# IR systems

# Vector space Model Vs Language Model Vs Okapi BM25

Cranfield collection

*Prepared By*

Raghad Ramazan

Department of computer Engineerıng

Çukurova University

([Raghad.mhd.ramadan@gmail.com](mailto:Raghad.mhd.ramadan@gmail.com))

***Advised by***

Prof.Dr.SELMA AYŞE ÖZEL

**JANUARY 25,2021**

Abstract

In this report I will show a comparison between three different methods of Information Retrieval, In the begging, I started with comparing between vector space model and language model but I found it really interesting so I tried to add a compersion also between them and Okapi BM25, using the Cranfield collection which is the test collection that has 1400 documents and queries to test, in this experience I will test 5 queries using each method when comparing the results together.

First of all, I will give the python codes that I used in my experimentation, talking about each method, then I will discuss the results, in the end, I will compare the results and evaluate the methods by them recalls and precisions.

TABLE OF CONTENTS

[IR systems 1](#_Toc62830445)

[Vector space Model Vs Language Model Vs Okapi BM25 1](#_Toc62830446)

[Cranfield collection 1](#_Toc62830447)

[Abstract i](#_Toc62830448)

[LIST OF FIGURES iii](#_Toc62830449)

[LIST OF TABLES iv](#_Toc62830450)

[1 Vector space model 1](#_Toc62830451)

[1.1 Vector Construction 1](#_Toc62830452)

[1.2 Code plan 1](#_Toc62830453)

[1.3 Imported libraries 2](#_Toc62830454)

[1.4 Collection Preparation 2](#_Toc62830455)

[1.5 Convert to Vectors 3](#_Toc62830456)

[1.6 Processing the Query 3](#_Toc62830457)

[2 Language Model 4](#_Toc62830458)

[2.1 Code plan 4](#_Toc62830459)

[2.1 Imported libraries 5](#_Toc62830460)

[2.2 Collection Preparation 6](#_Toc62830461)

[I done the same opertions that I made to vector space model to the language model. 6](#_Toc62830462)

[2.3 Language Model operations 6](#_Toc62830463)

[2.4 Processing the Query 7](#_Toc62830464)

[3 Okapi BM25 7](#_Toc62830465)

[3.1 Code plan 8](#_Toc62830466)

[3.2 Imported libraries 8](#_Toc62830467)

[3.3 Collection Preparation 8](#_Toc62830468)

[3.4 Okapi BM25 operations 9](#_Toc62830469)

[3.5 Implantation Operations 10](#_Toc62830470)

[4 Results 11](#_Toc62830471)

[5 Conclusion 13](#_Toc62830472)

[6 References 14](#_Toc62830473)

LIST OF FIGURES

[Figure 1 Imported libraries in vector space model code................................................2](#_Toc62830806)

[Figure 2 Collection Preparation……………………………………………………………..2](#_Toc62830807)

[Figure 3 Convert to Vector…………………………………………………………………3](#_Toc62830808)

[Figure 4 Processing the query……………………………………………………………..3](#_Toc62830809)

[Figure 5 Imported libraries for language model………………………………………….5](#_Toc62830810)

[Figure 6 Preprocesses to collection for language model…………………………………..6](#_Toc62830811)

[Figure 7 Language model operations ……………………………………………………..6](#_Toc62830812)

[Figure 8 Preprocessing the query ………………………………………………………….7](#_Toc62830813)

[Figure 9 Imported libraries for Okapi BM25………………………………………………8](#_Toc62830814)

[Figure 10 Collection Preparation for Okapi BM25………………………………………...8](#_Toc62830815)

[Figure 11 Class BM25 and fit function of Okapi BM25………………………………….9](#_Toc62830816)

[Figure 12 Search and Score Functions of Okapi BM25…………………………………10](#_Toc62830817)

[Figure 13 Implantation Okapi BM25……………………………………………………..10](#_Toc62830818)

LIST OF TABLES

[Table 1 Results of query 021 …………………………………………………………….12](#_Toc62830819)

[Table 2 Results of average of Recalls and Precisions…………………………………..12](#_Toc62830820)

# Vector space model

Vector Space Model-Both the query and each document are represented as vectors in the term space. A measure of the similarity between the two vectors is computed.

