CS6370 (NLP): PROJECT

Manikandan Sritharan (EE19B038) & Snehan J (EE19B027)

1 Introduction

Information Retrieval is an essential task with applications in various search engines. It is a class of problems where the goal is to obtain relevant documents from a database with the input as a query. It is the science of finding the most relevant set of k documents. One classic example of Information Retrieval is Google Search. Multiple web pages are indexed by the web crawlers. Based upon a given query, the search engine tries to retrieve the most relevant set of webpages (or documents) and ranks them. Although web search engines are the most popular IR applications, it has a wide variety of applications.

There are several methods to approach the information retrieval problem. One of the most common methods for IR in structured databases is SQL-based queries. However, the queries used in IR are usually given in a natural language, and therefore, SQL is not compatible with the task. Most of the existing IR-based systems compute a numeric score for every document in the database with the given query. This makes the ranking process more straightforward, and the user can then choose to retrieve the corresponding document/documents based on the shown results.

2 Problem Definition

We are given the Cranfield dataset. It is one of the standard classical datasets for the purpose of information retrieval. Cranfield dataset contains 1400 documents and 225 queries. Associated with each query is a list of relevant documents with their corresponding relevance score. The relevance score denotes whether a document answers the query, whether it is just an example or if it only refers to a few things relevant to the document.

Our task is to build a complete end-to-end IR system. This IR system has been documented in its database. It takes queries as input and reports k relevant documents as output.

3 Motivation

The current TF-IDF model assumes orthogonality between all the words. This assumption is not valid. Moreover, the vectors are very sparse and very large. The large size of the vector also amounts to the considerable time taken to compute the cosine similarity.

The other shortcomings of the vector space model are:

- Each time we add a new term into the term space we need to recalculate all vectors.
- The order in which the terms appear in the document is lost in the vector space representation.
- False negative matches and failure to capture synonymy and polysemy.

Transformers have played a significant role in advancing the field of NLP in recent years. Before the development of transformers, most NLP models were based on recurrent neural networks (RNNs) and convolutional neural networks (CNNs). However, RNNs and CNNs have limitations in capturing long-term dependencies and preserving contextual information. Transformers are based on the attention mechanism, which allows the model to selectively focus on different parts of the input sequence when generating the output.

4 Proposed Methodology

4.1 Baseline Model

The baseline model consists of

- Sentence Segmentation
- Word Tokenization
- Stopword removal
- Lemmatization
- Information Retrieval
- Evaluation

Most of the methods that we are using in this study follow the same preprocessing steps as the baseline model. The preprocessing steps include Sentence Segmentation, Word Tokenization, Stopword Removal and Lemmatization. These methods are also evaluated on the same metrics as mentioned above.

4.2 Latent Semantic Analysis

Latent Semantic Analysis (LSA) is used for analyzing relationships between a set of documents and the terms that are present in them. This is done by producing a set of concepts related to the documents and terms. LSA works on the assumption that the words that are close in meaning will occur in similar pieces of text.

For LSA, we generated the TF-IDF vectors corresponding documents and build a TF-IDF x document matrix. Then, we performed Singular value decomposition to obtain the concepts. The resulting vectors of the documents are linear combinations of individual concepts. This overcomes the assumption that is made in the TF-IDF vector space model that the terms are orthogonal to each other. Further, the query vectors also undergo a similar transformation, and cosine similarity is used to obtain the rankings of the documents.

4.3 Word2Vec Embeddings (Google News)

Word2Vec is a shallow neural network model to learn word associations. Word2Vec is tailored in such a way that cosine similarity between the words would represent their semantic similarity. It assumes that distributionally similar words would also be semantically identical.

Word2Vec is done using the gensim model. This model is trained on the google news dataset corpus. The Cranfield data contains very technical terms which are not present in the training corpus used for training Word2Vec. As a result, we skip the words which are not present in Word2Vec vocabulary when creating the sentence representation. In-order to overcome this issue we tried to train a Word2Vec model on the Cranfield dataset. But Word2Vec algorithm is quite data hungry and the results obtained with Word2Vec trained on Cranfield dataset are much lower compared to LSA method.

