082057 – Procesamiento del Lenguaje Natural
Trabajo Práctico Individual
CODIGO: 01 – Plagio

Autor: Gianpier Yupanqui Salvatierra

Legajo: 1592075

CONSIGNA
Se pide desarrollar un sistema que pueda detectar **el nivel de plagio** de un Trabajo Práctico **T**, presentado por un alumno **X**, incluyendo la posibilidad de detectar **parafraseos** que no hayan sido debidamente **citados**.

El código deberá ser escrito en Python 3. Se pueden usar todas las librerías que se crean necesarias.

Al ejecutar el código deberá recibir como entrada un documento de texto y devolverá en
texto la salida:
- Nombre del archivo de texto procesado
- Nombre y Apellido del alumno
- Tópico o Tema del texto procesado
- Porcentaje de plagio del texto en general
- Listado de frases (y su ubicación dentro del texto) que podrían ser plagios de otros
TPs previamente subido (fase de entrenamiento) o bien copiados de la web.

Desarrollo

Preparación de Datos
Disponemos de un dataset variado (archivos docx, pdf, pptx) el cual extraeremos su información en forma de oraciones. Cada documento será tratado como una lista de oraciones. Para esto usamos *textract* [1]
Limpiaremos los datos, quitando caracteres inválidos, saltos de líneas, etc.
Para tokenisar por oración usaremos *sent_tokenize* de nltk [2]
Eliminar las **stop words**(Palabras sin significado como artículos, pronombres, preposiciones, etc) en español con la librería *sklearn* [6]
Haremos esto con todos los archivos del corpus y el archivo a analizar

Análisis del grado de similitud entre dos oraciones
Lematizamos (llevar la palabra a su raiz) con *Spacy* [3]
Seteamos la lista para no tener repetidos [3]
Entre más cerca más parecidas serán las oraciones, como saber cuan cerca esta una oración de otra.
Con una métrica (en este caso distancia)
Para medir el grado de **similitud** disponemos de distintas distancias a considerar (*Haming*, *Coseno*, *Jaccard*, *Levenshtein*, etc.)
Una restricción de la distancia de *Hamming* es que las cadenas para ser comparadas deben ser del mismo tamaño.[4]
En este ejemplo usaremos Coseno [5]

Desarrollo
Una vez definido las funciones
- *preparar_datos()*
- *stem_analyze()*
- *similitud_coseno(X_set,Y_set)*

Crearemos la función *plagio(path, confianza)* que Comparamos cada oración del doc a analizar con cada oración de cada uno de los archivos del corpus
Primero los lematiza, Calcular si la similitud es mayor al grado
Devolver la ubicación de la oración y el contenido solo si el grado de similitud calculado es mayor al que se puso en el parámetro confianza

Referencias
[1] Issues Textract
https://github.com/deanmalmgren/textract/issues/194

[2] Documentation nltk.tokenize package
https://www.nltk.org/api/nltk.tokenize.html

[3] TP3 – PLN Lab3
https://github.com/gianpieryup/PLN-2022-1C/blob/main/Labos/082057%20-%20NLP%20-%20UTN%20-%20Lab3.pdf

[4] Diego Campos Sobrino
https://medium.com/@diego.campos.sobrino/m%C3%A9tricas-de-similitud-para-cadenas-de-texto-parte-ii-m%C3%A9tricas-basadas-en-operaciones-de-edici%C3%B3n-af9c8aa71bdb

[5] ACERVO LIMA
https://es.acervolima.com/python-mide-la-similitud-entre-dos-oraciones-usando-la-similitud-del-coseno/

[6] Juan Gabriel Gomila Salas (2021) Repositorio del Curso Machine Learning de A a la Z: R y Python para Data Science
https://bit.ly/3u0Eb8s