# Comparative study of Non-Orthogonal Explicit Semantic Analysis(NESA), Explicit Semantic Analysis(ESA) and Latent Semantic Analysis based Information Retrieval Systems*

D Sumanth[1][EE18B046] and Raghu Dinesh T[2][EE18B144]

Indian Institute of Technology, Madras[1,2]

**Abstract.** The goal of this project is to improve our Vector space model based information retrieval system by addressing its current limitations, specially the "low recall that arises because of not considering the synonymy between the words which can be overcome by establishing some sort of relatedness between the words. Some of the methods which we expect to perform well for this task are Latent Semantic Analysis(LSA), Explicit Semantic Analysis(ESA) and Non orthogonal Explicit Semantic Analysis(NESA). We explore all the three methods i.e. LSA, ESA, NESA and compute the concept correlation weights, and compare these approaches with the Vector space model. Queries or data may have some of the words misspelled so we used a spell checker on both data and queries to improve the performance of the system.

**Keywords:** LSA · ESA · NESA · VSM.

## 1 Introduction

An Information Retrieval(IR) system is used to retrieve and rank relevant words/ documents from the data set of interest for any given query. The simplest IR system is to use Term Frequency Inverse Document Frequency(TF-IDF) to represent documents and queries as vectors and calculate the relevance of a query to all the documents within the dataset by using measures like cosine similarity. This TF-IDF based Vector Space model(VSM) performs decently well but it has a few drawbacks as it inherently assumes the words are semantically unrelated which bring the need to develop alternative methods for IR.

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. These top-down resources

* CS6370 Natural Language Processing course project

can be utilized to guide bottom-up expeditions to identify relations among words/documents. Several works in this domain have surfaced recently.

One of the well-known bottom-up approach for capturing semantic relatedness in words is Latent Semantic Analysis (LSA). Explicit Semantic Analysis (ESA) is another approach for capturing semantic relatedness in words/documents. ESA uses for top-down knowledge from Wikipedia for improving word representations. Non-orthogonal Explicit Semantic Analysis (NESA) is another approach which is an extension to ESA. In this paper we compare the performances of three IR systems based on LSA, ESA and NESA to the TF-IDF based vector space model on Cranfield dataset.

## 2 Problem Definition

The TF-IDF based Vector Space model assumes the words are semantically unrelated therefore documents with similar content but different vocabularies (eg: synonyms) will not be retrieved as a result the IR system has a low recall. This can be overcome by establishing some sort of relatedness between the words. This paper produces a comparison between LSA, ESA and NESA which are expected to give a better performance over the TF-IDF based VSM.

## 3 Motivation

Two widely used methods for this purpose are Latent Semantic Analysis (LSA) and Explicit Semantic Analysis (ESA). Both these methods capturing semantic relatedness in words/documents therefore they are expected to perform better over the TF-IDF based VSM.

ESA inherently assumes that Wikipedia concepts are orthogonal to each other, therefore, it considers that two words are related only if they co-occur in the same articles. However, two words can be related to each other even if they appear separately in related articles rather than co-occurring in the same articles. This leads to a need for extending the ESA model to consider the relatedness between the explicit concepts (i.e. Wikipedia articles in Wikipedia based implementation) for computing textual relatedness. Non Orthogonal ESA (NESA) is used to overcome this shortcoming. NESA represents more fine grained semantics of a word as a vector of explicit concept dimensions, where every such concept dimension further constitutes a semantic vector built in another vector space. Thus, NESA considers the concept correlations in computing the relatedness between two words. Therefore all the three approaches have a potential to perform better than the TF-IDF based VSM which motivates to perform a comparative study on them.

## 4 Background and related work

There have been a variety of efforts to develop semantic relatedness measures. Classical approaches assess the relatedness scores by using existing knowledge bases or corpus statistics; Three of them are LSA, ESA and NESA.

