Trabajo práctico individual – detección de plagio

Alumno: Tomás Julián Antonio

Docente: Hernán Borré

Introducción

El presente trabajo consiste en la construcción de un programa que, dado un archivo como entrada, retorne como salida posibles textos que hayan sido plagiados en éste, a partir de un dataset inicial. A su vez, comprende el etiquetado del texto según su tópico. El desarrollo de la solución utiliza como herramientas el lenguaje de programación Python, el SGBD Microsoft SQL Server, el software de control de versiones Git y el gestor de paquetes Anaconda.

Detección de plagio

Dada la ruta al archivo, obtenida de la línea de comando, se obtiene el texto utilizando la biblioteca Textract. Se realiza la misma transformación sobre los textos del dataset. Por cada archivo del dataset, se obtiene la similitud coseno entre ambos textos, definida como el valor del coseno entre los vectores formados por c/texto, siendo cada palabra (sin stop words) una dimensión de un espacio vectorial multidimensional en el que se definen vectores. Cuanto mayor sea la similitud coseno, se considera que más cercanos son los textos; es una buena medición para estos fines puesto que, a diferencia de la distancia euclídea, tiende a despreciar el número de ocurrencias de cada palabra, siendo buena para encontrar similitudes entre textos de diferente longitud pero composición parecida. La similitud coseno se calcula vía la biblioteca Scikit Learn.

Se establece como benchmark de plagio una similitud coseno de 0,9. Una limitación del modelo es que los textos contienen no solo las respuestas, sino también las consignas, que son idénticas en muchos de ellos. Esto implica que la similitud detectada es mucho mayor que si sólo se hubiesen tenido en cuenta las respuestas.

Etiquetado de tópico

En la fase previa al entrenamiento, se etiquetó a mano cada texto del dataset con uno de once tópicos. Seguidamente, se construyó una BD relacional que contiene qué textos pertenecen a cada tópico (ingresando un 77% del dataset original), y también la cantidad de ocurrencias de cada token en cada tópico. Los tokens, antes de ser ingresados en la BD, pasan por un proceso de stemming y eliminación de stop words utilizando la biblioteca Natural Language Toolkit.

Al ingresar un nuevo texto, se realiza el análisis de stemming y eliminación de stop words, y luego se contrastan los tokens de él contra la información de la BD. Mediante Naïve Bayes, se encuentra el tópico con más alta probabilidad, con base en la prior probability y las feature likelihoods (siendo cada palabra una feature). Para este análisis, se asume independencia condicional entre las palabras, tanto del texto ingresado como de los de la fase de entrenamiento. Se utiliza la biblioteca NumPy para el cálculo del logaritmo de las probabilidades, evitando errores de underflow.