PROJECT REPORT

IMPROVEMENT IN VECTOR SPACE MODEL BY LSA AND ESA IMPLEMENTATION

Member1 - Hindanjali Harwanshi, Rollno. - CS20M025
Member2 - Krishna Narwani, Rollno. - CS20M031

1. INTRODUCTION

1.1 Introduction to VSM

Vector space model is a simple model based on linear algebra which represents the documents into vectors of words where each dimension corresponds to a separate term and then transforms to the numerical format using the tf-idf (term frequency-inverse document frequency) weighting scheme, which is intuitive and not very formal. Further, the similarity can be computed by computing the cosine similarity measure.
Being a simple linear algebra model it has many limitations such as it is calculation intensive, can result in false negative matches, term independence etc. The vector space model can be further extended by generalised vector space model, latent semantic analysis etc.

1.2 Introduction to LSA and ESA

Latent semantic analysis(LSA)

Latent semantic analysis(LSA) (Landauer and Dumais,1997) is an approach to information retrieval and automatic indexing that tries to overcome some of the problems of the vector space model(as mentioned above) by mapping the terms and document to a representation called latent semantic space. It extracts and represents the contextual meaning of words by statistical computations applied to a large corpus without explicit directions. It usually takes the high dimensional vector space representation of documents based on term frequencies and applies the dimensionality reduction, which fits the data best as possible but is not perfect.

Explicit Semantic Analysis(ESA)

Explicit Semantic Analysis(by Gabrilovich and Markovitch in 2007) operates under the assumption that the knowledge base contains orthogonal concepts. ESA represents and compares texts as vectors in a high dimensional concept space. It utilizes the wikipedia knowledge base to represent the semantics of a word by a vector where every dimension refers to an explicitly defined concept like a wikipedia article. As mentioned earlier, it assumes the wikipedia concepts are orthogonal to each other ie. two words are related only if they occured in the same articles.

2. Problem Definition

The vector space model has few limitations. Some problems are the problem of synonymy of two terms, the independence between the terms and the sparse representation of vectors.The words can have similar context but the exact word may not be present but can have the word which can be replaced with the other word that matches completely and correctly to that context, these words are called the synonymy which causes a problem in the vector space method. The terms are treated as independent of each other which should have been treated as dependent to related terms for better context understanding, which is not there in the vector space model. The sparsity also degrades the vector space model which is a matter to address. Our goal is to improve these shortcomings of the vector space model in terms of efficiency of retrieval by addressing these problems of the vector space model.

3. Motivation

The classical tf-idf method(vector space model) has few limitations as observed in the performance of it. The nDCG score was 0.72 in the trivial method, with the intention of improvisation in it, our focus was new methods to realize this improvement.

On further analysis we found the actual retrieval failures in the previous information retrieval system. These failures are due to the problem of synonymy, the word with similar context can be present but does not match exactly with the word which leads to degrading the performance of the classical method. The classical approach has independence of words, i.e. it takes each term independently. It does not account for the dependence in terms, which can be addressed by the new methods. One more concert is the sparse representation of the matrix in the classical approach which leads to the motivation of the new approach. In this new approach we will try to address all these problems which may further lead to the improved system.

4. Background and Related work

4.1 Mathematical Details of LSA

LSA takes input as m x n matrix where each entry m_(ij) is the term frequency count of word i in document j. This local frequency is then converted to its local weights of each word i in document j, which is given by

$$\text{loc\_weight}_{ij} = \log(\text{loc\_freq}_{ij}+1)$$

After this, the global weight of each word is calculated, which given by

$$\text{global\_weight} = \left(1+ \sum_{j=1}^{n} a_{ij} . \log(a_{ij}) \right) / \log(n)$$

Here, the quantity a_{ij} is defined as local frequency of word i divided by global frequency of that word in all documents j

$$a_{ij} = (\text{loc\_freq}_{ij}) / \sum_{j=1}^{n} \text{loc\_freq}_{ij}$$

Therefore, the weighted value is given by the division of local weight by global weight , ie.

$$\text{weight}_{ij} = \text{loc\_Weight}_{ij} / \text{global\_ weight}_{ij}$$

Term weight is directly related to local weight and inversely proportional to the global weight, i.e. a word is highly weighted if it occurs frequently in the document compared to the other words in the document and infrequently across all other documents.