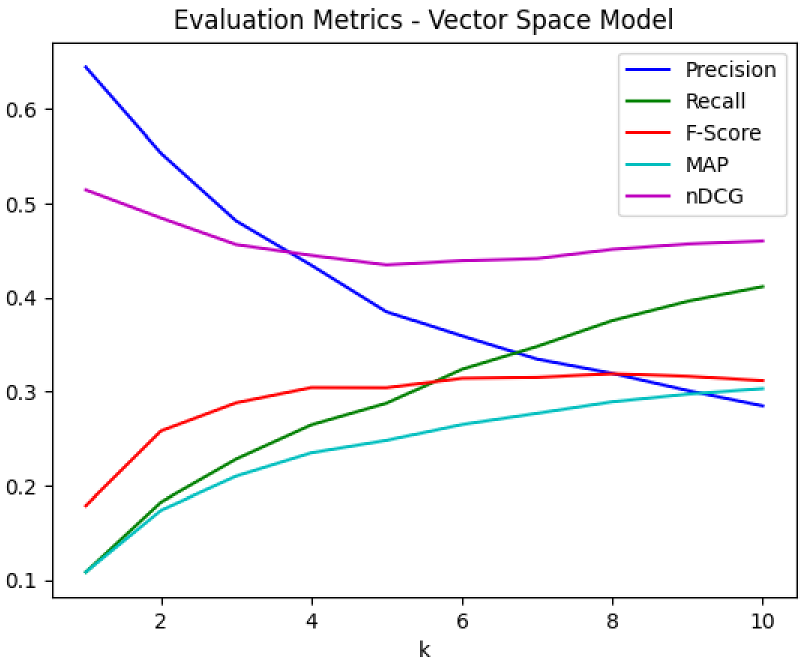
4.4 Bert Transformer

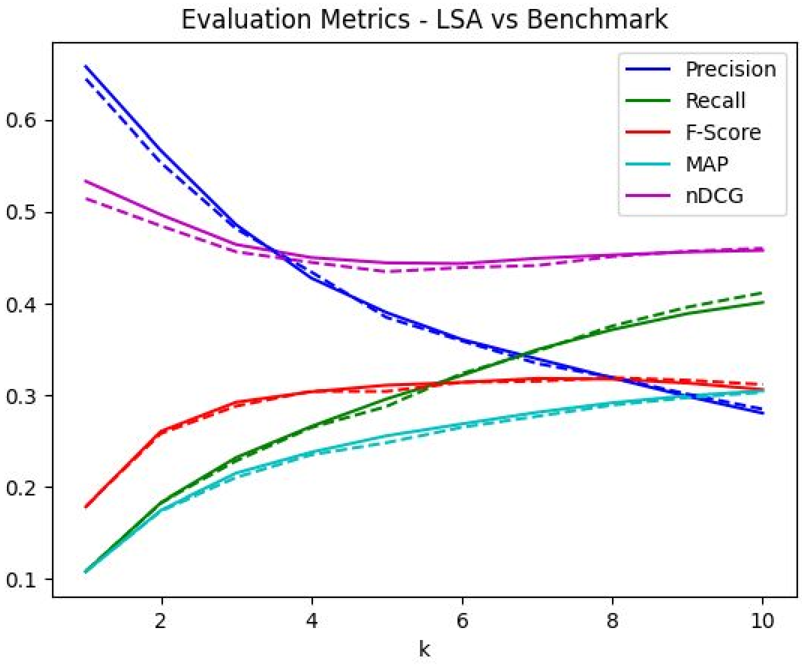
BERT is based on the Transformer architecture and is trained on a large amount of text data to learn representations of words and sentences in a way that captures their semantic meaning. A sentence transformer is a type of model that takes in a sentence as input and produces a vector representation of that sentence. Combining the power of BERT and sentence transformers, BERT sentence transformers are pre-trained models that can generate high-quality sentence embeddings.

It maps sentences and paragraphs to a 384 dimensional dense vector space and can be used for tasks like clustering or semantic search.