Corpus-based methods such as LSA (Landaueret al., 1998), Latent Dirichlet Allocation (LDA) (Blei et al., 2003), and ESA (Gabrilovich and Markovitch, 2007) employ statistical models to build the semantic profile of a word. LSA and LDA generate unsupervised topics from a textual corpus, and represent the semantics of a word by its distribution over these topics. LSA performs singular value decomposition (SVD) to obtain a latent concept space. On the contrary, ESA directly uses supervised topics such as Wikipedia concepts that are built manually, and considers that every concept is orthogonal to each other. Polajnar at el. (2013) proposed an approach to improve ESA by considering the concept relatedness using word overlap in Wikipedia articles' content. [https://www.aclweb.org/anthology/S15-1010.pdf] Employ Non-Orthogonal ESA (NESA) which represents more fine grained semantics of a word as a vector of explicit concept dimensions, where every such concept dimension further constitutes a semantic vector built in another vector space. NESA considers the concept correlations in computing the relatedness between two words and the paper also provides different approaches to compute the concept correlation weights.

### 4.1 Latent Semantic Analysis (LSA)

Latent Semantic Analysis is a natural language processing method that analyzes relationships between a set of documents and the terms contained within. It uses singular value decomposition, a mathematical technique, to scan unstructured data to find hidden relationships between terms and concepts. LSA resolves the orthogonality issue to some extent by building latent concept space in an unsupervised way (Landauer et al., 1998). However, the resulting latent concepts are not as clearly interpretable as the human-labeled concepts in the ESA model

### 4.2 Explicit Semantic Analysis (ESA)

Explicit Semantic Analysis (ESA) 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. ESA inherently assumes that Wikipedia concepts are orthogonal to each other, therefore, it considers that two words are related only if they co-occur in the same articles

### 4.3 Non-Orthogonal Explicit Semantic Analysis (NESA)

Two words can be related to each other even if they appear separately in related articles rather than co-occurring in the same articles. This leads to a need for extending the ESA model to consider the relatedness between the explicit concepts (i.e. Wikipedia articles in Wikipedia based implementation) for computing

textual relatedness. In this paper, we also present Non-Orthogonal ESA (NESA) which represents more fine grained semantics of a word as a vector of explicit concept dimensions, where every such concept dimension further constitutes a semantic vector built in another vector space. Thus, NESA considers the concept correlations in computing the relatedness between two words.

In this paper we implement **NESA-VSM Text**. This approach is based on plain Vector Space Model (VSM) for text. It calculates the relatedness scores between concepts by taking word overlap between their corresponding Wikipedia article content. The concept is transformed to a column vector mx1, where m is the total number of unique words in the Wikipedia corpus. The magnitude of each dimension is calculated on the basis of the number of occurrences of the different words in the associated Wikipedia article content.

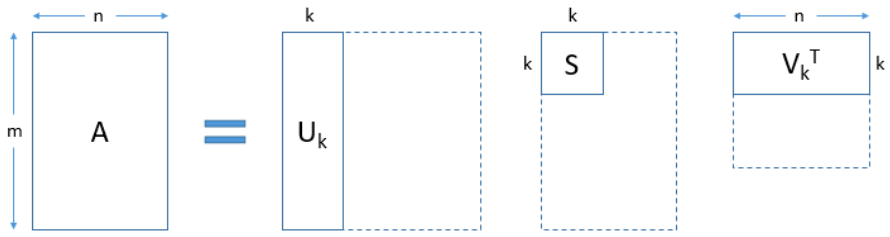
## 5 Proposed Methodology

### 5.1 LSA Implementation

Let's say, In cranfield dataset we have m number of text documents with n number of total unique terms (words). We wish to extract k topics from all the text data in the documents. The number of topics, k, is a hyper parameter and has to be tuned to get the best performance from the model.