Once this is done, the matrix is factored using singular value decomposition(SVD).
SVD is a method to decompose the matrix into three matrices U,Σ,V^T which is given by

$$M = U \cdot \Sigma \cdot V^T$$

where the U is mxm and has orthonormal columns , Σ is mxn diagonal matrix and V^T is nxn and has orthonormal columns. The values of non zeroes entries in Σ after applying SVD are called as singular values, which are collected and ordered as greatest to least along the main diagonal. The dimensionality can be reduced by replacing some of the singular values with 0 from least to greatest. Thus, resulting in

$$M_k = U \Sigma_k V$$

Similarly the words and documents can be represented as

$$U_k = U \Sigma_k \text{ and } V^T{}_k = \Sigma_k V^T$$

The queries can be transformed to the reduced space as $q^T U \Sigma^{-1}{}_k$, where q^T is term weighted query

The cosine similarity of vectors a and b is calculated as

$$\text{Similarity} = \cos(\theta)= (a.b) / ||a|| \, ||b||$$

where ||a|| is calculated as $\sqrt{a * a}$ and ||b|| is calculated as $\sqrt{b * b}$

The similarity between the vectors a and b is directly proportional to the cosine angle between the vectors a and b, greater the angle greater the similarity value i.e. nearer to 1 the vectors are more synonymous.

Using the cosine similarity we can calculate the query document and query word similarities.

4.2 Mathematical Details of ESA :

Wikipedia is a large diverse knowledge base,in ESA each article is considered as a concept and these concepts are then represented as a vector of words which occurred in the article.

Let D be an index collection of n documents, each of which describes a single concept and V be a vocabulary with m different terms that occur in D. Document x is represented as a concept vector u, where each entry in u corresponds to the inner product of x and d_i, where $d_i \in D$, which is given by

$$u^T = ((d1,x), (d2,x) ,..., (dn,x))$$

where x and d_i are the vector space model representations. Each concept is then represented as vectors which are n-dimensional vectors containing a tf-idf weight for each term in V.

The similarity between the documents x and y is defined by the cosine measure of concept vectors u and v can be found by the cosine similarity as mentioned above.

$$sim(x, y) = \cos(u,v) = (u \cdot v) / ||u|| \, ||v||$$

5. PROPOSED METHODOLOGY

5.1 Lexical Semantic Analysis

Lexical semantic analysis uses the document-term matrix for that first we have calculated the term frequency first, followed by the idf calculation and finally we get our tf-idf(term frequency- inverse document frequency).

On this tf-idf matrix we have applied the randomized SVD for rank =250, which decomposes the matrix with a relatively low rank and is more competitively efficient. Using the formula

$$M_k = U\ \Sigma_k\ V^T$$

After that we have calculated the reduced representation of document space using

$$D\ U_k = \Sigma_k\ D^T$$

Further, we have the query expansion and converted our query to the reduced space similar to the reduced document space using $q^T\ U\ \Sigma^{-1}{}_k$

Then, using the reduced document space and expanded query, for each query we find cosine similarity with respect to each document and using this cosine similarity, for each query we have fetched all relevant documents.

5.2 Explicit Semantic Analysis

Wikipedia is a great source for the articles for fetching these articles related to the domain we have used the wikipedia module from PyPi.

We have used 1652 words from the cranfield dataset titles to fetch 5298 wikipedia articles which are related to our domain. Then we have used 5498 more words from the cranfield dataset body to fetch another 16,683 wikipedia articles i.e. now we have a total of 21,981 wikipedia articles.

