Information Retrieval

Coursework 3 - Search Engine Implementation

Name: Kweku E. Acquaye
Group: 68 (single-member group)

This report is a summary accompanying the Jupyter notebook submitted together with it, and describes briefly the theory as well as the practical work in the notebook. Also submitted is a video presentation demonstrating how the model was built, its ability to receive a query, and then yield an output in the form of a ranked list of documents.

Declaration
Some of the images used in this report have been reproduced from Semantic Scholar, GitHub, YouTube, Towards Data Science, Modern Information Retrieval (Baeza-Yates and Ribeiro-Neto, 1999), and IR Lecture slides and Lab notes.

Designing a Search Engine for Covid-19 Publications

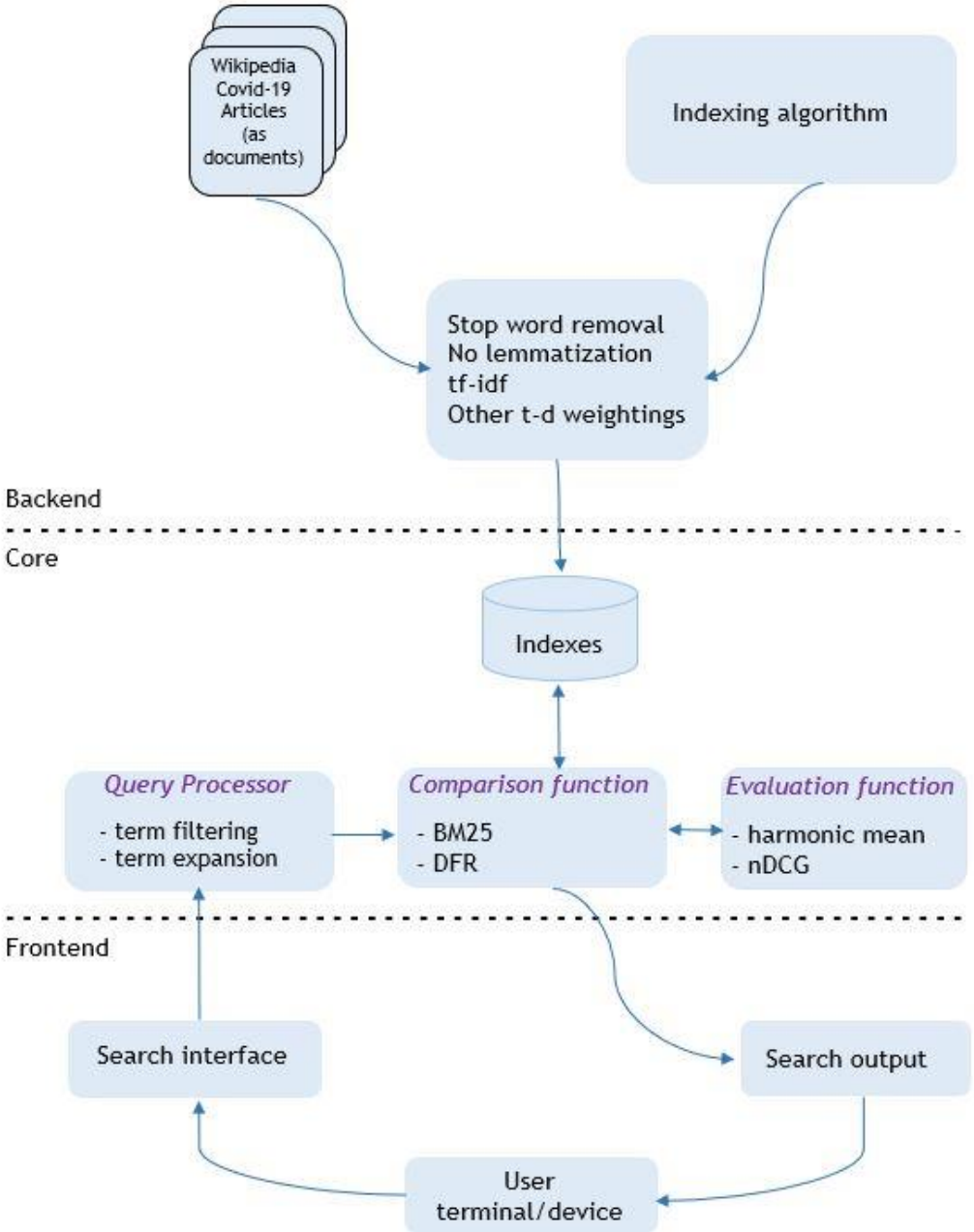
The Covid-19 (SARS-CoV-2 virus) pandemic has waned in western countries. At its peak it highlighted how, in the modern digital age, readily available information from multiple sources can lead to entrenched, dichotomous and multipolar views held with conviction, and thus the importance of information retrieval (IR) systems that provide *reliable* information.

