NLP Project Report

Submitted by : MUSTAFA (EP20B025)

Introduction

The goal of this project is to address the limitations of the basic Vector Space model based on TF-IDF term weighting measures. We try to improve on the following aspects – sensitivity of the VSM framework to different weighting schemes, it's ability to handle polysemy and synonymy and ways to incorporate some semantic context into the model. A detailed account of these approaches with appropriate hypothesis testing methods is listed below.

Limitation 1 - Sensitivity to weighting schemes

In information retrieval, tf–idf (short for term frequency–inverse document frequency), is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus [1]. It is often used as a weighting factor in determining the relevance ranking of documents wrt a given query.

$$w_i = tf_i * log\left(\frac{D}{df_i}\right)$$

Figure 1: TF-IDF formula

where tfi = term frequency (term counts) or number of times a term i occurs in a document, this accounts for local information, dfi = document frequency or number of documents containing term i, D = total number of documents in a database.

However, it is susceptible to the following:

1. Simply boosting the number of occurrences of a certain term in document increasing it's term frequency and leads to the unfortunate keyword spamming problem where useful search terms are drowned out by the skewed weighting towards spammed keywords.
2. It ignores the length of the document which can be an important factor in determining it's relevance. A rare term occurring frequently in a short document signals the document's specificity to that topic, whereas a single occurrence of the same term in a long document could be a passing mention. We would like to retrieve the former document and incorporating document is the way to do so.

The improved BM25 measure, first used in the Okapi IR system is as follows:

$$\text{score}(D,Q) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i,D) \cdot (k_1+1)}{f(q_i,D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)}$$

Figure 2: BM25 formula

where f(qi, D) is the number of times term qi appears in document D, —D— is the length of document D in words, avgdl is the average document length, k1 and b are hyper-parameters.

It saturates the effect of high term frequency with the normalizing denominator and incorporates the deviance of document length over the average length to address both shortcomings of the TFDIF model. This is evidenced by the graphs below.