The vocabulary size of this 21,981 raw articles (ie. not preprocessed) is equal to the 1,03,625 words. After eliminating the stop words and very common terms we have reduced our vocabulary size to 89,094 words. Further after performing article specific preprocessing vocabulary size reduced to 68,010 words.
But out of these 21,981 articles, due to the memory constraint we can only use 15000 of these articles for the evaluation.

Then, using these 15,000 preprocessed articles we have created the tf-idf(term frequency inverse document frequency) indexing i.e. term - article matrix of size 15000 X 68010.

Using 1400 cranfield dataset and 225 queries we created tf-idf indexing i.e. term-doc matrix of size 1625 x 5498 (where 5498 is the vocab of preprocessed cranfield dataset).

Then using the term-article matrix and term-document matrix, we created one new matrix, *say merged* for the one to one correspondence between the wikipedia articles and documents (including 225 queries) of size 68,010 x 1625.

Further we have calculated the cosine similarity for each query and using this cosine similarity, for each query we have fetched the relevant documents.

6. Experiments

We have performed several experiments on cranfield dataset, which is listed below

1. Performing ESA using 5298 articles and used cranfield dataset titles
2. Performing ESA using 5298 articles and used cranfield dataset body
3. Performing ESA using 15000 articles and used cranfield dataset titles
4. Performing ESA using 15000 articles and used cranfield dataset body

The result of experiments can be visualised below on Fig. 1.

