Exploring Information Retrieval on technical dataset with traditional NLP and modern Deep Learning methods

Akiti Gunavardhan Reddy^{CH18B035} and Narendhiran.R^{CH18B015}

Indian Institute of Technology,Madras
{ch18b015,ch18b035}@smail.iitm.ac.in

Abstract. The main objective of this work is to improve the recommendation list of relevant documents from the Cranfield Dataset when given a query. This work describes our two approaches and corresponding results for the information retrieval task on technical dataset. Taking classical Vector Space Model through computing Tf-Idf scoring approach as a benchmark for comparison. Our first approach is implementing Latent Semantic Analysis model and to evaluate metrics and correlation coefficients on test data. The second approach leverages the modern methods of Transformer models(Deep learning architectures) and their capability to learn contextual representations of the query and corpus in order to perform semantic search. The proposed evaluation techniques are P-R curves and ndcg score plots.

Keywords: LSA · Transformer · DeepLearning.

1 Introduction

Natural language processing is a key technology for building the information systems of the future. Unlike today's relatively crude search engines that retrieve long lists of documents of often questionable relevance, the future systems will deliver the exact information that the user is seeking, and will do so with the highest precision and reliability. To accomplish this will require the systems to "understand" both the user's information need, as well as the information they possess in their databases.[1]

Methods of capturing semantic relatedness in words/documents have been explored to a significant extent by the NLP community. Major approaches include data-driven learning of vector representations from a very large corpora. While such bottom-up approaches have proven to be effective in recent times, the outcomes of such models can be further amplified by using some form of top-down knowledge gained over time, by humans. Such information has been crystallized into structured databases since as early as 1995. These top-down resources can be utilized to guide bottom-up expeditions to identify relations among words/documents.

During the past decades, there has been a trend moving from traditional approaches to IR toward deep learning approaches to IR.The capability of neural ranking models to extract features directly from raw text inputs overcomes many limitations of traditional IR models that rely on handcrafted features. Moreover, the deep learning methods manage to capture complicated matching patterns for document ranking.

Latent Semantic Analysis approach is a popular technique that is designed to overcome a fundamental problem that plagues existing retrieval techniques that try to match words of queries with words of documents. The problem is that users want to retrieve on the basis of conceptual content, and individual words provide unreliable evidence about the conceptual topic or meaning of a document. There are usually many ways to express a given concept, so the literal terms in a user's query may not match those of a relevant document. In addition, most words have multiple meanings, so terms in a user's query will literally match terms in documents that are not of interest to the user. This proposed approach tries to overcome the deficiencies of term-matching retrieval by treating the unreliability of observed term-document association data as a statistical problem. We assume there is some underlying latent semantic structure in the data that is partially obscured by the randomness of word choice with respect to retrieval. We use statistical techniques to estimate this latent structure, and get rid of the obscuring "noise." A description of terms and documents based on the latent semantic structure is used for indexing and retrieval.' [2]

Using Deep Learning we would represent or create vector representation of queries and documents, and these vector representations can be compared to observe similarity. These vector representations are known as sentence embeddings. These embeddings can be customly trained or can be used from pretrained embeddings generated from large corpora. In addition our work also comprises of implementing the paper - **BERT meets Cranfield:Uncovering the Properties of Full Ranking on Fully Labeled Data.** This paper trains custom BERT embeddings for documents specific to the Cranfield Dataset. We experiment over several hyperparameters of model on a number of evaluation benchmarks in cranfield data set and cross-validating the results cited in the paper.

2 Problem Definition

The goal of an information retrieval system is to maximize the number of relevant documents returned for each query. Keyword information retrieval systems often return a proportion of irrelevant documents because matching keywords is imprecise: words can have different meanings when used in different contexts, and a single idea can often be expressed by several different words or synonyms.

Information retrieval systems can be made more precise by matching concepts, keywords for which the intended meaning has been identified, either with information from a lexicographic database in the case of documents, or by asking the user to choose one meaning from several possible meanings in the case of queries[3].

Hence we plan to improve and analyse our search engine application on technical dataset considering the factors mentioned above and compare it with classical vector space model.

