**Informe canalización**

Nicolas mesquiatti

#Aclaracion

Realice el tp en canalizacion.py y canalizacion2.py: Debido al texto que empieza con “corpus = [ lematizar(quitarStopwords\_eng(word\_tokenize("Python...” Y no como un txt normal que veníamos trabajando, en este caso habría que entrar a las funciones, pero no supe cómo hacerlo, así que hice de dos formas en canalizacion.py traje las oraciones solas y las puse en una lista, en canalizacion2.py hice que lea el archivo y solo traiga las oraciones después es igual en ambos archivos

corpus CorpusLenguajes.txt

-Aplicar stop\_word,

-Lematización

-Tf-Idf

-Mostrar el corpus preparado

-Mostrar la matriz TF-IDF generada

-Mostrar el vocabulario generado

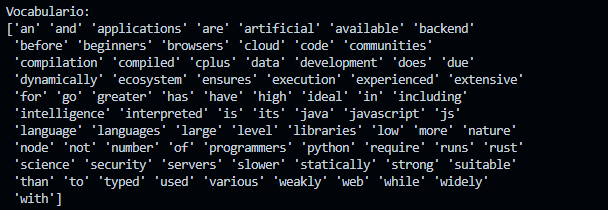
1)Para aplicar el stop Word tuve que tokenizar, luego aplico stop words y despues sin las palabras vacías aplique la lematización

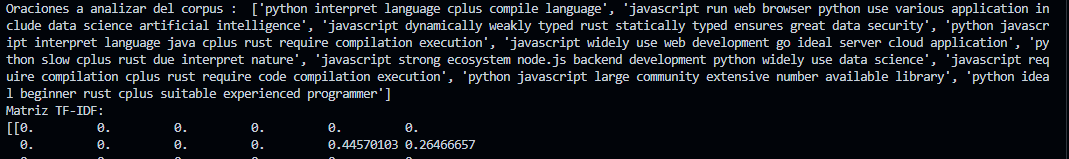
2)Después aplicamos el Tf-Idf donde mi corpus fueron todas las oraciones y con el vocabulario correspondiente, no sé si estuvo bien haber puesto todas las oraciones como corpus, pero si ponía otra cosa me daba mal y todo 0

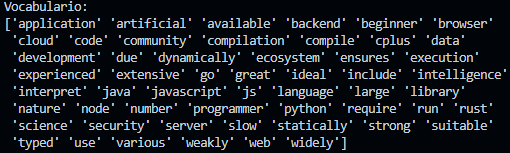
***Análisis de tf-idf***

Anteriormente había realizado mal el tf-idf luego de la clase del miércoles me di cuenta de que mi corpus estaba siendo todo el texto completo sin tokenizar ni quitar nada... Luego vi que había que hacerlo.

Mi vocabulario había quedado de esta manera: con palabras vacías y todo

Luego de arreglar mi código y ver mi error quedo de la siguiente manera:

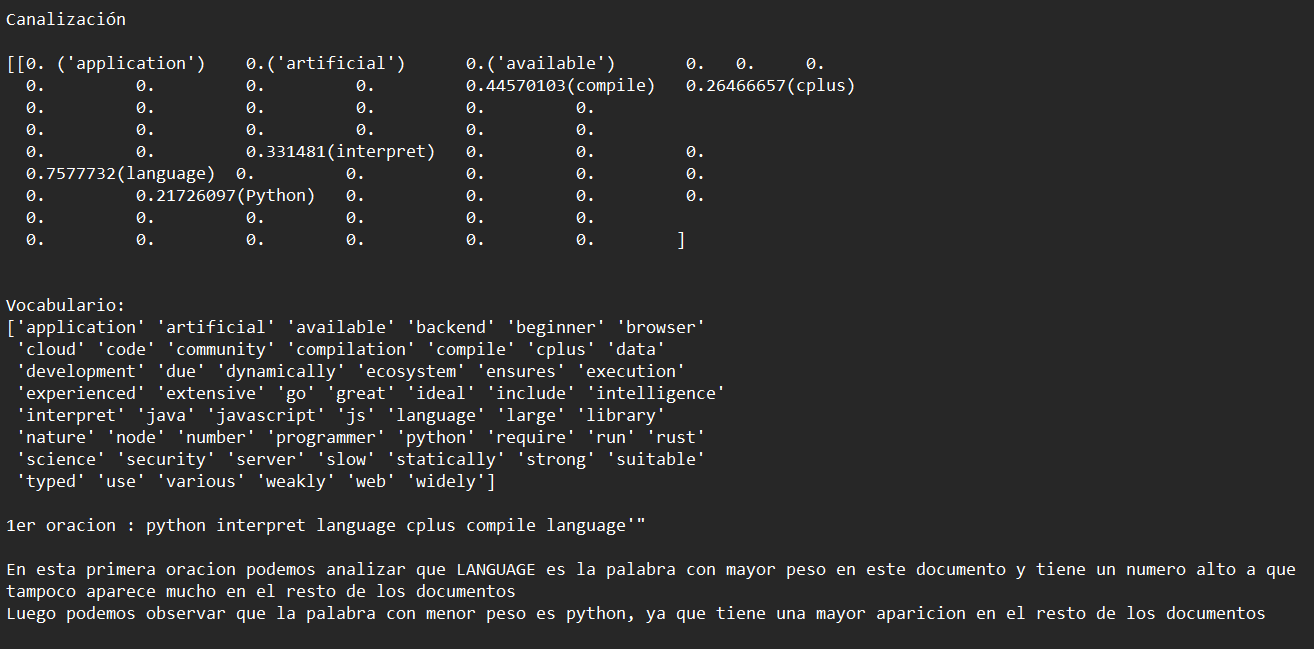
Con las oraciones bien organizadas mi vocabulario es el siguiente, se encuentra ordenado alfabéticamente porque el metodo get\_feature\_names\_out de TfidfVectorizer() lo devuelve por defecto:



