**Reporte Técnico Proyecto Integrador**

**Descripción**

Se tiene un conjunto de 980 papers en ingles con formato txt, se requiere realizar un sistema y/o aplicación para la ingesta, almacenamiento, indexación, búsqueda, recuperación, navegación, visualización de documentos y caracterización por temas afines en los documentos.

**Proceso de limpieza de los papers**

Se debe leer cada uno de los 980 papers para ejecutar el proceso de limpieza, el cual esta compuesto de los siguientes pasos: Tokenización, remoción de caracteres especiales, remoción de Stopwords, conversión de mayúsculas a minúsculas, descartar palabras con un solo termino, utilización del diccionario de inglés, Stemming y lematización, en este mismo proceso se almacena la frecuencia de aparición de las palabras y se asigna el Term Frequency (TF) y elInverse Document Frequency (IDF). Al finalizar el proceso de limpieza de la totalidad de papers se crea el Bag of words con palabras únicas.

**Tokenización:** Este proceso consiste en separar las palabras por espacio y almacenarlas en una lista.

**Stopwords:** Son los artículos o palabras de parada, ejemplo: "a", "por", "el", "la", "los", "y", "o", entre otras.

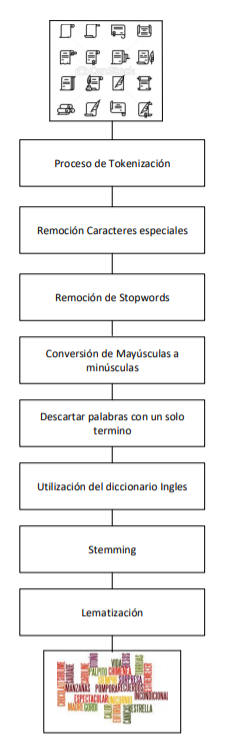
**Stemming:** Es el método para reducir las palabras a su raíz, ejemplo: Cumputer -> Compute, Computation -> Compute, Computing -> Compute.

**Lematización:** Este método consiste en llevar las palabras a singular, los adjetivos los convierte a forma masculino y los verbos los transforma a su forma infinitiva.[[1]](#endnote-1)

**[[2]](#footnote-1)**

**[[3]](#footnote-2)**

**Diagrama de flujo proceso de limpieza**



**Construcción del índice invertido**

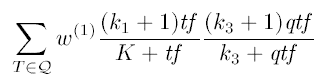
En este proyecto se usó el índice invertido, el método más usado de indexación. Para la creación del índice invertido se definió el diccionario de palabras, el cuál se compone de tres columnas. La primera columna del diccionario contiene cada término individual, la segunda columna contiene la cantidad de documentos donde aparece cada uno de los términos y la tercera columna contiene las frecuencias de dichos términos en total. El posting tiene también tres columnas, una con un ID de cada documento, una con la frecuencia de cada término en cada documento y una con las posiciones respectivas de cada término en los documentos donde se encuentra.

Así, se puede realizar una búsqueda. Después de pasar el proceso de tokenización, los términos de búsqueda pasan por el proceso de índice invertido. El primer paso es buscar las palabras en el diccionario, luego se busca en cuantos documentos está el término de búsqueda con sus frecuencias. Finalmente en el posting, se buscan los ID de los documentos de cada término de búsqueda, y luego se busca la frecuencia de los términos en cada documento y finalmente, se trae cada posición en el documento. De esa forma, se tienen los documentos donde se encuentran los términos de búsqueda y sus posiciones, que es el resultado de la búsqueda.

Para realizar este proceso, se hicieron dos pasos. Primero, pese a tener librerías existentes disponibles para esta técnica, se realizó el cálculo del índice invertido manualmente y segundo, se realizó esta técnica con la librería metapy, que ya tiene esa función disponible. El objetivo de hacer el ejercicio de manera manual y con una librería ya existente era poder realizar una comparación.