3 Motivation

A fundamental deficiency of classical vector space model information retrieval methods is that the words searchers use often are not the same as those by which the information they seek has been indexed. There are broadly two sides to this issue namely synonym and polysemy. Synonymy refers to: "...words that are pronounced and spelled differently but contain the same meaning." and polysemy refers to: "...words that are pronounced and spelled same but contain the different meaning." [2]

These are the main factors which result in the poor precision and recall. Core problem of such approachs is disregarding the relatedness between the words. But can be taken advantage of if considered. Which is exploited in our proposed approach, where we try to overcome the deficiencies of term-matching retrieval by treating the unreliability of observed term-document association data as a statistical problem. We assume there is some underlying latent semantic structure in the data that is partially obscured by the randomness of word choice with respect to retrieval.

4 Background and Related work

Unlike other collections, Cranfield's main feature is a complete judgment.This collection is built using abstracts of aerospace-related documents such as papers, research reports and articles from the collection of the College of Aeronautics, Cranfield, England. (Richmond, 1963). The documents' authors were asked to provide a set of related terms for their documents which were turned into natural language queries (Robertson, 2008). The collection contains 225 queries and 1400 documents.

Using the Cranfield collection, we address the following questions:
Q1: Can we solve limitations of Vector Space Model without Deep Learning through LSA methods?
Q2: How does pretrained Bert models perform on technical dataset and how good is performance of these architectures on custom training against BenchMark models?

4.1 Vector Space Models

Vector space model or term vector model is an algebraic model for representing text documents (and any objects, in general) as vectors of identifiers (such as index terms). It is used in information filtering, information retrieval, indexing and relevancy rankings.

Here Documents and queries are represented as vectors.

$$d_j = (w_{1,j}, w_{2,j} ...., w_{t,j})$$

$$q = (w_{1,q}, w_{2,q} ....., w_{n,q})$$

Each dimension corresponds to a separate term. If a term occurs in the document, its value in the vector is non-zero. Several different ways of computing these values, also known as (term) weights, have been developed. One of the best known schemes is tf-idf weighting.

This vector space model has the following advantages over the Standard Boolean model: Simple model based on linear algebra, Term weights not binary, Allows computing a continuous degree of similarity between queries and documents, Allows ranking documents according to their possible relevance, Allows partial matching. Thought there are advantages there are also disadvantages which prompt us to look into different models.[4]

Of these shortcomings we could tackle a few of them to improve our performance which are Semantic sensitivity as Vector Space Model(VSM) is based on term vocabulary, because of which the documents with similar context but different terms(synonyms) would not be associated, resulting in reduce of recall. Word Relatedness as VSM assumes the words in the documents are independent, which results in retrieval of irrelevant documents, resulting in reduce of precision. Collocations and Cooccurrences as VSM(especially uni-gram approach) disregards the Collocations and Cooccurrences of words as it just takes appearance of word in document into consideration.

Apart from these main issues we can also address, Computation complexity as consideration of all words are impractical as each word is a dimension results in very high dimensional space which is computationally expensive. Model flexibility as each time a new word is encountered the term space needs to be recalculated for all vectors through the model described below.

4.2 Latent Semantic Analysis

Latent semantic indexing (also referred to as Latent Semantic Analysis) is a novel approach of analyzing a set of documents in order to discover statistical co-occurrences of words that appear together which then give insights into the topics of those words and documents.

It is a technique in natural language processing, in particular distributional semantics, of analyzing relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms. 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. More details of methodology will be discussed in following sections.[5]

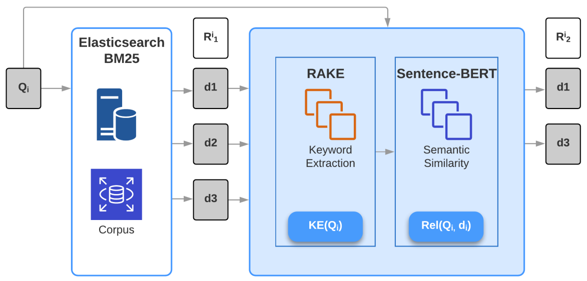
Two main problems (among several) that LSI sets out to solve are the issues of synonymy and polysemy. LSI is able to statistically predict which meaning of a word represents by statistically analyzing the words that co-occur with it in a document.