From the Cranfield dataset a document-term matrix of shape m x n having TF-IDF scores is generated. Then, we will reduce the dimensions of the above matrix to k (no. of desired topics) dimensions, using singular-value decomposition (SVD). SVD decomposes a matrix into three other matrices. Suppose we want to decompose a matrix A using SVD. It will be decomposed into matrix U, matrix S, and VT (transpose of matrix V).

$$A = U * S * V^T$$



Each row of the matrix Uk (document-term matrix) is the vector representation of the corresponding document. The length of these vectors is k, which is the number of desired topics. Vector representation for the terms in our data can be found in the matrix Vk (term-topic matrix).

So, SVD gives us vectors for every document and term in our data. The length of each vector would be k. To obtain the query vector for each query in the cranfield dataset the product of the IDF and the term vector is calculated for all word in the query and added to obtain the query vector.

Using the document vectors and query vector the relevant documents are retrieved using the cosine similarity method.

### 5.2 ESA Implementation

Basic idea of ESA is that the more similar the words are, they will be found in more common Wikipedia articles. And the importance of a word in an article is also taken into account; determined using TF-IDF (term frequency - inverse document frequency) algorithm. Thus the ESA algorithm takes (latest) Wikipedia dump and determines importance of different words (terms) in different articles using TF-IDF algorithm. 2490 wikipedia articles were used for this task which were extracted from the whole wikipedia corpus by querying with the titles of the cranfield dataset. We also built a ESA model with 4,00,000 randomly documents from the wikipedia corpus but this model was observed to performly poorely compared to the model built on 2490 cranfield dataset titles related wikipedia articles.

An intermediary result is a set of vectors for each Wikipedia article (concept) with corresponding TF-IDF values for words that are found relevant in this article. Then the inverse vector for each word is calculated (containing its TF-IDF values in different articles).

A document vector for each document in the cranfield dataset is calculated by adding the vectors of dimension D corresponding to all the words in the document which are obtained from ESA algorithm. Similarly the vectors for all the queries in the cranfield dataset are also calculated. Using the document vectors and query vector the relevant documents are retrieved using the cosine similarity method.

### 5.3 NESA Implementation

To compute text relatedness, NESA uses relatedness between the dimensions of the distributional vectors to overcome the orthogonality in ESA model.In addition to represent the words as distributional vectors, where each dimension is associated with a Wikipedia concept as in ESA model, NESA also utilizes a square matrix $C_{n,n}$ n is the total number of dimensions containing the correlation weights between the dimensions. Thus, to obtain the relatedness score between the words w1 and w2, NESA formulates the measure as follows:

**relNESA(w1, w2)** = $w1^{T}_{1,n}$*$C_{n,n}$*$w2_{n,1}$

where $w1_{n,1}$ and $w2_{n,1}$ are the corresponding distributional vectors consisting of n dimensions. Every concept dimension can be further semantically interpreted as a distributional vector in some other vector space of m dimensions. This transformation allows the computation of the correlation weights between the concept dimensions. Thus, a transformation matrix Em,n can be built, where each column corresponds to a transformation vector for each concept dimension. Using the matrix $E_{m,n}$, we can compute the matrix $C_{n,n}$ by multiplying Em,n with its transpose.

$C_{n,n} = E^T_{n,m} * E_{m,n}$

To calculate the $E_{m,n}$ we implement NESA-VSM Text approach. This approach is based on plain Vector Space Model (VSM) for text. It calculates the relatedness scores between concepts by taking word overlap between their corresponding Wikipedia article content. The concept is transformed to a column vector mx1, where m is the total number of unique words in the Wikipedia corpus. The magnitude of each dimension is calculated on the basis of the number of occurrences of the different words in the associated Wikipedia article content multiplies by its Inverse Document Frequency (IDF) value. 2490 wikipedia articles were considered same as in ESA. Using the document vectors, query vector and the $C_{n,n}$ matrix the relevant documents are retrieved using the given relNESA formula.