**Ránquin**

Una vez se recuperen los documentos de acuerdo a la búsqueda realizada, el siguiente paso es ubicar los documentos resultantes en un determinado orden, según una relevancia, denominado ránquin. En este proyecto se usó el método de ránquin BM25. Este método está basado en modelos probabilísticos de recuperación de información. El sistema BM25 usa esta función simplificada:

, donde

es el Query, que contiene los términos

es el peso Robertson/Spark Jones de en

es el número de documentos en la colección.

es el número de documentos con el término.

es el número de documentos relevantes con el término.

es el número de documentos conocidos no relevantes de un tema específico

es el número de documentos no relevantes del término

es

, , y son parámetros que dependen de la naturaleza de los querys y quizá del dataset.

es el term frequency.

es la frecuencia del término en los temas del Q

y son la longitud del documento y el promedio de longitud del documento en unidades arbitrarias respectivamente.

El método de ránquin más usado es el BM25, y es también usado por la libería Metapy. Para este proyecto se usó el BM25, que se comparó con Metapy.

Se definió la función para calcular el BM25, con los parámetros query, que es la búsqueda realizada, vocabulary es la función vectorizer.vocabulary\_, que es el índice invertido, prom tiene un valor de 27544, el parámetro k1 tiene un valor de 1.2, b tiene un valor de 0.75 y top tiene un valor de 20.

Se realizó una comparación de los resultados del ránquin del BM25 con los del Metapy, que se muestran en la siguiente tabla. Esta comparación es conocida como sensibilidad, donde 1 significa coincidencia perfecta y 0 significa coincidencia nula.[[4]](#footnote-3)

|  |  |  |
| --- | --- | --- |
|  | Query | Sensibilidad |
| 0 | Data Science | 0.65 |
| 0 | Machine Learning | 0.70 |
| 0 | Computer Science | 0.40 |
| 0 | Algorithms In Dynamic Networks | 0.75 |
| 0 | Triangle Free Process | 0.50 |
| 0 | Biology | 0.70 |

**Topic Detection**

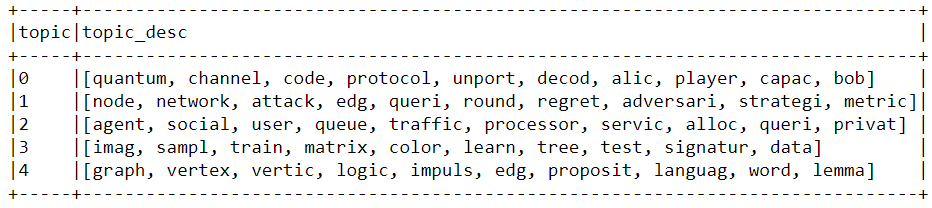
La detección de tópicos en el procesamiento del lenguaje natural es útil para tener una idea de lo que habla un documento, con el fin de descubrir estructuras relacionadas con palabras que aparecen recurrentemente. Por ejemplo, carro y gasolina a parecerá con mayor frecuencia que gasolina y patineta; esto dará a entender que un documento podría tener varios temas, unos más importantes que otros y así dar un espectro general de los tópicos y obtener los documentos que tengan más información del tema que estemos consultando. Esto se realiza a través de modelos estadísticos que nos permite evaluar el balance de temas en cada documento.

Para la realización del Topic detection se utilizó la librería “pyspark” que presenta una forma fácil de realizar el tratamiento de los datos y de calcular el modelo de clasificación LDA; el cual, es un modelo estadístico generativo que permite que los grupos de observaciones explican por qué algunas partes de los datos son similares. Por ejemplo, si las observaciones son palabras recopiladas en documentos, se postula que cada documento es una mezcla de un pequeño número de temas y que la presencia de cada palabra es atribuible a uno de los temas del documento.

Para el ejercicio se realizó la búsqueda de cinco tópicos de la siguiente forma:



El cuál, encontró en los papes los siguientes tópicos:



1. https://en.wikipedia.org/wiki/Latent\_Dirichlet\_allocation [↑](#endnote-ref-1)
2. mcda-sir-02-textretrieval-impl-20191.pdf [↑](#footnote-ref-1)
3. mcda-sir-02-textretrieval-impl-20191.pdf [↑](#footnote-ref-2)
4. Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. M., & Gatford, M. (1995). Okapi at TREC-3. Nist Special Publication Sp, 109, 109. [↑](#footnote-ref-3)