4.3 Deep Learning based modelling

BERT is a state of the art language model which has achieved groundbreaking results in many NLG and NLI tasks.The BERT architecture and pre-training strategy set the de facto standard for how to generate rich token embeddings utilising an enormous corpus. Its architecture lends itself to be adopted for different kinds of tasks, either through adding task-specific tokens in the input or task-specific networks to the end of the model, utilising its token embeddings. These modifications allow us to use BERT for, just to name a few, classification, regression, and sentence similarity.

Sentence-BERT is one such model which enable us to derive the semantically meaningful sentence embeddings by modifying original BERT using Siamese networks. With Sentence-BERT (SBert) we can now take advantage of BERT embeddings for the tasks like semantic similarity comparison and information retrieval via semantic search.

BERT makes up the base of this model, to which a pooling layer has been appended. This pooling layer enables us to create a fixed-size representation for input sentences of varying lengths.Various pooling strategies and number of linear layers top of that can be experimented.This architecture was trained on combining datasets Stanford Natural Language Inference (SNLI) with the Multi-Genre NLI (MG-NLI) to create a collection of 1,000,000 sentence pairs.Loss functions can also be experimented. Thus for every document,sentence or query embedding can be generated by using Word2Vec or GloVe vector representation of words and further using the pretrained/custom trained SBert model.



IR-Bert Architecture

Recent research work are building upon Siamese-Bert architecture by applying various improvisation algorithms in other pipelines of sentence embedding such data preprocessing or word-vector representations.In latest research work Bert and SBert models are used after implementing NER or Keyword extraction algorithms on sentences.
The paper **IR-BERT: Leveraging BERT for Semantic Search in Background Linking for News Articles**,2020 uses **RAKE** algorithm for keyword extraction and generates embedding on taking these keywords as sentences.

5 Proposed Methodology

There are many classical NLP methods to tackle the issues of word relatedness like: ESA(Explicit Semantic Analysis), Wordnet based models, LSA(Latent Semantic Analysis). Of which our proposed approach is LSA - Latent Semantic Analysis as mentioned before.

5.1 Idea:

The core idea is to take a matrix of what we have documents and terms and decompose it into a separate document-topic matrix and a topic-term matrix.

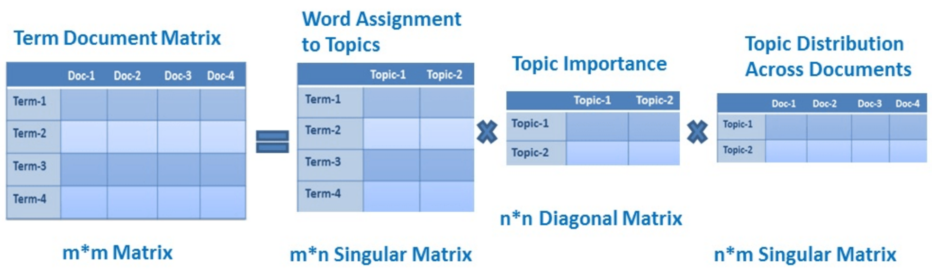
5.2 Approach:

This proposed approach treats the unreliability of observed term-document association data as a statistical problem. The assumption used here is each document consists of a mixture of topics, and each topic consists of a collection of words known as latent topics and words which are partially obscured by the randomness of word choice with respect to retrieval. This description of latent semantic structure after getting rid of noise is used for indexing and retrieval.

5.3 Steps:

The first step is generating our document-term matrix. Given m documents and n words in our vocabulary, we can construct an $m \times n$ matrix A in which each row represents a document and each column represents a word. In the simplest version of LSA, each entry can simply be a raw count of the number of times the j-th word appeared in the i-th document. In practice, however, raw counts do not work particularly well because they do not account for the significance of each word in the document. Hence we plan to use document term frequency matrix with TF-IDF score. Once we have our document-term matrix A, which most likely will be very sparse, very noisy, and very redundant across its many dimensions. As a result, we can find few latent topics that capture the relationships among the words and documents, by performing dimensionality reduction on A.