And it is the basic information retrival system model,so we can understand that all the models became after it .

•Most commonly used strategy is the vector space model (proposed by Salton in 1975).

•Documents are mapped into term vector space.

•Each dimension of the vector represents tf\*idffor one term.

•Queries are treated like documents.

•Documents are ranked by closeness to the query.

•Closeness is determined by a similarity score calculation.

## Vector Construction

•𝑡= number of distinct terms in the document collection

•𝑡𝑓𝑖𝑗= term frequency = number of occurrences of term 𝑡𝑗in document 𝐷𝑖

•𝑑𝑓𝑗= document frequency = number of documents which contain term 𝑡𝑗

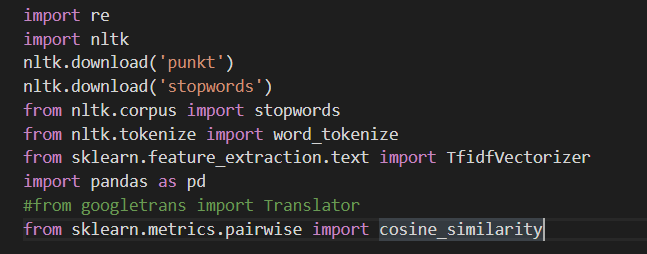
•𝑖𝑑𝑓𝑗= inverse document frequency= log(𝑑𝑑𝑓𝑗)

where d is the total number of documents in the collection.

## Code plan

I used python to write my code, first I will show the libraries that I imported to my project , then I will show the operations taht I made to clean up the collection , then I will talk about the vectorization steps .

## Imported libraries

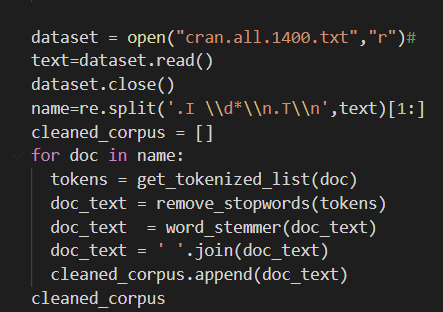


Figure

Imported libraries in vector space model code

In figure 1 we can see that I used nltk, re, sklearn, pandas libraries.

## Collection Preparation

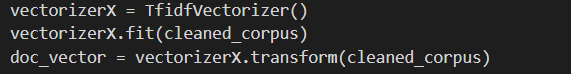


Figure

Collection Preparation

In figure 2 I am showing how I opened the collection and how to split the collection to documents, and then I cleaned the documents, by removing the stop words and converting the words to stemmers, to get the cleaned documents.

## Convert to Vectors



Figure

Convert to Vector

In figure 3 I converted the documents to vector, to complete the steps of vector space model.

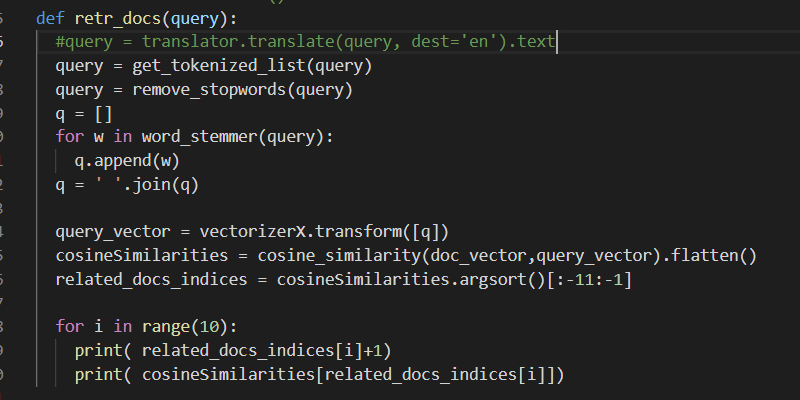
## Processing the Query

retr\_docs('''is it possible to relate the available pressure distributions

for an ogive forebody at zero angle of attack to the lower surface pressures

 of an equivalent ogive forebody at angle of attack .''')

I sent the query to function that I made to process it, the function showing down :



Figure

Processing the query

In the function above I preprocessed the query, then I converted it to vector to calculate the similarity between the query and documents by the cosine similarity, at the end I printed the relevant documents.