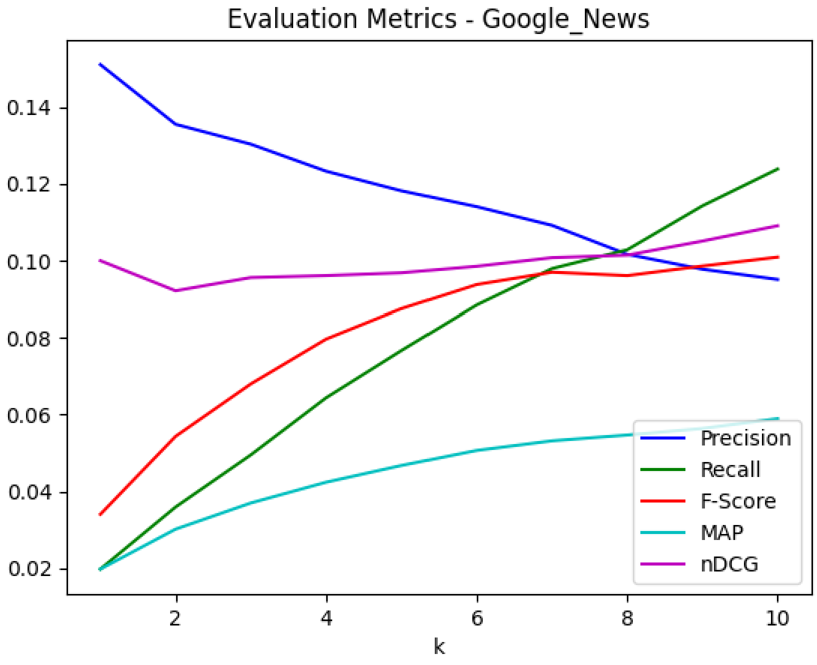
5 Results

5.1 Information Retrieval Performance

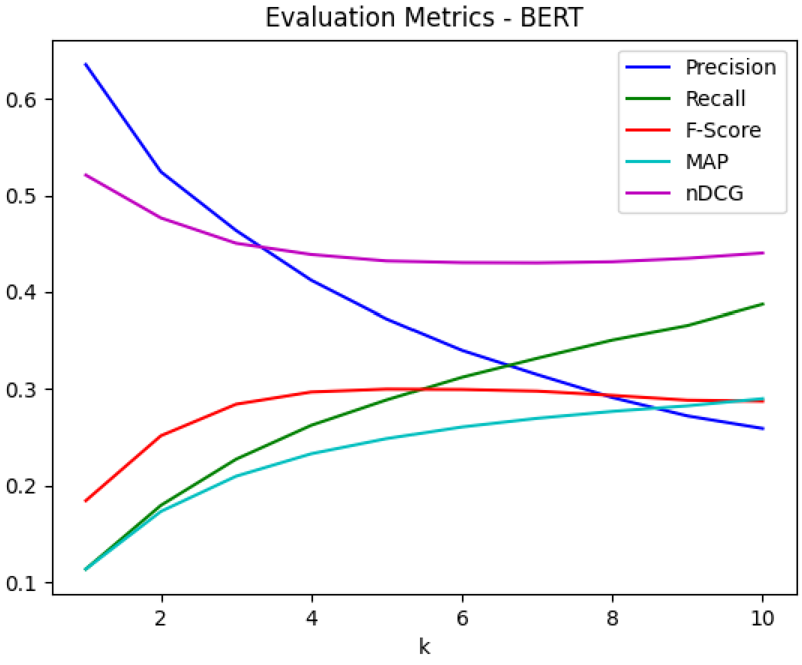




It is seen that Latent Semantic Analysis model performs marginally better when compared to the baseline model. Also, the LSA model was trained on a modified corpus, taking into account, the body and the title of the Cranfield dataset. There is improvement in all the performance metrics that we are considering as compared to the baseline model.



The Word2Vec embedding model seems to perform significantly worse than the baseline model. This suggests that maybe if the model was trained on a larger dataset than Cranfield but having similar properties as Cranfield documents and queries, it can potentially beat the baseline model.



Even the BERT transformer model seems to perform slightly worse than the baseline model. This could be the quality of data that the transformer was trained on when compared to the test data (Cranfield dataset). Thus, the performance can also be affected by the training parameters used. Also, finetuning the BERT model could lead to significant outperformance.

6 Conclusion

The performance scores that we are getting are consistent with Aguilar et al. [4] as they also had LSA as the best performing model.

We first tried baseline, because it seemed too simplistic we tried using LSA. Later we tried using word embeddings and BERT sentence transformer to capture richer semantic relatedness. The better performance of simpler models such as LSA and the baseline model can be attributed to the fact that their corresponding parameters are estimated much better than the other complex models with multiple parameters since their models are simpler. These other complex models probably require much larger amounts of data for better parameter estimation. In other words, the other complex models were underfitted, while LSA and baseline models had an advantage over them as the number of documents in the Cranfield dataset is low.

7 Acknowledgement

We are grateful to Prof Sutanu Chakraborti for providing us with an opportunity to work on this project and equip us with the tools necessary to build a complete Information Retrieval System. We are also grateful to the TAs of the NLP course for helping us with our difficulties.

8 References

1. https://huggingface.co/sentence-transformers/paraphrase-MiniLM-L6-v2

2. https://jonathan-hui.medium.com/machine-learning-singular-value-decomposition-svd-principal-component-analysis-pca-1d45e885e491

3. https://huggingface.co/fse/word2vec-google-news-300

4. https://repository.eafit.edu.co/handle/10784/28641