## 6 Experiments

- In LSA, The reduced dimension K is varied and all the Evaluation Metrics are plotted and observed to select the best K.
- In ESA, the different models were built with different number of randomly selected wikipedia articles upto 4,00,000 and cranfield dataset documents title related 2490 wikipedia articles and their performance were compared.
- Precision, Recall, Fscore, MAP, nDCG metrics are calculated and plotted for all the models individually.
- Precision@k and Recall@k where k=1,2,.. are calculated for all the models and plotted on the same plots to observe the relative performance.
- Precision Vs Recall for different models are plotted on one plot to observe the relative performance.

## 7 Results

We used the Precision, Recall, Fscore, MAP and nDCG to evaluate VSM, LSA, ESA and NESA-VSMText based information retrieval systems. The following are the tables of obtained values and the plots of the metrics for each model.

| k | Precision | Recall | Fscores | MAPs | nDCGs |
|---|---|---|---|---|---|
| 1.0 | 0.662222 | 0.111965 | 0.183956 | 0.662222 | 0.530000 |
| 2.0 | 0.548889 | 0.180811 | 0.255574 | 0.704444 | 0.485836 |
| 3.0 | 0.484444 | 0.229116 | 0.289246 | 0.706296 | 0.461385 |
| 4.0 | 0.434444 | 0.264830 | 0.304456 | 0.697654 | 0.448742 |
| 5.0 | 0.384889 | 0.288338 | 0.304279 | 0.695191 | 0.438092 |
| 6.0 | 0.357778 | 0.320186 | 0.311629 | 0.681330 | 0.440686 |
| 7.0 | 0.335238 | 0.349974 | 0.316332 | 0.670745 | 0.444771 |
| 8.0 | 0.318333 | 0.375244 | 0.318000 | 0.662166 | 0.451838 |
| 9.0 | 0.300741 | 0.395400 | 0.315709 | 0.652796 | 0.458944 |
| 10.0 | 0.282667 | 0.409927 | 0.309650 | 0.642723 | 0.461033 |

VSM METRICS

| k | Precision | Recall | Fscores | MAPs | nDCGs |
|---|---|---|---|---|---|
| 1.0 | 0.595556 | 0.094295 | 0.156906 | 0.595556 | 0.448889 |
| 2.0 | 0.520000 | 0.164911 | 0.235119 | 0.655556 | 0.428429 |
| 3.0 | 0.462222 | 0.211427 | 0.269854 | 0.671852 | 0.413164 |
| 4.0 | 0.413333 | 0.245293 | 0.284867 | 0.666420 | 0.404759 |
| 5.0 | 0.383111 | 0.279907 | 0.298952 | 0.662086 | 0.404344 |
| 6.0 | 0.360000 | 0.312073 | 0.308763 | 0.656983 | 0.409349 |
| 7.0 | 0.346032 | 0.345937 | 0.319298 | 0.642203 | 0.418905 |
| 8.0 | 0.324444 | 0.369106 | 0.318827 | 0.631018 | 0.423229 |
| 9.0 | 0.305185 | 0.388574 | 0.315959 | 0.617799 | 0.428385 |
| 10.0 | 0.286667 | 0.404361 | 0.310320 | 0.612597 | 0.433290 |