I called the function for five queries, to allow me calculating the Average Recalls and Average Precisions.

# Language Model

Language Models: A language model is built for each document, and the likelihood that the document will generate the query is computed.

A statistical language model is a probabilistic mechanism for “generating” a piece of text.

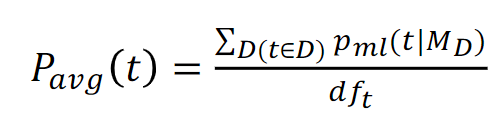
•The core idea is that documents can be ranked on their likelihood of “generating” the query. 𝑆𝐶(Q,Di)=𝑃(𝑄|𝑀Di)

Where 𝑀Di is the language model implicit in document 𝐷..

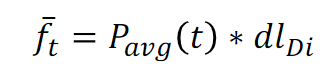
•The presence or absence of any term is modeled as an independent Bernoulli event, and it views the generation of the whole query as a joint event of observing all the query terms and not observing any terms that are not in the query.

## Code plan

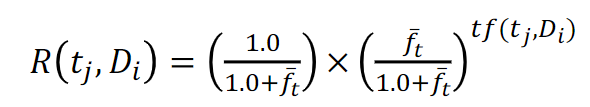
In this model I wrote long code, and I wrote the smoothed language model because I wanted to avoid zeros as much as I can, first I calculate tf(ti,Di) of term t in document Di , then I calculate Maksimum likelihood for each term Pml(tj|MDi)=, then I calculate Average probability for each term is computed as



After that, I calculate mean term frequency for each term in each document Di by using the equation:

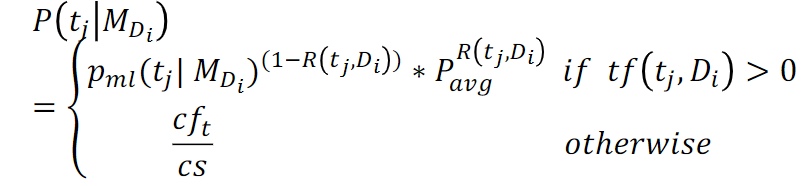


Then I used the following risk function to compute risk of each term in each document.

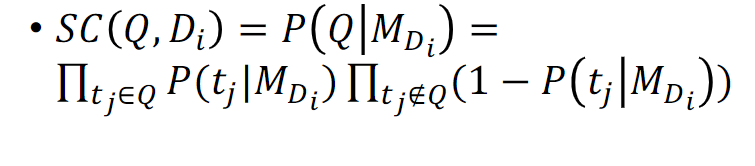


I used the risk value as a mixing parameter to calculate P(Q|MDi).

Before that I calculated P(t|MDi) by the following equation:



Finally I found:



## Imported libraries

## 

Figure

Imported libraries for language model

In figure 1 we can see that I used nltk, re, sklearn, pandas libraries.

## Collection Preparation

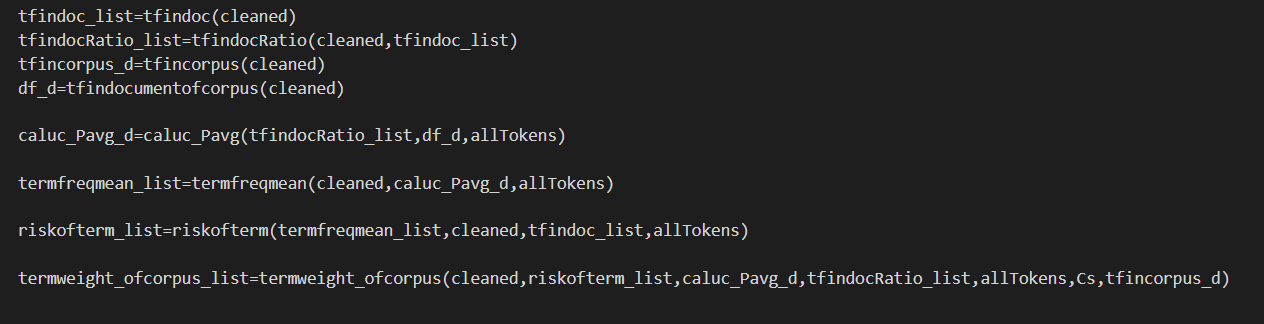
## I done the same opertions that I made to vector space model to the language model.