As an exercise in understanding and appreciating the technical details of how search engines work, this miniproject builds a fully functional model search engine (small scale), and evaluates its performance in terms of the validity (*relevance*) and also “trueness” (*precision*) of the information it returns to simple and complex queries.

Responsibilities

Being a single-member group (Group 68), I am responsible for developing the entire search engine.

Schematic Representation of Search Engine Architecture



Dataset

This design architecture is now implemented in the accompanying Jupyter notebook, with two minor difference - the dataset has had to be changed – program execution time of early trials with the 30-gigabyte CORD-19 dataset (Wang *et al* 2020), even in the simplest of operations, proved too long to be of use as a ***model*** search engine. It has therefore been substituted with a fairly and appropriately wide dataset of Wikipedia articles on Covid-19. Instead of local or cloud storage of the dataset, data is accessed and parsed directly by utilising Wikipedia API for Python version 1.4 (online reference 1).

Also, instead of determining the Mean Average Precision (MAP) as the 2nd method of evaluation, the normalized Discounted Cumulative Gain (nDCG) method is used.

Reason

Wikipedia articles tend to range in objectivity and reliability, and are therefore a good choice for this exercise.

Index

- standard pandas, NumPy and scikit-learn libraries
- stop word removal
- no lemmatization (to improve *context*)
- t-d weights
- tf
- idf
- tf-idf,

Comparison/Retrieval Function

- Okapi Best Match 25 (BM25) model
- technical language in data
- length norm
- tuning constants