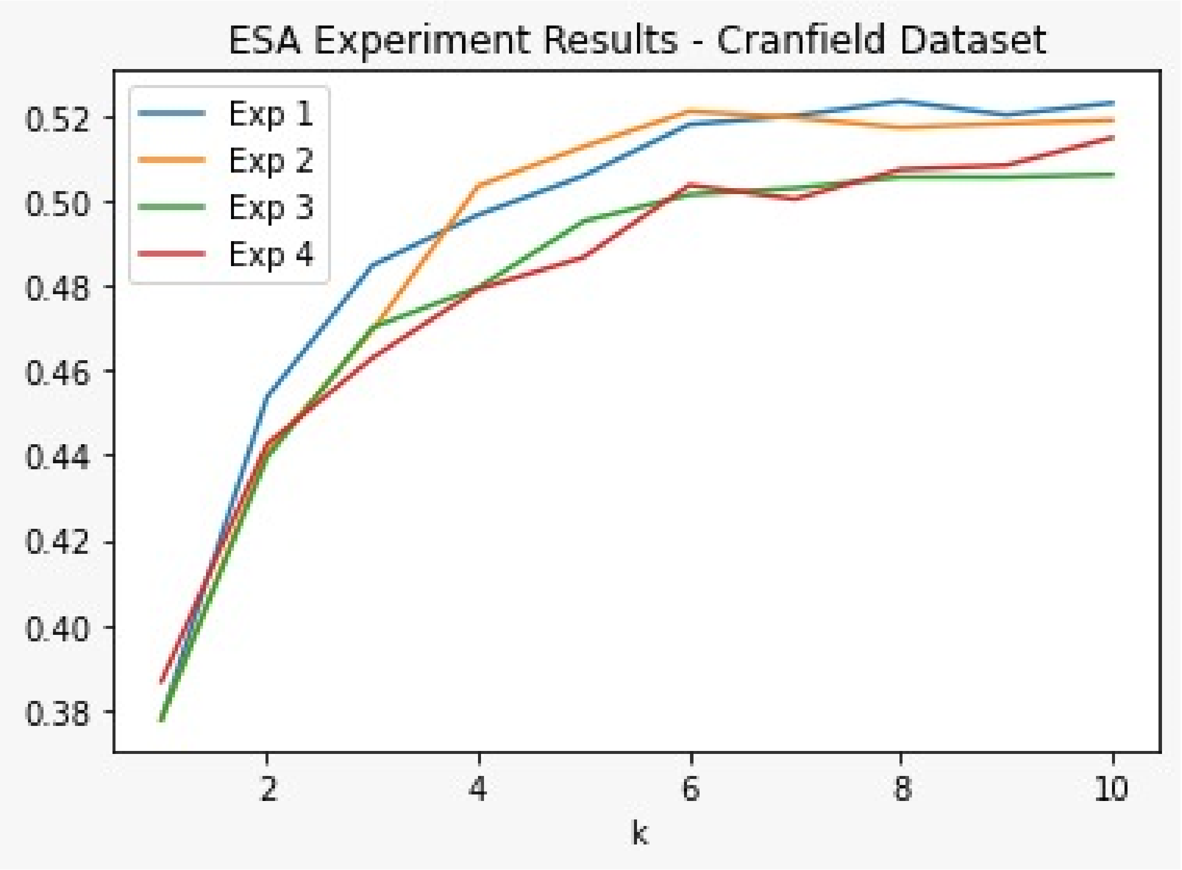

Fig.1 ESA Experiment Result

7. RESULT

The result obtained previously from the tf-idf on cranfield dataset can be visualized below in fig. 2. We can see that the precision is decreasing and recall is increasing with increase in k.

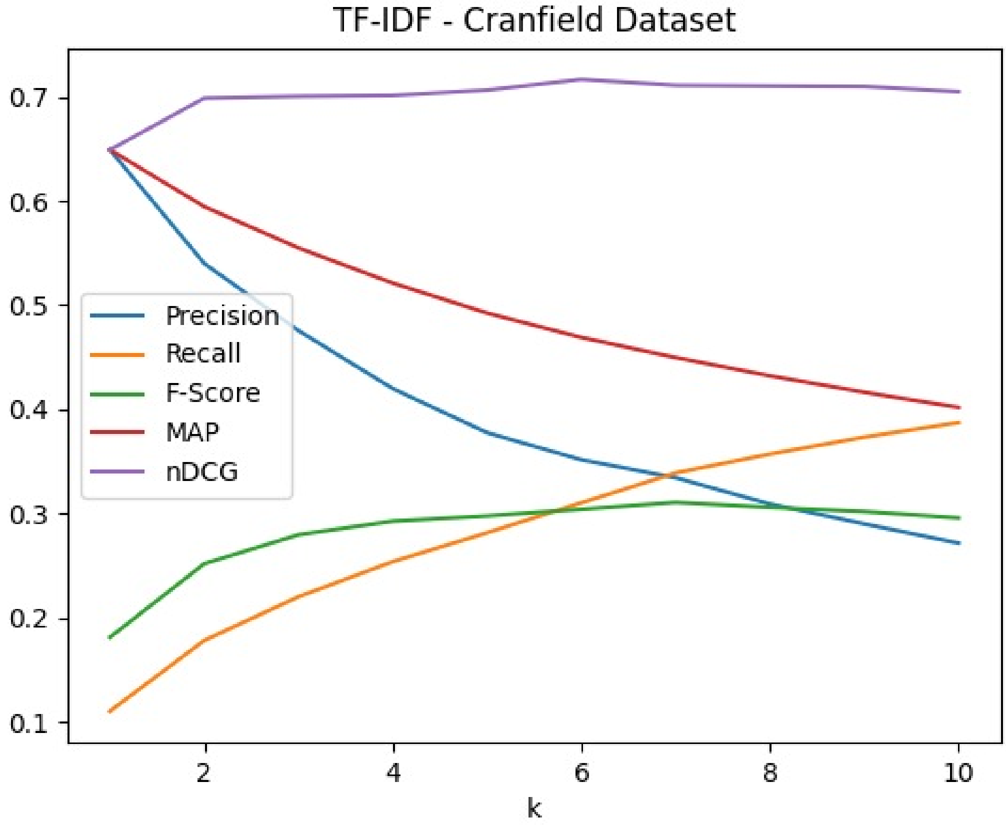

Fig. 2. TF-IDF on cranfield dataset

The result obtained for the LSA on cranfield dataset can be visualised below in Fig. 3.