Figure

Preprocesses to collection for language model

## Language Model operations



Figure

Language model operations

In the figure 7 I called the functions that make the operations of the model languages that I mentioned on 2.

## Processing the Query

## 

Figure

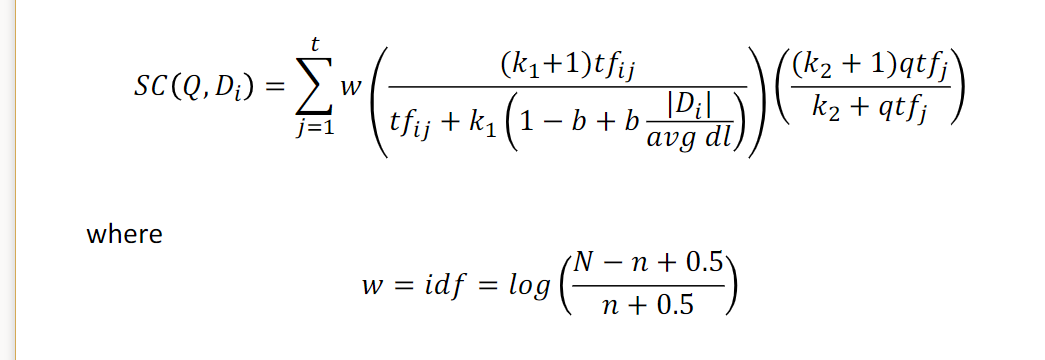
Preprocessing the query

In figure 8 I show the preprocessing operation to the query, then I calculated P(Q|MDi) which will give me the similarity between the query and each document .

I repeated for five queries, to allow me calculating the Average Recalls and Average Precisions.

# Okapi BM25

In information retrieval, Okapi BM25 (BM is an abbreviation of best matching) is a ranking function used by search engines to estimate the relevance of documents to a given search query. It is based on the probabilistic retrieval framework developed in the 1970s and 1980s.



idf is generally used as above.

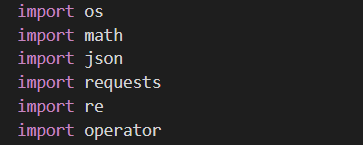
𝑘1, 𝑘2, and 𝑏 are parameters to be empirically determined

k1=1.2; k2: 0 –1000; b=0.75 in many cases.

## Code plan

I Imported the nessecary libraries , then I remove stop words and tokenize them, build a word count dictionary so we can remove words that appear only once,then compute tf (term frequency) per document,and compute df (document frequency) per term,finalilly query our corpus to see which document is more relevant.