Parameters

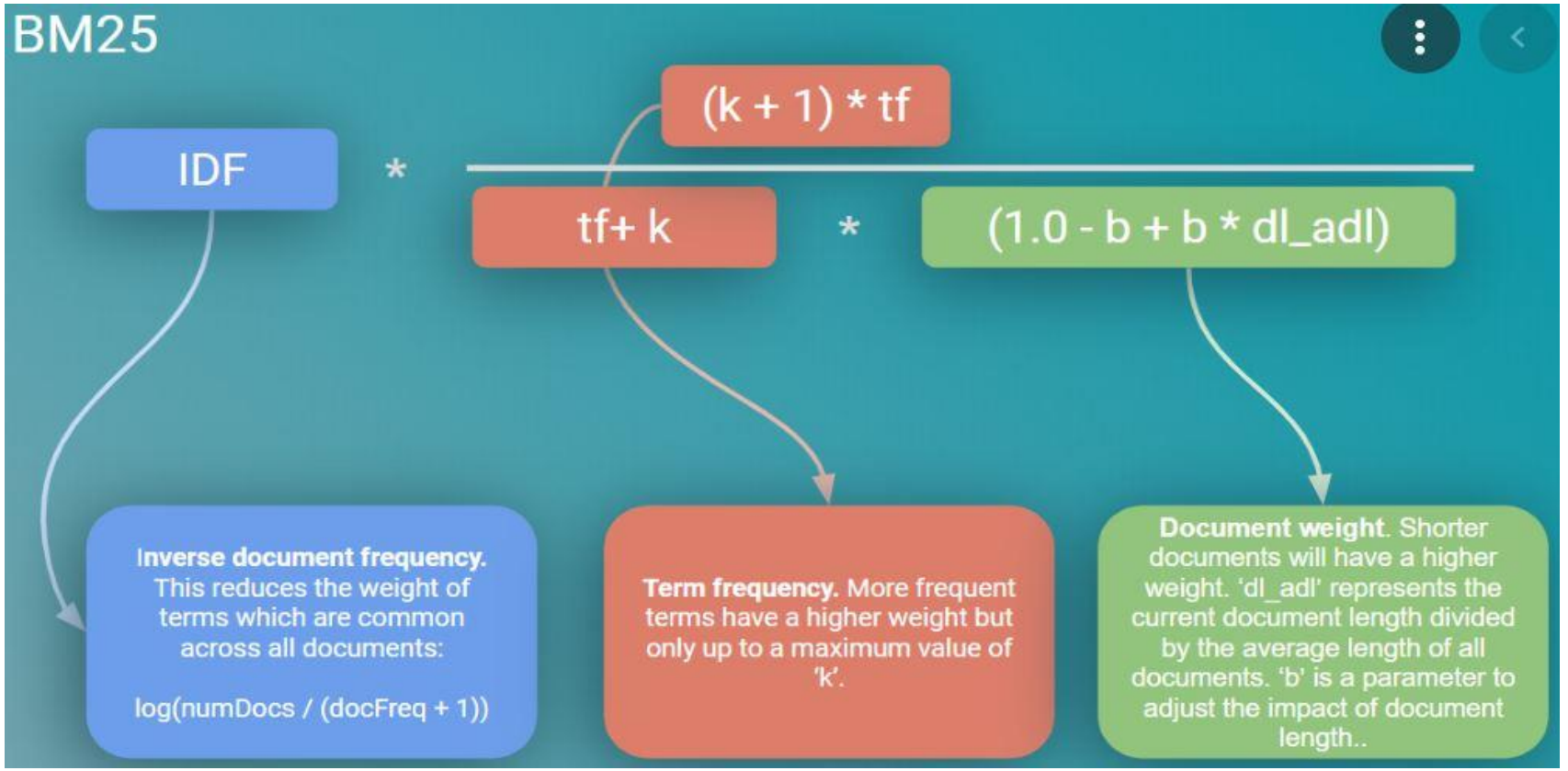


image source: towards data science

BM25: an intuitive view

Common words less important

Repetitions of query words ➔ good

$$\log \frac{P(D \mid R=1)}{P(D \mid R=0)} \approx \sum_{w} \left(\frac{d_w (1+k)}{d_w + k\left((1-b) + \frac{b \cdot dl}{avg.dl}\right)} \cdot \log \frac{N - N_w + \frac{1}{2}}{N_w + \frac{1}{2}} \right)$$

More words in common with the query ➔ good

Repetitions less important than different query words

But more important if document is relatively long (wrt. average)