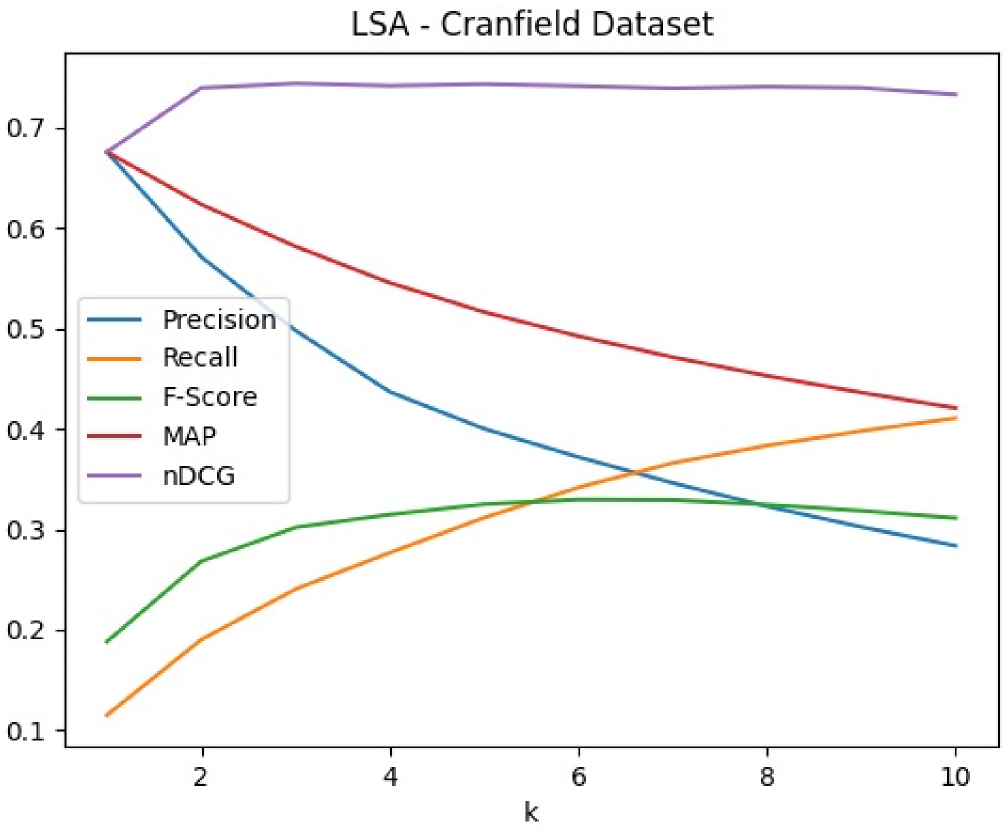

Fig. 3. LSA on cranfield dataset

We can clearly compare the tf-idf and LSA, the increase in nDCG score and recall in LSA is clearly visible when compared to the previous method, which means that the performance is improved in LSA.

Further when we perform the ESA on the cranfield dataset we can observe the poor performance as compared to tf-idf , the decrease in nDCG and recall is clearly visible in the Fig. 4.