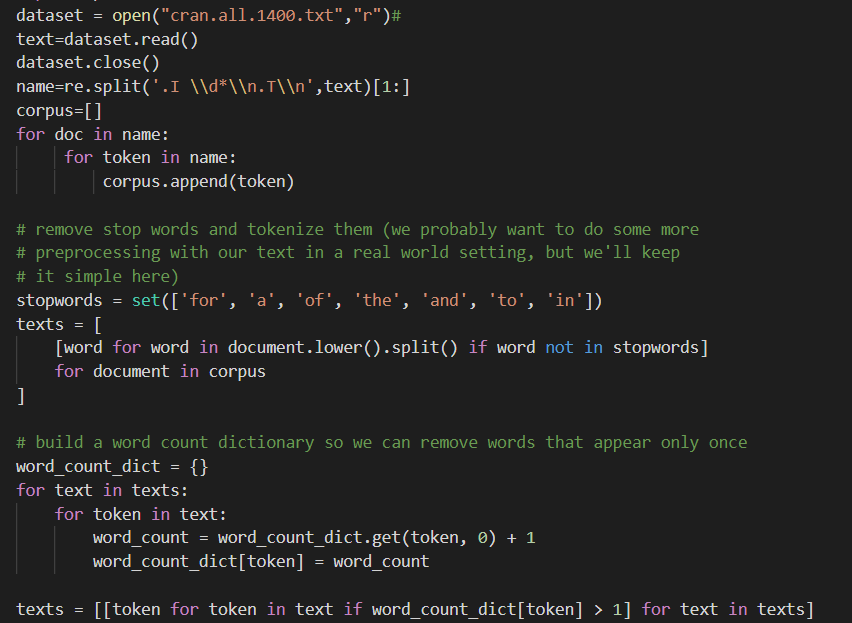
## Imported libraries



Figure

Imported libraries for Okapi BM25

## Collection Preparation

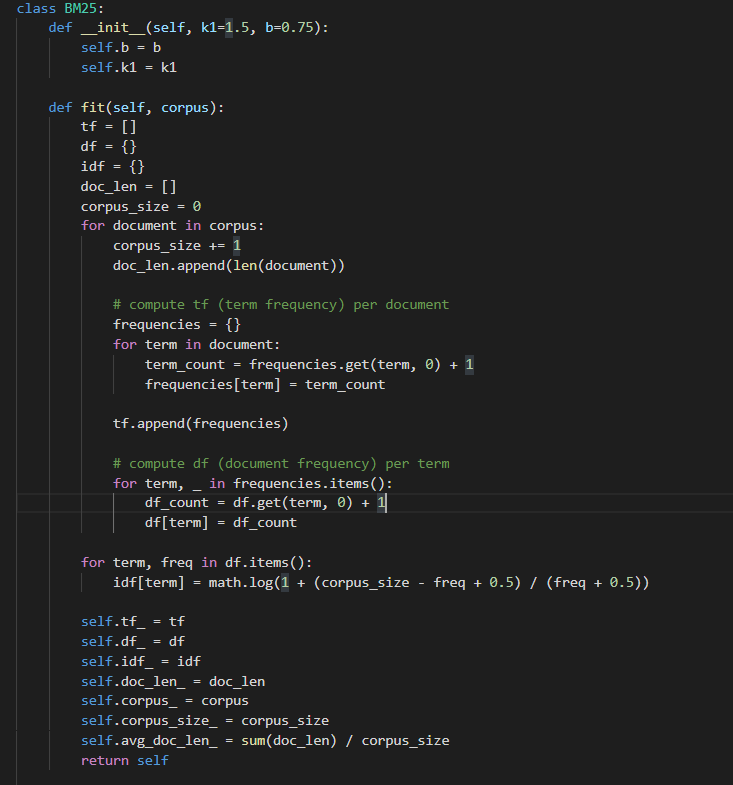


Figure

Collection Preparation for Okapi BM25

In the figure 10 above I made all the preparation for collection, to use it.

## Okapi BM25 operations

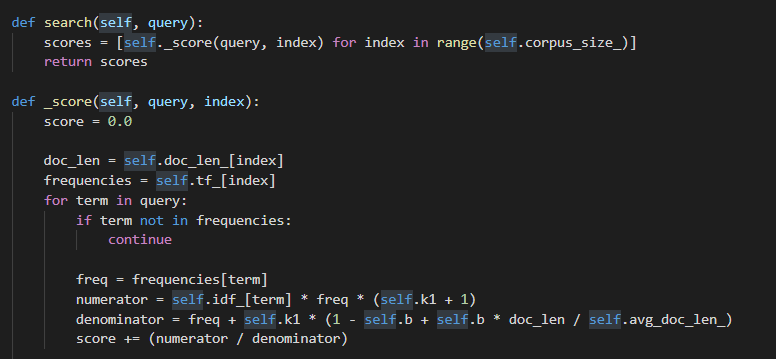


Figure

Class BM25 and fit function of Okapi BM25

In figure 11 we can see that I made class BM25 which have two variables b and k , b=0.75 and k1=1.5.

In the fit function, we call it by sending the collection and it is return it after compute term frequency per each document and compute the document frequency per each term.

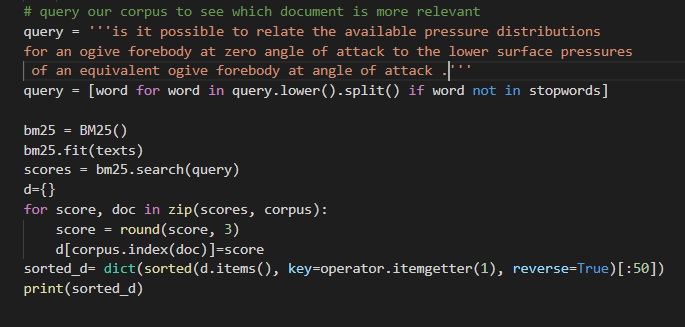


Figure

Search and Score Functions of Okapi BM25

In the above figure I am showing the search function which take the corpus after sending it to fit function and the query which we want to search , and in this function I called \_score function which is shown bottom it ,this function calculate the scores of the documents with the query .