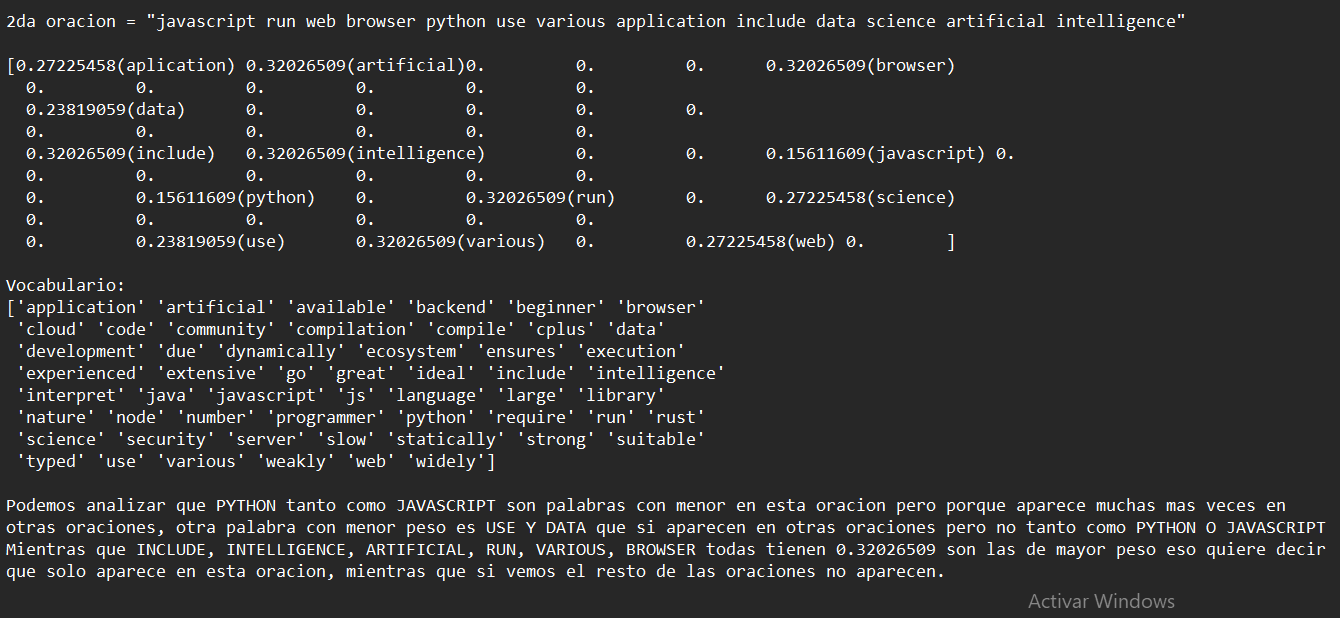
En la siguiente captura se ve un análisis del tf-idf de la primera oración que ya limpiada es “python interpret lenguage cplus compile lenguage” y podemos analizar que

LANGUAGE es la palabra con mayor peso en este documento y tiene un numero alto a que tampoco aparece mucho en el resto de los documentos

Luego podemos observar que la palabra con menor peso es python, ya que tiene una mayor aparición en el resto de los documentos



Si analizamos una SEGUNDA ORACION :



Podemos analizar que PYTHON tanto como JAVASCRIPT son palabras con menor en esta oración, pero porque aparece muchas más veces en otras oraciones, otra palabra con menor peso es USE Y DATA que si aparecen en otras oraciones, pero no tanto como PYTHON O JAVASCRIPT

Mientras que INCLUDE, INTELLIGENCE, ARTIFICIAL, RUN, VARIOUS, BROWSER todas tienen 0.32026509 son las de mayor peso eso quiere decir que solo aparece en esta oración, mientras que si vemos el resto de las oraciones no aparecen.