This dimensionality reduction can be performed using truncated SVD. SVD, or singular value decomposition, A=U*S*V, where S is a diagonal matrix of the singular values of A. Critically, truncated SVD reduces dimensionality by selecting only the t largest singular values and only keeping the first t columns of U



and V. which in our sense corresponds to t highly related documents and words. Where t is a hyperparameter which can be adjusted accordingly.

Pseudo code:

```
$U, S, V = SVD(TFIDF)$
$k = ?(selecting criteria 70 percent variance)$
$u = U[:, : k]$
$s = U[: k]$
$v = U[: k, :]$
$w = u * s$
querytokens = np.where(queryTFIDF != 0)
query = dot(queryTFIDF * W[querytokens, :]
cosinesimilarty = v * query/norm(v)/norm(q)
orderedDocuments = argsort(cosinesimilarity)[::-1]
```

In this case, U $\in R^{mXt}$ emerges as our document-topic matrix, and V $\in R^{nXt}$ becomes our term-topic matrix. In both U and V, the columns correspond to one of our t topics. In U, rows represent document vectors expressed in terms of topics; in V, rows represent term vectors expressed in terms of topics.[7]

5.4 Measures:

With these document vectors and term vectors, we can now easily apply measures such as cosine similarity to evaluate: 1)the similarity of different documents 2)the similarity of different words 3)the similarity of terms (or “queries”) and documents (which becomes useful in information retrieval, when we want to retrieve passages most relevant to our search query).

5.5 Addressed:

From the measures mentioned above we have solved the problem of word relatedness and as LSA considers the documents that have many words in common to be semantically close this solves the issue of semantic sensitivity like the synonyms and also collocations and cooccurrences. Hence where a plain keyword search will fail if there is no exact match, LSI will often return relevant documents that don't contain the keyword at all.

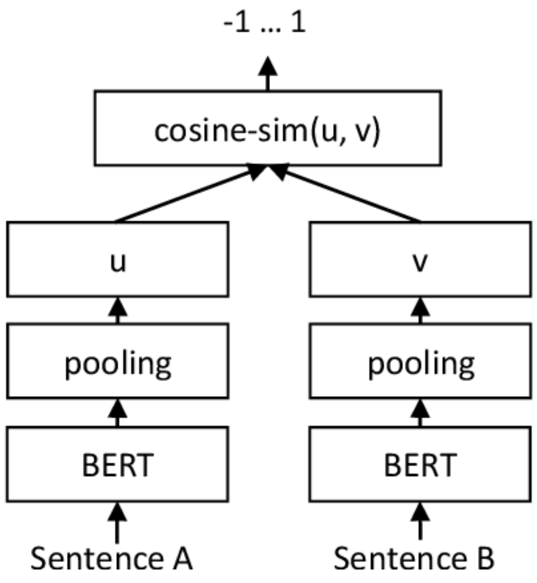
5.5 Deep Learning:

5.5.1 Pretrained Embeddings:

Using the pretrained Sentence Bert models,trained on the general corpora SNLI+MGNLI dataset through Siamese Bert architectures and various assosiated variants we generate embedding for documents and queries. Once embedding vector for documents and queries are obtained, iterating over the queries we rank the documents in accordance with user-defined metric such as cosine similarity or euclidean distance between the query embedding vector and document embedding vector generated. Top-k results are retrieved from the ranked documents.

5.5.1 Custom Training:

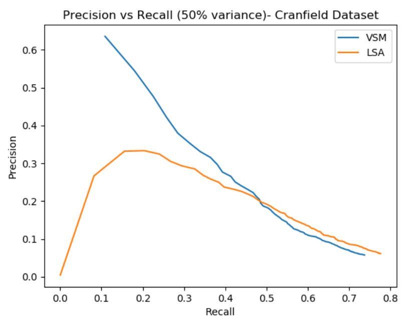
We train a custom model with BERT as base architecture and further customising it and generate embeddings from that. For this purpose, we implemented the paper **BERT meets Cranfield:Uncovering the Properties of Full Ranking on Fully Labeled Data.** The architecture was built upon Bert Siamese-Bert architecture :