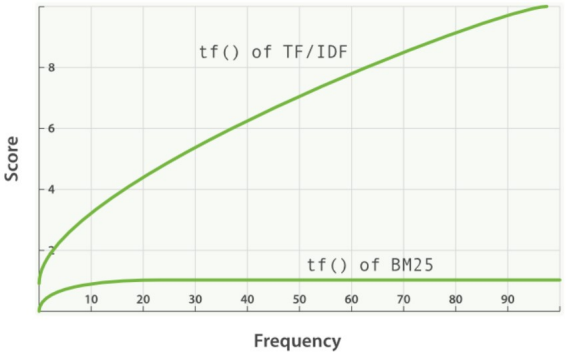

Figure 3: Saturation of TF

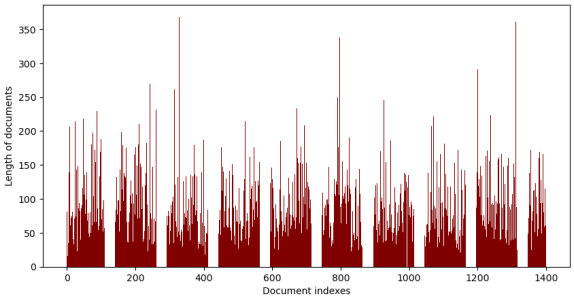

Figure 4: Cranfield doc lengths barplot

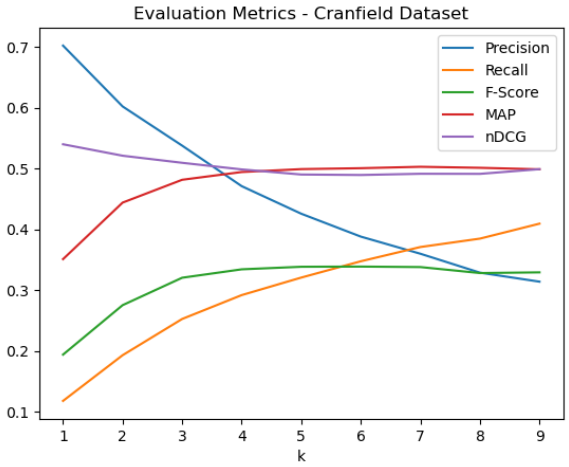

Figure 5: BM25 metrics

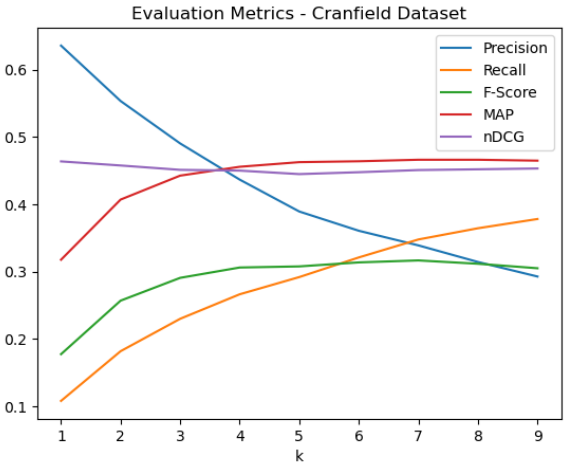

Figure 6: TFIDF metrics

The erratic nature of the Fig.4 document length graph suggests the BM25 measure might be a good fit to the Cranfield dataset and this hypothesis is validated by the evaluation measures graph 5 and 6. It shows better performance over the base VSM model in evaluation metrics. It fails to retrieve appropriate documents for 16 queries, whereas the base VSM has a higher failure rate of 20 documents. This shows superior performance.

```
------------------------------------------------
Number of queries with 0 precision -  16
Query IDs -  [8, 21, 27, 34, 43, 61, 62, 63, 108, 109, 115, 150, 151, 206, 215, 218]
------------------------------------------------
```

Figure 7: BM25 analysis

```
------------------------------------------------
Number of queries with 0 precision -  20
Query IDs -  [4, 8, 18, 21, 27, 34, 43, 61, 62, 71, 73, 84, 109, 114, 115, 150, 151, 166, 206, 215]
------------------------------------------------
```

Figure 8: TFIDF analysis

An analysis of the zero precision queries: Query 28 inquires about 'curved wings' and it's relevant document 512 contains similar words like 'cylindrical surface', 'half-ring wings'. Query 35 inquires about accoustic wave propagation in chemically reactive gases and it's relevant documents 166 and 517 contain similar words such as 'dissociating gases', 'chemical reactive mixtures' and 'shock wave'. This common error shows that both weighting schemes cannot capture word similarity as they are simple boolean matching measures.

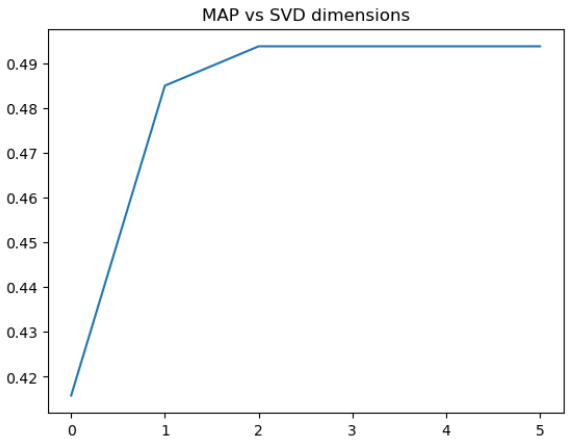

Figure 9: MAP vs number of SVD dimensions

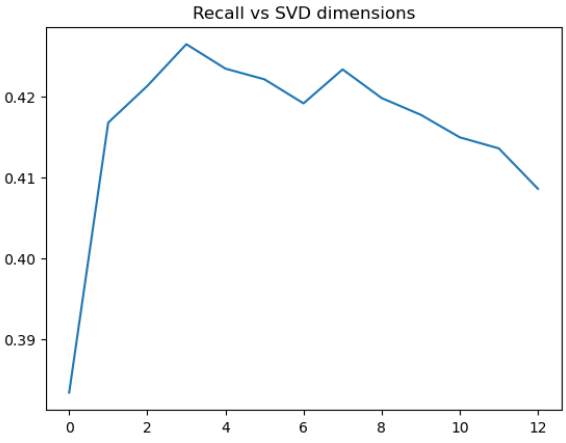

Figure 10: Mean recall vs number of SVD dimensions

Limitation 2 - Latent Concept Space

The previous approach boosted retrieval performance but the failure analysis indicates that the model struggles with synonymy cases – where similar context is described with different words in the query and document. The technique used to resolve this issue is Latent Semantic Analysis. According to [2], LSA assumes that words that are close in meaning will occur in similar pieces of text (the distributional hypothesis). A matrix containing word counts per document (rows represent unique words and columns represent each document) is constructed from a large piece of text and a mathematical technique called singular value decomposition (SVD) is used to reduce the number of rows while preserving the similarity structure among columns. Documents are then compared by cosine similarity between any two columns.

Our implementation of LSA is inspired by Deerwester's Indexing by LSA paper and we utilize the sparse SVD algorithm provided in the scipy.sparse.linalg library to accelerate indexing with sparse term-document matrices.

LSA handles synonymy well by condensing similar senses of distinct words into one underlying vector in the latent concept space. The number of latent dimensions used for the SVD is a useful toggle to control to the precision and recall of the model. A low rank approximation is useful for broad search across concepts and is intended to improve recall. Conversely, a high rank approximation implements a granular search to improve precision. These trends are reflected in the following graphs, with precision increasing with k, and recall decreasing with k. Based on these trends and manual experimentation, k=700 dimensions was chosen to be the optimum number to balance metrics and runtime.

