos varios documentos de texto en el cual compararemos uno solo contra un conjunto y por último recibiremos dos conjuntos de texto que se comparan entre sí buscando la similitud.

- Se recibirán documentos txt con los textos que se desean comparar para la similitud

Descripción de los datos de salida

Como datos de salida regresaremos un archivo de texto que contendrá el texto y el porcentaje de similitud entre los mismos, además también recibiremos una gráfica que modele la similitud entre los textos ingresados en nuestros datos de entrada.