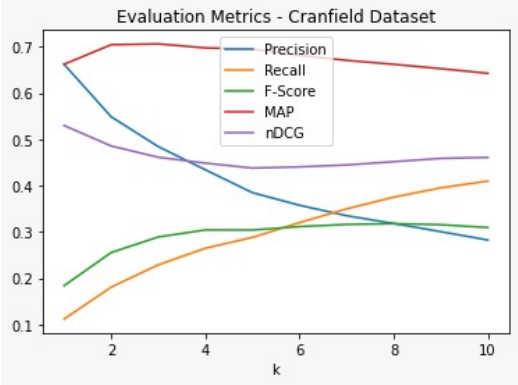
LSA METRICS

| k | Precision | Recall | Fscores | MAPs | nDCGs |
|---|---|---|---|---|---|
| 1.0 | 0.248889 | 0.039745 | 0.065450 | 0.248889 | 0.176667 |
| 2.0 | 0.204444 | 0.064991 | 0.092962 | 0.282222 | 0.169111 |
| 3.0 | 0.170370 | 0.082901 | 0.103687 | 0.288889 | 0.158552 |
| 4.0 | 0.151111 | 0.093048 | 0.106737 | 0.293827 | 0.156641 |
| 5.0 | 0.144889 | 0.108161 | 0.114313 | 0.302525 | 0.158813 |
| 6.0 | 0.137037 | 0.120206 | 0.118036 | 0.299206 | 0.159614 |
| 7.0 | 0.128889 | 0.129801 | 0.119245 | 0.299488 | 0.161474 |
| 8.0 | 0.120000 | 0.135369 | 0.116860 | 0.299163 | 0.162424 |
| 9.0 | 0.113086 | 0.143147 | 0.116229 | 0.297181 | 0.164278 |
| 10.0 | 0.108444 | 0.151507 | 0.116535 | 0.294766 | 0.166810 |

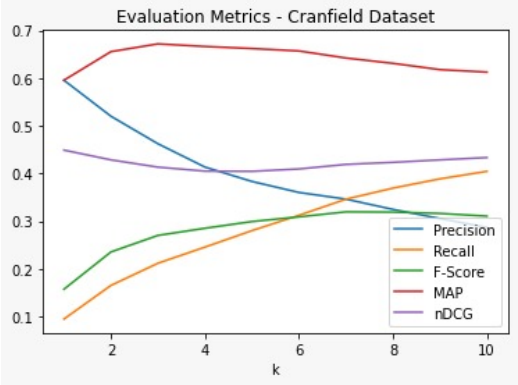
ESA METRICS

| k | Precision | Recall | Fscores | MAPs | nDCGs |
|---|---|---|---|---|---|
| 1.0 | 0.133333 | 0.022056 | 0.036852 | 0.133333 | 0.086667 |
| 2.0 | 0.100000 | 0.032658 | 0.047070 | 0.151111 | 0.078877 |
| 3.0 | 0.088889 | 0.042202 | 0.054069 | 0.161111 | 0.076234 |
| 4.0 | 0.084444 | 0.053629 | 0.061789 | 0.163210 | 0.079776 |
| 5.0 | 0.078222 | 0.065869 | 0.066752 | 0.161531 | 0.082238 |
| 6.0 | 0.072593 | 0.070581 | 0.066724 | 0.165049 | 0.082443 |
| 7.0 | 0.067302 | 0.075598 | 0.066490 | 0.167166 | 0.083613 |
| 8.0 | 0.062778 | 0.078862 | 0.065097 | 0.166092 | 0.084878 |
| 9.0 | 0.064198 | 0.089445 | 0.069218 | 0.164286 | 0.089700 |
| 10.0 | 0.061333 | 0.094339 | 0.069156 | 0.162391 | 0.091343 |