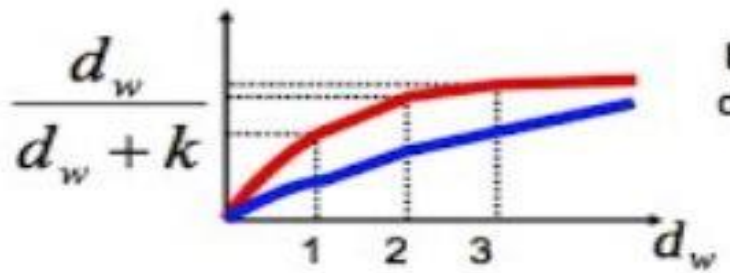


image source: YouTube

IDF comparison

BM25

$$w^{IDF}(t) = \log\left(\frac{N - df_t + 0.5}{df_t + 0.5} + 1\right)$$

TF/IDF

$$w^{IDF}(t) = \log\left(\frac{N + 1}{df_t + 1} + 1\right)$$

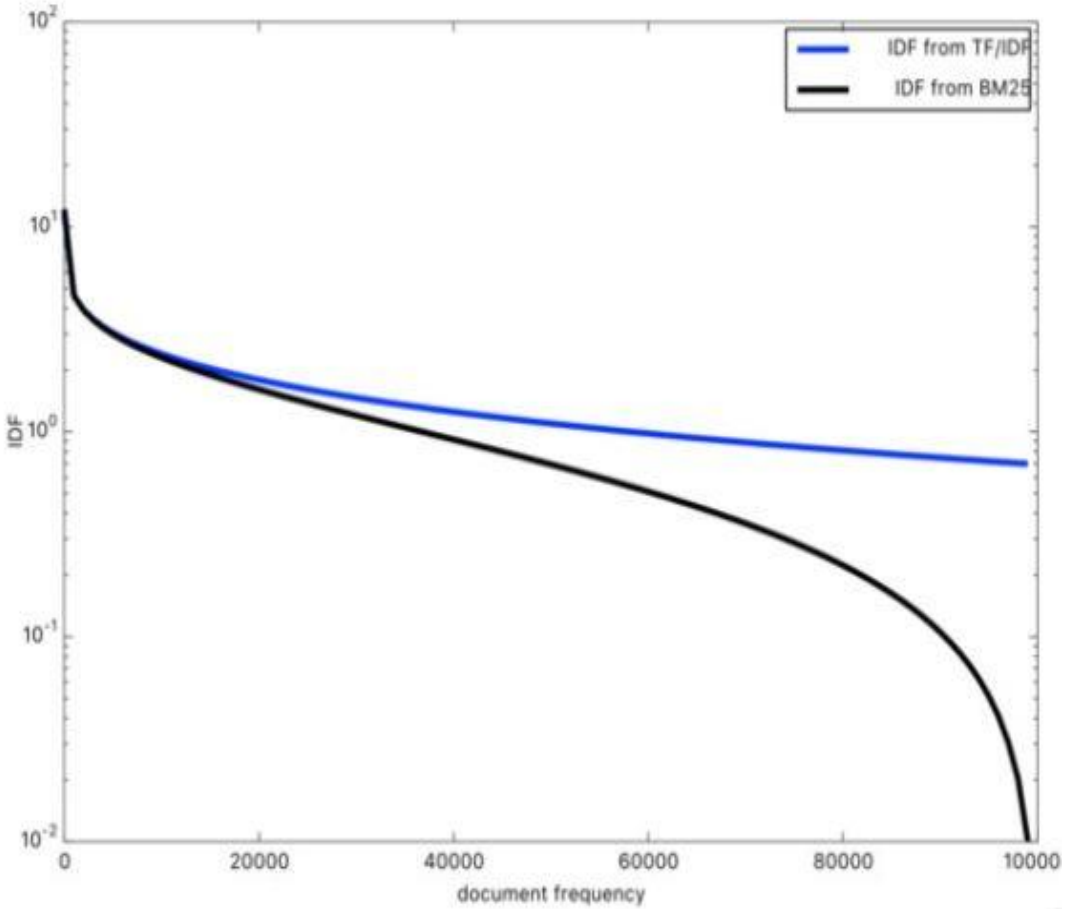


image source: semantic scholar

Schematic Representation of Retrieval Function

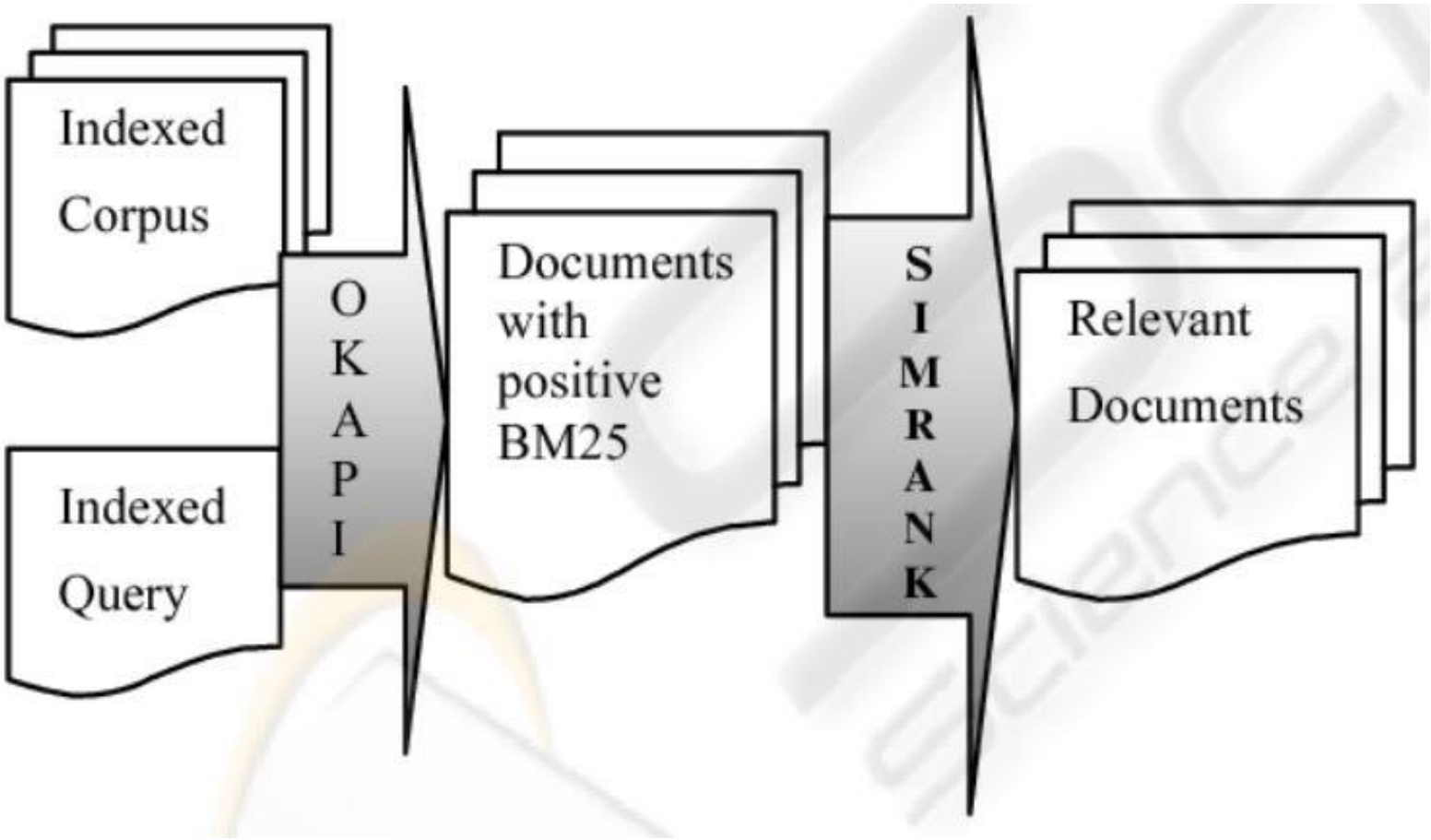


image source: semantic scholar

Evaluation Functions

Evaluation of the effectiveness of the search engine, i.e. the quality of its retrieval.

- **Harmonic Mean (F1-score)**
  - ✓ standard pandas, NumPy and scikit-learn libraries
  - ✓ precision-recall plane
  - ✓ manual relevance judgements of retrieved documents
- **Normalized Discounted Cumulative Gain (nDCG)**
  - ✓ *ndcg_score* module from the *metrics* library of scikit-learn utilised
  - ✓ y_true from qrels and y_score from document rankings
  - ✓ ndcg scores

Harmonic Mean (F1-score)

- Defined as the reciprocal of the mean of the reciprocals of a set of numbers.
- Intuitively the reciprocal of the arithmetic mean, obtained by calculating the reciprocal of the mean of individual observations.
- Accuracy measure combining precision and recall measures.

$$H = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \cdots + \frac{1}{x_n}} = \frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}} = \left(\frac{\sum_{i=1}^{n} x_i^{\,1}}{n} \right)^{-1}$$