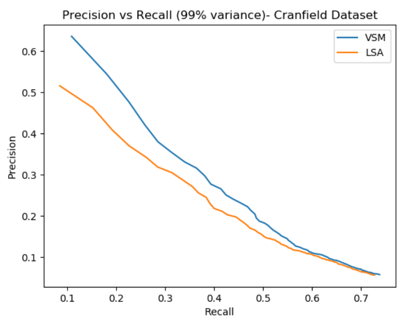


Siamese Bert Architecture

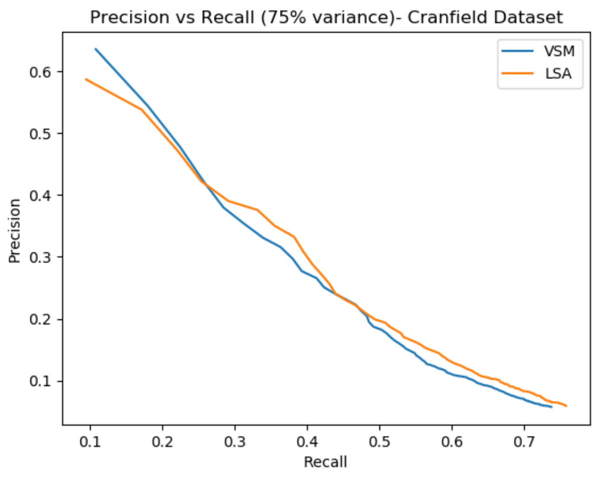
6 Experiments

We ran many experiments to select the criteria for the value of k as described in the above methodology, which is one main parameter that impacts model performance. Value of k is the number of principal components to be considered or the amount of variance of the original matrix to be stored or taken. there are various effects of taking high k and low k because of which we need to take optimal value of k.





Taking a low value of k might not capture entire essence or the main latent concept and latent variable of the data set resulting in poor precision, in the same sense taking high value of k might result in consideration of unimportant latent concepts or terms resulting in poor recall and also computationally heavy. In view of above discussion, plots have been plotted of precision vs recall curve for 3 different values of k based on variance low(50 percent), mid(75 percent), high(99 percent).

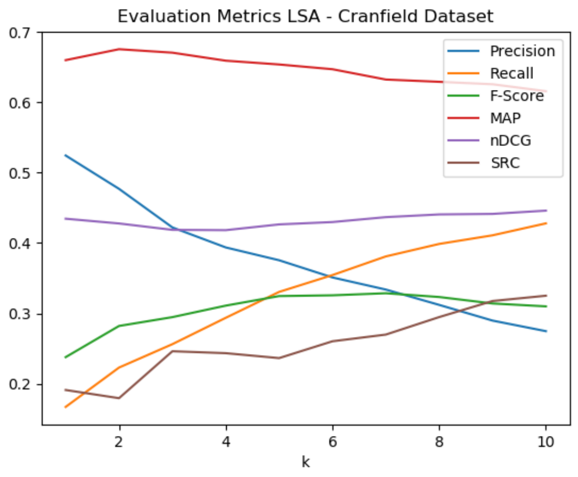


As noticed above after conducting various experiments we notice that the optimal value of k is when it captures around **72 percent of the variance** and following results are for the same.

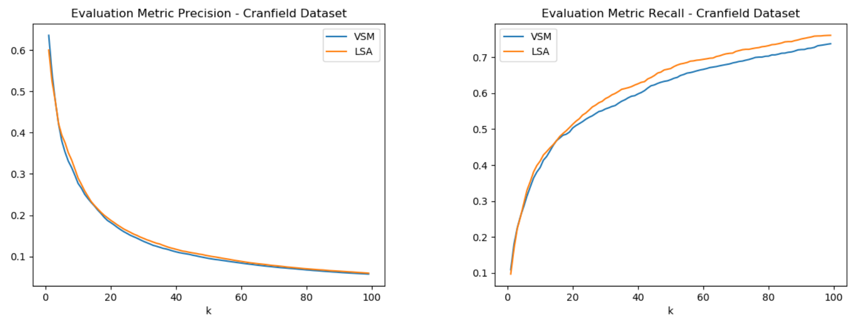
Comping to BERT : Completely pretrained BERT model was used to generate embeddings and evaluation metrics were obtained. Experiments were run over several versions and methods of pretrained models(trained on different types of datasets). Custom training of Bert model was run over several combinations of hyper-parameters.