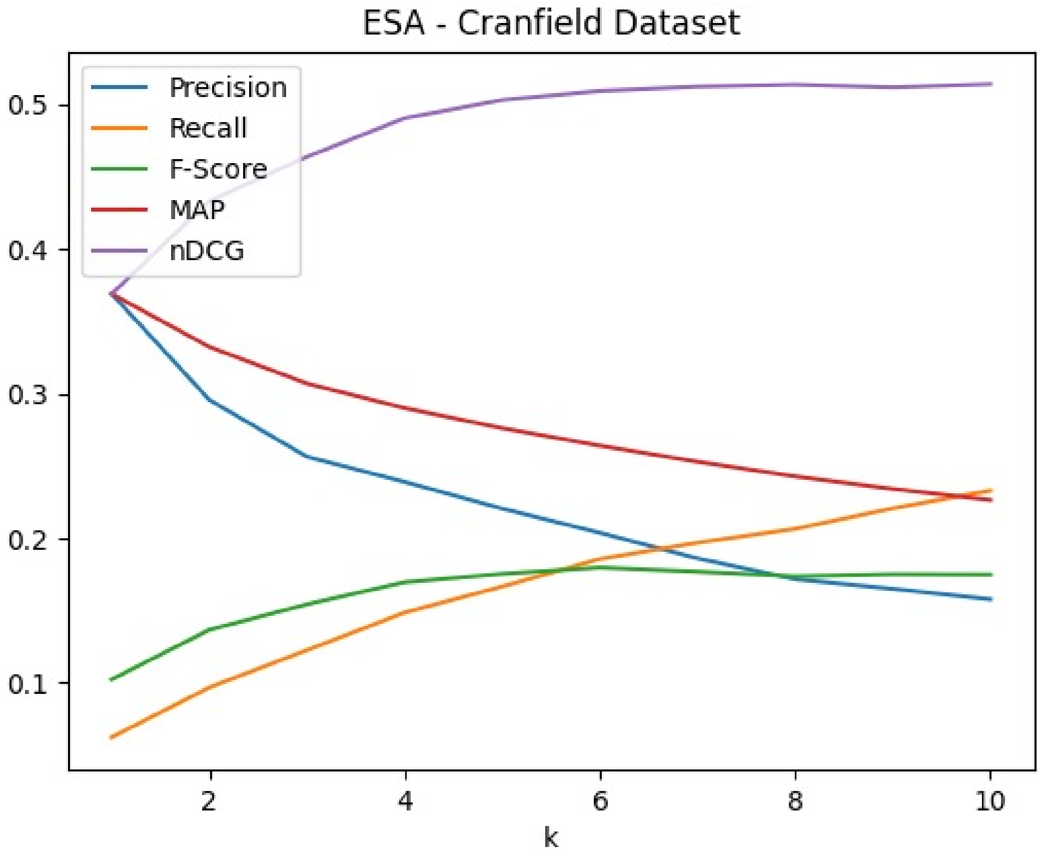

Fig. 4. ESA on canfield dataset

The overall result for the comparison of three models namely the tf-idf, ESA and LSA on cranfield dataset can be seen below in Fig. 5.