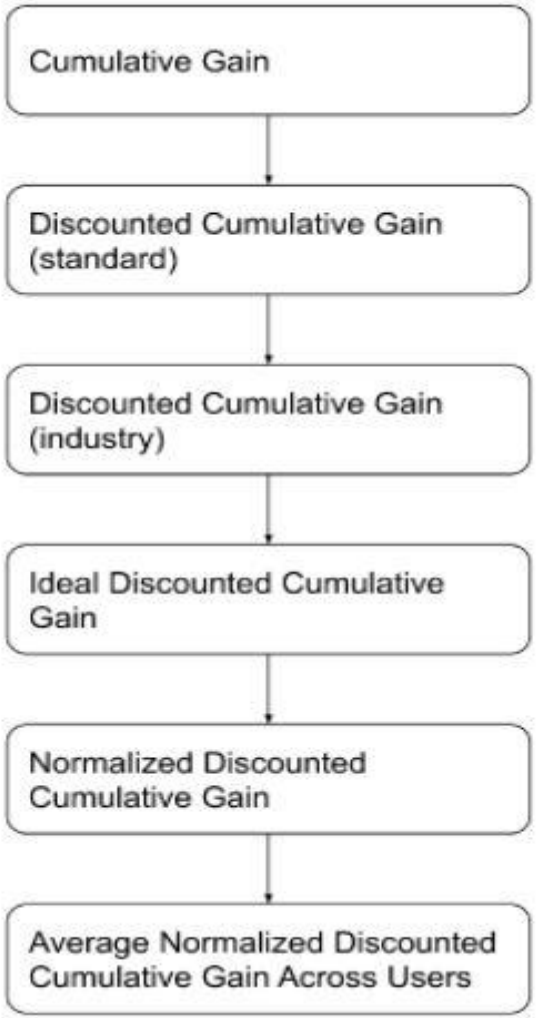
image source: machine learning mind

Normalized Discounted Cumulative Gain (nDCG)

- Uses *graded relevance* as a measure of usefulness, or *gain,* from examining a document
- Gain is accumulated starting at the top of the ranking and may be reduced, or *discounted*, at lower ranks
- Typical discount is 1/log *(rank)*
  - With base 2, the discount at rank 4 is 1/2, and at rank 8 it is 1/3

source: IR Lecture slides

Normalized Discounted Cumulative Gain (nDCG)



$$\mathrm{CG_p} = \sum_{i=1}^{p} rel_i$$

p elements in the recommended ranking

graded relevance of the result at position i (gain)

$$\mathrm{DCG_p} = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)}$$

logarithmic reduction factor to penalize proportionally to the position of the result

$$\mathrm{DCG_p} = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

graded relevance to get a stronger emphasis on retrieving relevant documents

$$\mathrm{IDCG_p} = \sum_{i=1}^{|REL_p|} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

list of relevant documents (ordered by their relevance) in the corpus up to position p

$$\mathrm{nDCG_p} = \frac{DCG_p}{IDCG_p}$$

source: GitHub

Evaluation Results

The Harmonic Mean method yielded F1 scores of 0.333 and 0.444 with precision @5 of 1.0 and 0.4 respectively for the 2-term queries "vaccine effectiveness" and "cell membrane".

The Normalized Discounted Cumulative Gain method gave ndcg scores of 0.170 and 0.441 with precision @5 respectively for the same 2-term queries, i.e. "vaccine effectiveness" and "cell membrane".

The reason for the apparent doubling in score for query term “vaccine effectiveness” while remaining identical for query term “cell membrane” in the Harmonic Mean method compared to the nDCG, is still being contemplated.

Overall, this exercise is deemed a success in terms of the purpose for which it is undertaken.

References

1. Baeza-Yates, R. and Ribeiro-Neto, B., 1999. *Modern information retrieval* (Vol. 463). New York: ACM press.

2. Online Reference 1, Wikipedia API for Python, [online]: https://pypi.org/project/wikipedia/