## Implantation Operations



Figure

Implantation Okapi BM25

At the figure 13 I wrote the query then I preprocessed it , then I made an object of Bm25 class to fit it by the fit function then calculate the scores of similarity to print the documents numbers and it scores.

I repeated for five queries, to allow me calculating the Average Recalls and Average Precisions.

# Results

For the query 012 when I processed by the three models I found the following results:

|  |  |  |
| --- | --- | --- |
| Vector Space Model | Language Model | Okapi BM25 |
| 492 | **492** | **492** |
| 124 | **124** | **56** |
| 232 | **57** | **57** |
| 57 | **248** | **124** |
| 413 | **232** | **248** |
| 56 | **56** | **434** |
| 233 | **197** | **122** |
| 248 | **248** | **197** |
| 232 | **443** | **48** |
| 122 | **122** | **443** |
| 973 | **973** | **947** |
| 801 | **541** | **688** |
| 694 | **947** | **694** |
| 636 | **712** | **712** |
| 541 | **688** | **1000** |
| 688 | **638** | **973** |
| 567 | **709** | **638** |
| 695 | **572** | **709** |
| 612 | **815** | **815** |
| 992 | **992** | **572** |
| 1443 | **1443** | **1040** |
| 1301 | **1307** | **1231** |
| 1194 | **1136** | **1307** |
| 1136 | **1347** | **1347** |
| 1041 | **1041** | **1114** |
| 1188 | **1188** | **1350** |
| 1067 | **1067** | **1355** |
| 1195 | **1112** | **1192** |
| 1112 | **1492** | **1104** |
| 1492 | **1104** | **1229** |

Table

Results of query 021

Because I used many different similarity calculations methods I could not get the same results, then I calculate the Averages of Recalls and Averages of Precisions by make a searches of fives queries :

012 : is it possible to relate the available pressure distributions for an ogive forebody at zero angle of attack to the lower surface pressures of an equivalent ogive forebody at angle of attack .

058: how do interference-free longitudinal stability measurements (made using free-flight models) compare with similar measurements made in a low-blockage wind tunnel .

148 :given that an uncontrolled vehicle will tumble as it enters an atmosphere, is it possible to predict when and how it will stop tumbling and its subsequent motion .

157: in summarizing theoretical and experimental work on the behavior of a typical aircraft structure in a noise environment is it possible to develop a design procedure .

**206 :** have any analytical studies been conducted on the time-to-failure mechanism associated with creep collapse for a long circular cylindrical shell which exhibits both primary and secondary creep as well as elastic deformations under various distributed force systems .

|  |  |  |  |
| --- | --- | --- | --- |
|  | Averages of Precisions | Averages of Recalls | Run Time |
| Vector Space Model | **0.1200215** | **0.1021128** | **17 S** |
| Language Model | **0.1700002** | **0.2156001** | **2h 42min 25s** |
| Okapi BM25 | **0.122001** | **0.1231254** | **52 S** |

Table

Results of average of Recalls and Precisions

# Conclusion

At the end of my report , As we see at the results that the language model was realy very complected and took long time to finish, while the shorter one was the vector space model , on other hand I can say that the language model was the best model in avrages of Recalls and Precisions , while the okapi BM25 was not long as language model and give in my expermintal good avrages.

At the end I want to mianition that Evaluation is not straight-forward , acutally I didnot made the search for all quiers to know the exacitly Recalls and precisions so it is not surely results , and of course Still researched today, and The task is paramount to the correct choice of evaluation measure.

# References

[1] Information Retrieval Algorithms and Heuristics Second Edition by David A. Grossman Illinois Institute of Technology, Chicago, IL, U.S.A. and Ophir Frieder Illinois Institute of Technology, Chicago, IL, U.S.A..

[2] <https://chauff.github.io/documents/ir-2011_12/lecture7.pdf> .

[3] <https://en.wikipedia.org/wiki/Okapi_BM25> .

[4] <http://ir.dcs.gla.ac.uk/resources/test_collections/cran/> .