Procesamiento del lenguaje natural Trabajo Práctico Individual: Plagio

Ivan Casanova

July 2020

1 Introducción

En el presente trabajo, se describirá como se desarrollo una aplicación para verificar el nivel de plagio de un paper/essay en base a un dataset utilizando técnicas de procesamiento de lenguaje natural con el objetivo de crear una herramienta eficaz y certera para la detección de plagio.

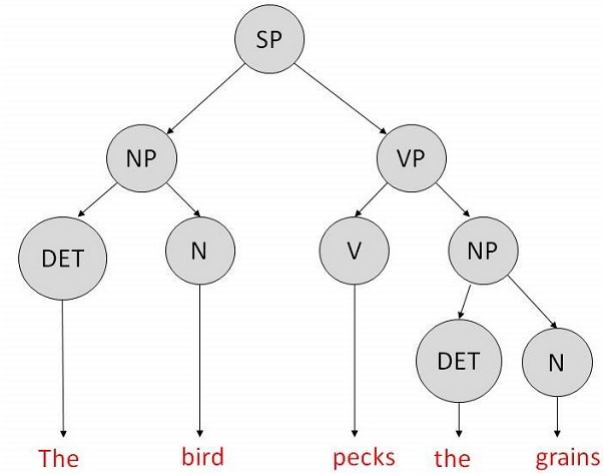

Figure 1: Árbol de parseo de procesamiento del lenguaje natural

2 Conceptos para el Procesamiento Previo de los Datos

Se procede a realizar simplificaciones y preparatorias de los lotes de datos. Este proceso permite abordar la entrada de información netamente útil en el marco de trabajo dispuesto, así poder focalizar en aquello de lo cual se extraerán

conclusiones o resultados. Se trata de una normalización del conjunto de datos. Informalmente se lo puede denominar remoción de ruido al proceso por el cual se eliminan datos carecientes de utilidad en el dominio.

2.1 NLTK

El kit de herramientas de lenguaje natural, o más comúnmente NLTK, es un conjunto de bibliotecas y programas para el procesamiento del lenguaje natural (PLN) simbólico y estadísticos para el lenguaje de programación Python. NLTK incluye demostraciones gráficas y datos de muestra [1]. En el presente trabajo practico se utilizo NLTK mediante la librería text-matcher que hablaremos a a continuación en la cual utilizamos para la tokenizacion, distancia y calculo de radio de las palabras que se encuentran en los textos.

2.2 Tokenización

El proceso de tokenización refiere a la división de cadenas de texto de considerable longitud en pequeñas piezas, denominadas tokens. Puesto que los trozos de texto extensos pueden ser convertidos en oraciones, las oraciones pueden subdividirse a su vez en palabras. Puede considerarse la tokenización como el proceso de segmentación de texto o análisis léxico, en un enfoque que se produce de forma exclusiva en palabras [4].

2.3 Amazon Comprehend

Amazon Comprehend es un servicio de procesamiento de lenguaje natural (NLP) que usa el aprendizaje automático para encontrar información y relaciones en textos. No se requiere experiencia en aprendizaje automático. En el presente

Figure 2: Amazon SageMaker Logo

trabajo se utilizo Amazon Comprehend para realizar una análisis de los textos pudiendo obtener elementos clave para su análisis. Entre dichos elementos se encuentran: autores, tópicos y lenguaje. Amazon Comprehend utiliza distintas técnicas de natural language processing (NLP) para detectar distintos elementos sintácticos del texto.

2.4 Similitud coseno

La similitud coseno es una medida de la similitud existente entre dos vectores en un espacio que posee un producto interior con el que se evalúa el valor del coseno

del ángulo comprendido entre ellos. Esta función trigonométrica proporciona un valor igual a 1 si el ángulo comprendido es cero, es decir si ambos vectores apuntan a un mismo lugar. Cualquier ángulo existente entre los vectores, el coseno arrojaría un valor inferior a uno. Si los vectores fuesen ortogonales el coseno se anularía, y si apuntasen en sentido contrario su valor sería -1. De esta forma, el valor de esta métrica se encuentra entre -1 y 1, es decir en el intervalo cerrado [-1,1].