The LSA method performs better than the base VSM with better MAP, nDCG scores and improves to 15 underperforming query matches against the VSM's 20.

```
------------------------------------------------
Number of queries with 0 precision -  15
Query IDs -  [21, 27, 34, 43, 58, 62, 73, 84, 86, 108, 109, 116, 166, 204, 215]
------------------------------------------------
```

Figure 11: LSA query analysis

However, the downside of this approach is that it is difficult to interpret the latent concept space in clear linguistic terms. We looked to word embeddings to encode more high-level semantic context into the IR system.

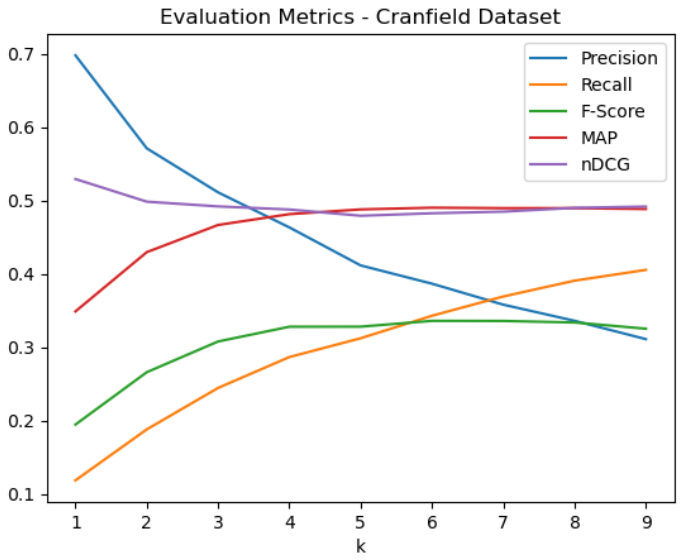

Figure 12: LSA metrics

Limitation 3 - Word embeddings

The word2vec algorithm uses a neural network model to learn word associations from a large corpus of text. Once trained, such a model can detect synonymous words or suggest additional words for a partial sentence. As the name implies, word2vec represents each distinct word with a particular list of numbers called a vector. The vectors are chosen carefully such that they capture the semantic and syntactic qualities of words; as such, a simple mathematical function (cosine similarity) can indicate the level of semantic similarity between the words represented by those vectors. [3]

Upon fitting the gensim word2vec model on our dataset and converting it to a compact KeyedVector instance, cosine similarity was used to calculate the relevance of the document to a particular query. The document was represented as sum of the vectors of it's constituent words.

However, this representation did not yield good results despite expectations. MAP and NDCG scores of this approach are lower than the base VSM model's metrics, suggesting that encoding semantic similarity might be less useful than pure term weighting approaches in Information Retrieval.

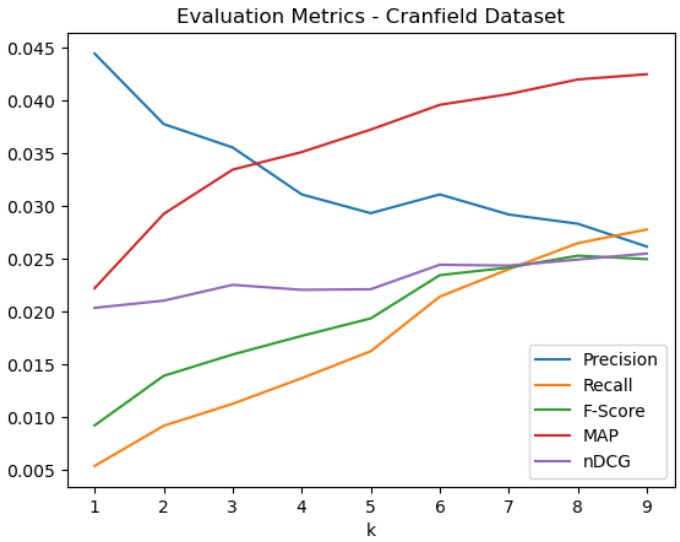

Figure 13: Word2Vec metrics

Sidenote - A bigram approach to capture ultra specific word combinations was tried but it provided disappointing results and had an infeasible runtime.

Figure 14: Zero precision query matches with word2vec

Conclusion

Appropriate term weighting schemes like BM25 which incorporate as much feature information about the term frequencies perform best in tandem with similarity techniques like latent semantic analysis.

Future ideas - An ensemble technique to combine various similarity measures like LSA, ESA along with statistical and boolean methods has been proved to be useful according to [2]. This is could be implemented in the next extensive project.

References

[1] - tf–idf. (2023, March 6). In Wikipedia. https://en.wikipedia.org/wiki/Tf
[2] - Latent semantic analysis. (2023, February 22). In Wikipedia. https://en.wikipedia.org/wiki/Latent-semantic-analysis
[3] - Word2vec. (2023, May 9). In Wikipedia. https://en.wikipedia.org/wiki/Word2vec
[4] - Saturation figure - https://www.elastic.co/blog/practical-bm25-part-2-the-bm25-algorithm-and-its-variables