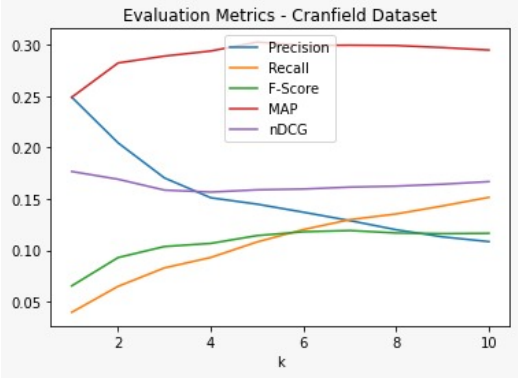
NESA METRICS



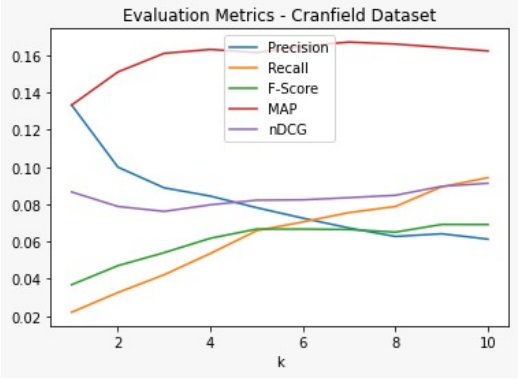

VSM Metrics Graph




LSA Metrics Graph

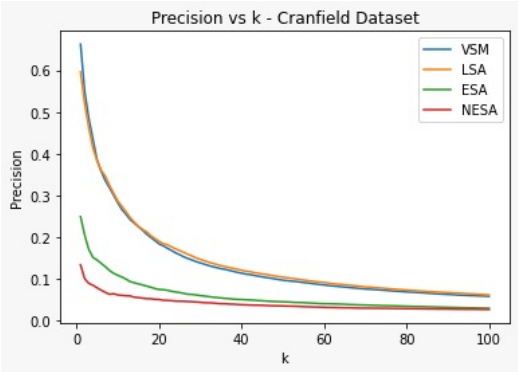



ESA Metrics Graph

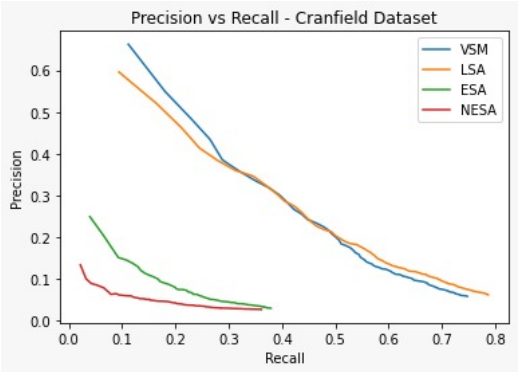



NESA Metrics Graph

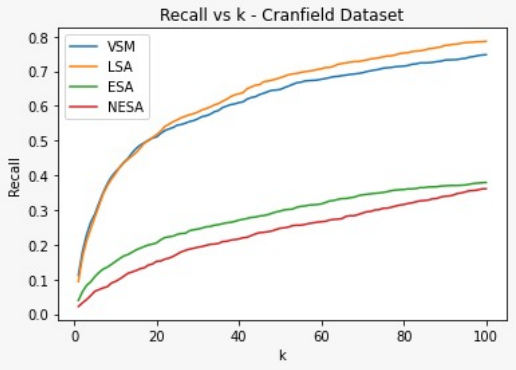
The following are the Precision Vs K, Recall Vs K and Precison Vs Recall plots obtained for all the models.




Precision vs K Graph




Precision vs Recall Graph




Recall vs K Graph

## 8 Conclusion

We presented LSA, ESA and NESA-VSMText. From the Precision and Recall graphs it can be observed that LSA has higher precision and recall over than Vector space model for large k whereas vector space model was still performing better for smaller values of k. It can be generalized that LSA and VSM models outperformed ESA and ESA outperformed NESA-VSMText. This problem is also pointed in the NESA paper($https://www.aclweb.org/anthology/S15-1010.pdf$). NESA-VSMText does not capture the semantics of Wikipedia concepts as the textual description in Wikipedia article also contains generic terms

which are not enough to specify the precisely semantics of Wikipedia concepts which resulted in lower accuracy when compared to ESA. Better variations of NESA (NESA-DISER) can be implemented which can outperform both ESA and LSA. The poor performance of the ESA could be because only a small size of the wikipedia corpus was used due to which the semantic relatedness between the words is not captured properly as expected. This could be overcome by building the ESA model over all the 65,85,000 wikipedia articles using parallel processing and high-performance computers.