Esta distancia se emplea frecuentemente en la búsqueda y recuperación de información representando las palabras (o documento) en un espacio vectorial En minería de textos se aplica la similitud coseno con el objeto de establecer una métrica de semejanza entre textos En minería de datos se suele emplear como un indicador de cohesión de clústeres de textos. La similitud coseno no debe ser considerada como una métrica debido a que no cumple la desigualdad triangular [5].

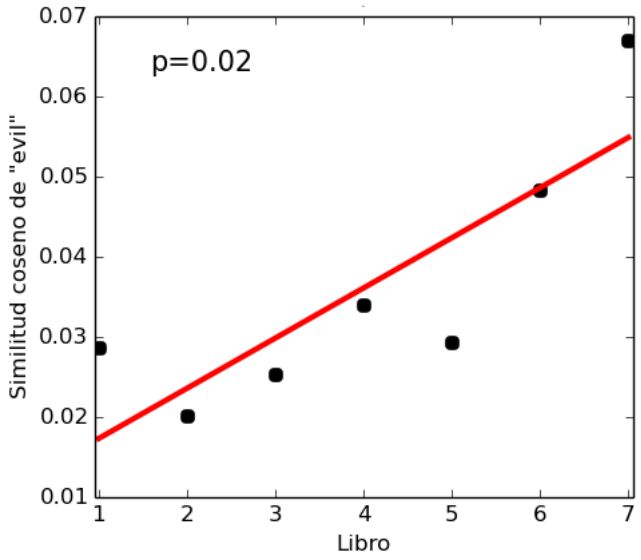

Figure 3: Grafico de similitud coseno

2.4.1 Similitud Coseno Suave

El Coseno Suave [7] es una medida de similitud "suave" entre dos vectores, es decir, la medida considera la similitud entre pares de características. La similitud coseno tradicional considera que las características en el modelo espacio vectorial (MEV) son independientes o completamente diferentes, mientras que el coseno suave propone considerar la similitud de características en el MEV, lo cual permite la generalización de los conceptos de similitud coseno y también la idea de similitud (similitud suave).

Por ejemplo, en el área de procesamiento de lenguaje natural (PLN) la similitud entre las características es bastante intuitiva. Las características tales como, palabras, n-gramas, n-gramas sintácticos4 pueden ser muy similares, aunque formalmente son consideradas como características diferentes en el MEV. Por ejemplo, las palabras "play" y "game" (en inglés) son palabras diferentes y por lo tanto se mapean a dimensiones diferentes en el modelo de espacio vectorial; sin embargo, es obvio que estas palabras están relacionadas semánticamente. En el caso de n-gramas o n-gramas sintácticos se puede usar la distancia de Levenshtein para calcular la similitud entre características.

Para el cálculo del coseno suave, se introduce la matriz s que contiene la similitud entre las características. Se puede calcular utilizando la distancia Levenshtein u otras medidas de similitud, por ejemplo, diversas medidas de similitud de WordNet. Luego solo se multiplica por esta matriz.

Dado dos vectores a y b de dimensión N, el coseno suave es calculado como sigue:

$$soft\_cosine_1(a, b) = \frac{\sum_{i,j}^{N} s_{ij} a_i b_j}{\sqrt{\sum_{i,j}^{N} s_{ij} a_i a_j} \sqrt{\sum_{i,j}^{N} s_{ij} b_i b_j}},$$

Si no existe similitud entre características (sii = 1, sij = 0 para i j), la ecuación dada es equivalente a la fórmula de similitud coseno convencional.

La complejidad de esta medida es cuadrática, lo cual la hace completamente aplicable a problemas del mundo real. La complejidad incluso puede ser transformada a lineal.