3)

Analizar el mismo y redactar un informe con las conclusiones obtenidas.

-Obtener las jerarquía de 6 palabras mas usadas en todo el corpus

-La palabra menos utilizada

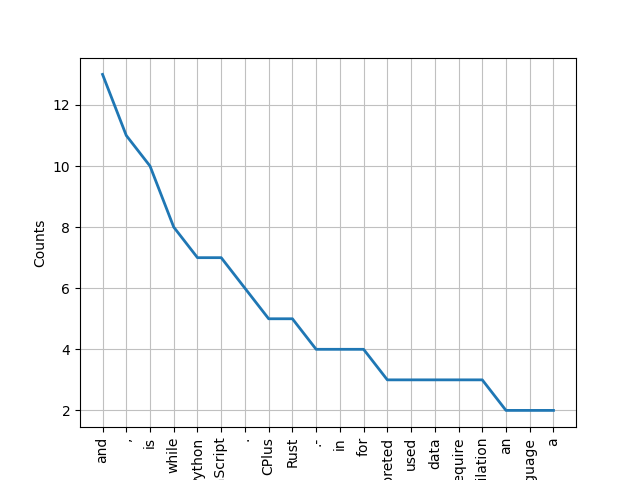
-Las palabras mas repetidas en la misma oración

-Imprimir el gráfico de Distribución de Frecuencia.

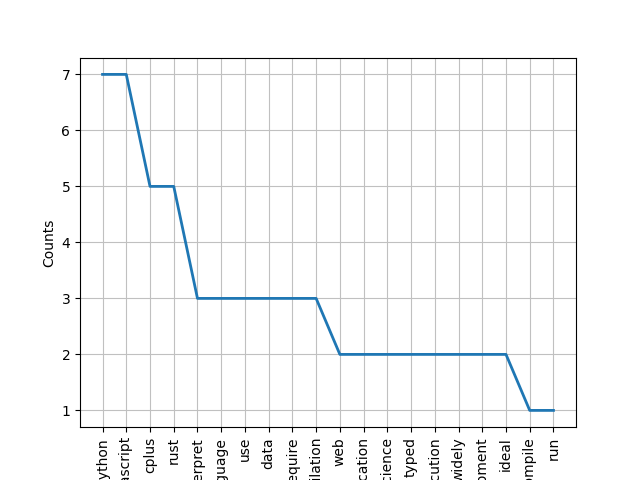
En esta parte hice todo lo más bien, excepto el de las palabras más repetidas en la misma oración, en ese en específico le tuve que pedir ayuda a chat gpt

* La jerarquía de las palabras más usadas en todo el corpus fue: [('Python', 7), ('JavaScript', 7), ('cplus', 5), ('rust', 5), ('interpret', 3), ('language', 3)], esa es la salida que da mi código, ya lematizado y sin stopwords,es algo esperado ya que son oraciones hablando de los lenguajes de programación.
* La palabra menos utilizada la saque utilizando la función MIN a la frecuencia de cada palabra.
* Las palabras más repetidas en la misma oración, la entendí porque palabra aparece más veces en cada oración. Como dije anteriormente le pedí ayuda a chat gpt pero entendí como funciono todo.
* Luego imprimí dos gráficos uno que muestra las palabras más frecuentes sin lematizar ni quitar las palabras vacías y otro ya con el texto limpio

***Gráfico sin quitar las stopwords ni lematizado***



***Gráfico quitando las stopwords y lematizado***



*Debido al grafico nos podemos dar cuenta de la importancia de quitar las palabras vacías y lematizarlo.*  Conclusion : El ejercicio estuvo bueno ya que pudimos aplicar lo que aprendimos, en un principio me fue difícil empezar debido a que no supe cómo tratar el texto, después de eso ya era ver lo que habíamos hecho y entender para que se utiliza. Lo que tendría que repasar sería el tf-idf y ver si hay más ejercicios de este estilo.