7 Results

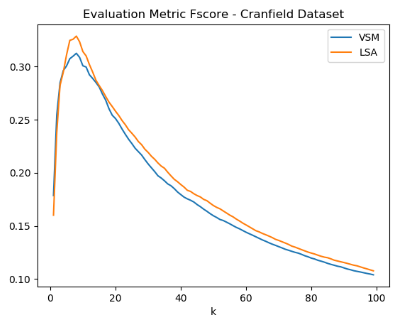
After implementing our classical NLP Latent Semantic Analysis model various metrics such as Precision, Recall, Fscore, nDCG, MAP and Spear Man Correlation have been chosen to evaluate the model based on the ideal data present in cranfield dataset and the results obtained are plot below vs k which here is count of retrieved documents.

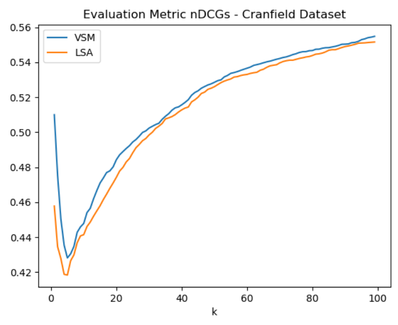


To understand the performance improvement we compare Latent Semantic Analysis model over widely used classical vector space model. As shown below we notice slightly improved performance, this can be attributed to the fact that cranfield documents can be grouped into 10-15 topics hence making it highly correlated and good dataset to check synonym and polysemy effect on LSA[6].

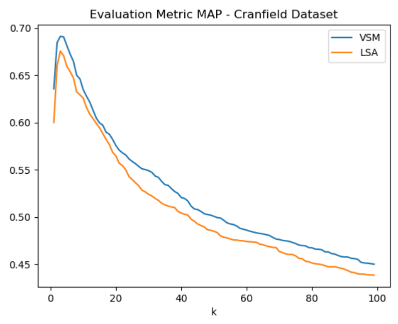


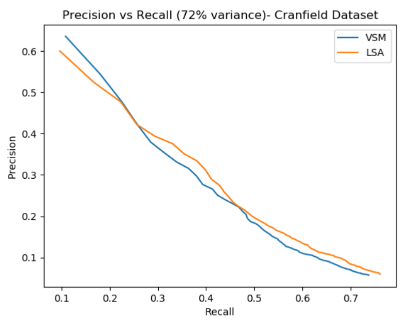
Finally precision vs recall curves for both VSM and LSA models plotted together as below, shows us that for lower values k i.e, for few retrieved documents





VSM performs better but for larger values of k LSA performs better with better recall which means we get more relavant documents compared to VSM in LSA.

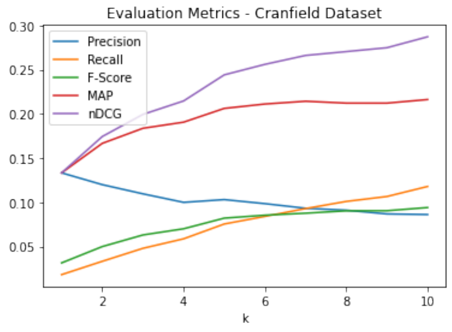




Finally coming to Spearman Rank Correlation metric, which takes into account the relevance degree of the documents into account. Similar to the nDCG which also takes relevance degree of the documents into account measure where VSM and LSA performs almost same in that record. Hence we can estimate the same in the case of Spearman Rank Correlation as shown below for the retrieved documents of 14-16 where LSA performs slightly better.

SRC	VSM	LSA
k = 14	0.344	0.347
k = 15	0.35	0.353
k = 16	0.354	0.357

Results from Deep Learning based model experiments are summarised below: When using fully pretrained embeddings evaluation metrics were quite similar for all versions and type of datasets upon which model pretraining was done.



It is evident that these completely pretrained models are not performing good on Cranfield Dataset.
Results of ndcg score over a combination of hyper-parameters are summarised in below table (Due to limitations experiment was run for 1 epoch).