3 Características Heurísticas de Evaluación

A lo largo del desarrollo, se opta por realizar un análisis de los ensayos bajo un grupo de características que se detallan a continuación:

3.1 Titulo

Para obtener el titulo del documento se utilizo al librería os del archivo y mediante la función split() se corto el path del archivo para obtener únicamente su nombre. De esta manera, fue posible obtener el titulo del archivo para poder cargarlo en el JSON de resultados.

3.2 Posibles Autores

Los posibles autores, se obtienen mediante Amazon Comprehend. Para lograr dicha tarea Amazon Comprehend reconoce los nombres propios del texto, permitiendo reconocer así, posibles autores del texto. No es posible reconocer el concepto de "Autor" ya que a nivel automático es difícil detectar quien realmente escribió el articulo.

3.3 Porcentaje de plagio

Para calcular el porcentaje de plagio se utilizo una ecuación de porcentajes simple en la cual, basándose en las palabras "plagiadas" se divide por la cantidad de palabras totales del texto. Este porcentaje tiene un factor de error en el cual se tiene en cuenta el caso en los que las frases no sean plagiadas cuando realmente no lo son.

3.4 Tópicos o Temas

Los temas del texto fueron encontrados utilizando Amazon Comprehend. Al igual que con los Posibles autores, se estimaron que los valores principales o mas repetidos del texto pueden ser considerados como Temas o Tópicos del texto. Se espera a futuro realizar alguna tarea de labeling [3] para poder reconocer los tópicos y poder entrenar el sistema para poder reconocerlo.

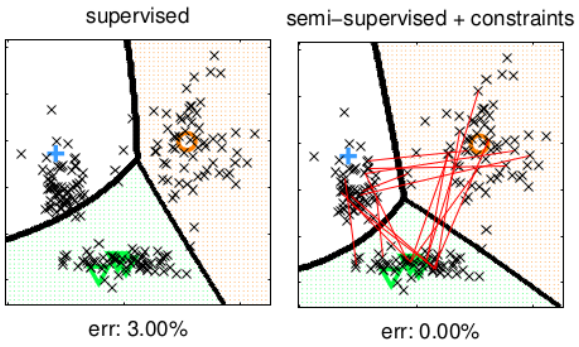

Figure 4: Entrenamiento supervisado vs semi supervisado

3.4.1 Similitud Coseno Suave

4 Herramientas

4.1 Almacenamiento de dataset

El almacenamiento del dataset se realizo mediante Amazon Simple Storage Service (S3) [2] en la cual se cargaron todos los documentos del dataset para poder generar el corpus con el que se usara detectar el plagio. Para eso, se ingreso a la consola de Amazon Web Services y se cargaron los archivos directamente desde allí. Finalmente, desde la aplicación se lo consume utilizando la librería boto3, leyendo la ruta de la carpeta en la cual se encuentran los datos. Dichos datos, compuestos de archivos del tipo PDF, DOCX, DOC, etc, son analizados mediante distintas librerías y convertidas a un archivo txt el cual mas adelante sera utilizado para detectar plagios. Se descarga todos los archivos del s3 en

Figure 5: Amazon S3

una carpeta temporal (./tmp) en la cual son directamente convertidos en un solo archivo corpus con todos sus palabras y caracteres.

4.2 Análisis de Plagio

Para el análisis de plagio se utilizo text-matcher [6]. Es una herramienta simple de detección de similitudes en archivos entre 2 textos. Para encarar el problema se tomo como corpus a texto principal un txt de todos los documentos del dataset. Luego el usuario puede elegir entre los distintos documentos para detectar si esta mal citado o si se plagio de otro documento/paper. También aportara todos los valores previamente comentados.