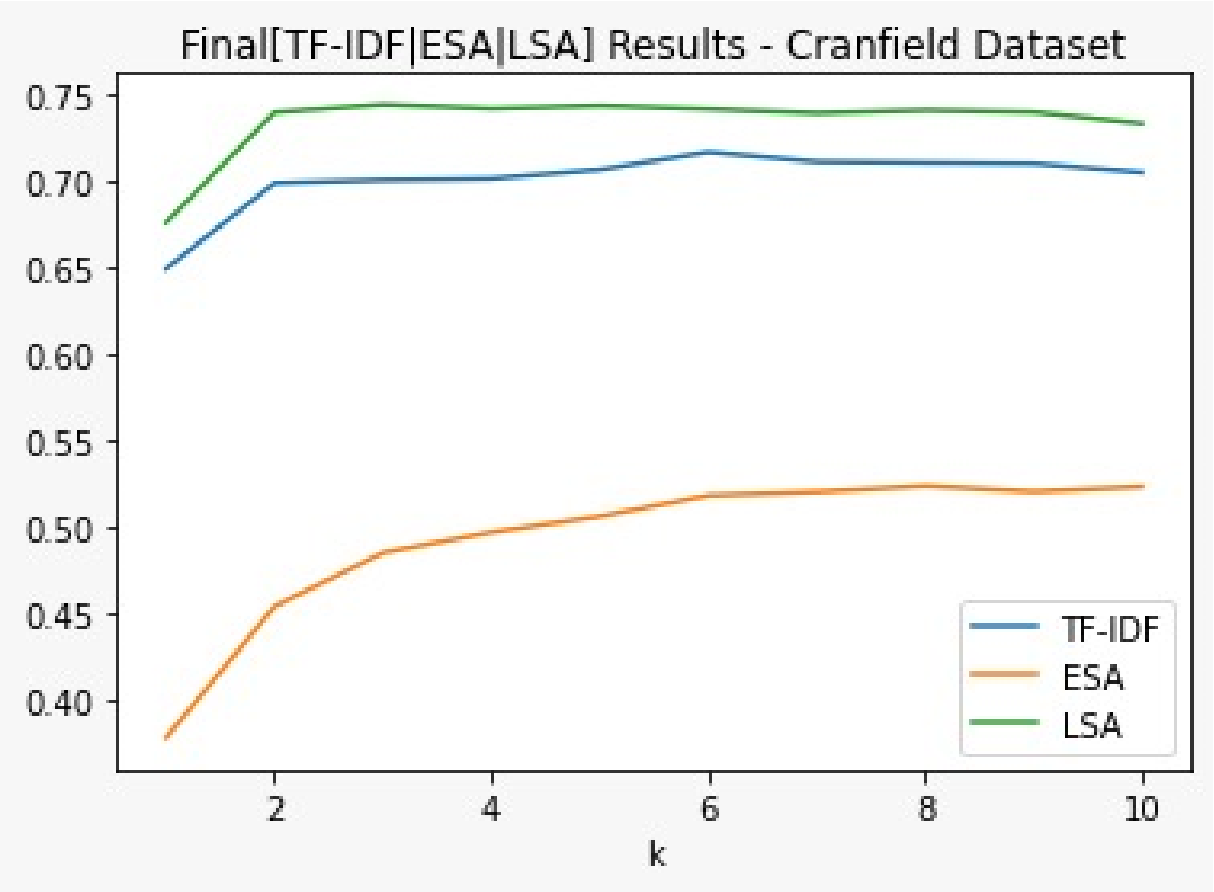

Fig. 5. Comparison of three methods

The comparison is based upon the nDCG score of each method. It is clearly visible that the LSA can perform marginally better than the tf-idf and ESA performs poorly than that of the other two methods.

8. Conclusion

As we have seen in the result that the performance of LSA is better than the tf-idf so we can say that our hypothesis for LSA i.e. for addressing the problem of sparse representation and the problem of synonymy are well addressed and thus there is improvement in the nDCG score. This improvement is due to the well addressel ability of LSA for the problem of synonymy otherwise which causes the problem of mismatch in the vocabulary in the information retrieval system which degrades the performance as seen in tf-idf. Although this improvement in result can not be further improved as LSA can result in some wrong dimensions which can be difficult to interpret the result, another problem is that the LSA can only partially capture the polysemy, otherwise it could be performed much better. We can also conclude that the LSA can not solve some of the problems of vector space models like LSA is also calculation intensive and requires high memory, it also cannot address the problem of bag of words etc. but performs well and solves the mentioned problem very well.

When we compare the ESA and the tf-idf clearly the ESA hypothesis fails, it could not perform better than the prior method. It could not address the problem of synonymy and dependence well, thus resulting in the poor performance. In the analysis of this poor performance of ESA we found that one of the reasons is that ESA considers dimensions as orthogonal to each other. It assumes that the words are only related if they co-occur in the same articles but, these words can also be related if they are not present in the same article but appeared in the related article for example football and soccer do not share the same concept but are related to each other, although this problem is solved by LSA upto some extent and the difference can also be seen in the result. Another problem can be because of the data that we fetched according to some selected vocabulary from the title and body and taking only some top articles on that even also that many articles we cannot take into consideration for the evaluation due to memory constraint, otherwise it could improve the performance.