Learning Rate/MAX-LENGTH	$2e^-1$	$2e^-3$	$2e^-5$
32	0.4701	0.4862	0.5144
64	0.4713	0.5124	0.52
128	0.5104	0.5347	0.5601

So far we have tested the method only with queries formulated to be used against other retrieval methods; The method almost certainly could do better with queries in some more appropriate format. Also the representation of documents by LSI is economical; each document and term need be represented only by something on the order of 150 to 250 values. Through our understand of this method we can say that LSI method deals nicely with the synonymy problem, it offers only a partial solution to the polysemy problem. It helps with multiple meanings because the meaning of a word can be conditioned not only by other words in the document but by other appropriate words in the query not used by the author of a particular relevant document. The failure comes in the fact that every term is represented as just one point in the space. That is, a word with more than one entirely different meaning (e.g., “bank”), is represented as a weighted average of the different meanings[2].

8 Conclusions

Concluding, In this project we tried to implement a statistical model of tackling the shortcomings of classical Vector space model which is Latent Semantic Analysis. To gain better understanding of the concept and how research ideas are formed, improved upon, hypothesized, evaluated. We understand through this concept that this statistical approach namely SVD has a way of dealing with problems of multiple terms referring to same meaning. As a result it solves the issues of synonym and partially the issue of polysemy hence it can be a regarded as a potential component of a retrieval system, rather than as a complete retrieval system as such.

Results from experiments on Deep Learning models suggest that though the pretrained word embeddings are producint SOTA results over general datasets and other experiments, in an entirely technical data corpus such as Cranfield Dataset the pretrained embeddings are performing poor than traditional vector space model. At the same time, custom training indicates that embeddings generated specific to this corpus could produce better results and incorporating more sophisticated algorithms in preprocessing parts could help produce SOTA results.

Finally based on our experimentation and evaluation done using various metrics, we can say with confidence that latent semantic indexing method and custom trained embedding based on corpus are superior to simple term matching methods such as vector space model. In future these methods could be evaluated on much larger datasets for better understanding of the effects and problems it tackles since we had limited ourselves to a small Cranfield dataset.

References

1. Author, Tomek Strzalkowski : Natural Language Information Retrieval. https://link.springer.com/content/pdf/bfm%3A978-94-017-2388-6%2F1.pdf (1999)
2. Susan T. Dumais*, George W. Furnas, and Thomas K. Landauer : Indexing by Latent Semantic Analysis - https://courses.iitm.ac.in/pluginfile.php/344755 /mod_resource/content/1/deerwester-jasis90.pdf
3. Improving Information Retrieval System Performance with Concepts : https://jeffreymorgan.io/articles/information-retrieval-concept-matching/
4. Vector Space Model : https://en.wikipedia.org/wiki/Vector_space_model
5. Latent Semantic Analysis : https://en.wikipedia.org/wiki/Vector_space_model
6. Than Than Wai; Sint Sint Aung : Enhanced frequent itemsets based on topic modeling in information filtering : https://en.wikipedia.org/wiki/Vector_space_model
7. Jay Gopalakrishnan : Latent Semantic Analysis : http://web.pdx.edu/ gjay/teaching/mth271_2020/html/17_LSA.html
8. Negin Ghasemi : BERT meets Cranfield : https://github.com/pp30/NLP-Wikipedia_Relatedness
9. Pyspellchecker : https://pypi.org/project/pyspellchecker/
10. Faruqui, Manaal and Dodge, Jesse and Jauhar, Sujay K. and Dyer, Chris and Hovy, Eduard and Smith, Noah A. : Retrofitting Word Vectors to Semantic Lexicons : https://github.com/pp30/NLP-Wikipedia_Relatedness
11. IR-BERT: Leveraging BERT for Semantic Search in Background Linking for News Articles, https://arxiv.org/pdf/2007.12603.pdf
12. BERT meets Cranfield:Uncovering the Properties of Full Ranking on Fully Labeled Data,https://djoerdhiemstra.com/wp-content/uploads/eacl2021swr.pdf
13. ElasticBERT: Information Retrieval using BERT and ElasticSearch, https://towardsdatascience.com/elasticsearch-meets-bert-building-search-engine-with-elasticsearch-and-bert-9e74bf5b4cf2
14. NEURAL RANKING MODELS FOR DOCUMENT RETRIEVAL, https://arxiv.org/pdf/2102.11903.pdf