5 Implementación

La implementación se realizo de la siguiente manera: 1) Primero se descarga el dataset desde el bucket de S3 para obtener a nivel local todos los archivos. Se los parsea a un formato común, en este caso txt en el cual se añaden a un archivo (corpus.txt) todos sus caracteres. Se borran los archivos terminado este proceso para no generar espacio de mas. 2) Luego se elige un documento para detectar el nivel de plagio que posee. Se utiliza text-matcher para detectar entre el archivo corpus que contiene todos los documentos del dataset y el archivo elegido el cual también es convertido a txt para poder ser utilizado. 3) Finalmente, se genera un JSON con todos los atributos descritos anteriormente para poder ser guardado a futuro en una base NoSQL y utilizado en próximos trabajos.

6 Conclusión

El propósito principal del documento y el desarrollo del trabajo es el aprendizaje tanto con un enfoque empírico como uno teórico. Se logran entender conceptos que abordan el procesamiento del lenguaje natural para llegar a la conclusión de que la complejidad y naturaleza del campo es inmenso. Al encarar el trabajo desde su inicio con características heurísticas como se detalla en el documento previamente y se observa en profundidad en el libro de notas

Jupyter, los números comienzan a alinearse a la realidad. Se comienza a recabar resultados más acertados a instancias de conceptos matemáticos que estudian la combinación de elementos de estructuras abstractas como algebra, que si no fuese por la comprensión de este tipo de herramientas sería más dificultoso poder entender la idea de ellos. Tal es asi el caso de los espacios vectoriales, que en esta ocasión se ven aplicados en text-matcher sobre palabras. Es muy positivo recapacitar y hacer catarsis para darse cuenta que, utilizando conceptos abstractos y avanzados, se puede refinar a niveles muy altos un modelo que evalua texto. Esto conlleva a pensar en qué tantos ámbitos se puede emplear la disciplina del procesamiento del lenguaje natural para perseguir ciertos objetivos que podrían simplificar la vida de muchas personas. Por otra parte el proyecto permite dilucidar la idea de lo complejo que son las estructuras lingüisticas frente a cuestiones propias de los humanos y su expresividad textual, como es el caso de la ambigüedad o sarcasmo, entre muchos otros. Aunque no sea uno de los problemas que se enfrenta el plagio automático de ensayos, no es algo que no esté presente en cualquier desarrollo de texto. Es conveniente tambien destacar la ventaja que permite la realización de este tipo de análisis sobre textos. Se supone el caso de una persona que desea certificarse en un examen internacional de un idioma en particular. ¿Qué tan positivo

puede resultarle a aquella el tener en cuenta cuáles son las características que más atención e importancia prestan los examinadores? Esto podría resultar en precisión a la hora del desarrollo del examen, y por ende en una exitosa certificación, siendo el procesamiento de lenguaje natural una ayuda con la cual no cualquier persona cuenta. Todo lo que se menciona en esta sección ofrece una percepción de lo fundamental de tener un dominio completo sobre la disciplina, y da cuenta de la vasta cantidad de investigación que resta por efectuar. La importancia que esto posee desemboca en un inexorable progreso que llevará al cabo de los años al descubrimiento de aplicaciones matemáticas al aprendizaje de modelos para facilitar la vida de la humanidad en varios aspectos.

References

[1] Ewan Klein; Edward Loper Bird, Steven. *Natural Language Processing with Python*. 2009.

[2] Huijun Huang, Dijiang; Wu. *Mobile Cloud Computing: Foundations and Service Models*. 2017.

[3] Leif Johnson. *What is the difference between labeled and unlabeled data?* 2017.

[4] Mohamed Zakaria Kurdi. *Natural Language Processing and Computational Linguistics: speech, morphology, and syntax, Volume 1*. 2016.

[5] M. Steinbach V. Kumar P.-N. Tan. *Introduction to Data Mining*. 2005.

[6] Jonathan Reeve. Text-matcher. https://github.com/JonathanReeve/text-matcher, 2020.

[7] Alexander; Gómez-Adorno Helena; Pinto David Sidorov, Grigori; Gelbukh. *Soft Similarity and Soft Cosine Measure: Similarity of Features in Vector Space Model*. 2014.