Improvements of Vector Space Model for Information Retrieval

Manideep Ladi and Milan Chartterjee

Indian Institute Of Technology,Madras

1 Introduction

Information retrieval (IR), one of the NLP advanced techniques, is defined to be the science of enhancing the effectiveness of term-based document retrieval. It could be also defined as " finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)."
Vector Space Model suffers from the following limitations :
1. Long documents are poorly represented because they have poor similarity values.
2. Curse of dimensionality becomes more prominent as doc size increases dimensionality increases,so , the volume of the space increases so fast that the available data become sparse.
3. It tries to match the exact words and does not take care of the context in which words are used . So it does not take care of semantic sensitivity. Ie. synonyms will be considered as different word all together
4. It assumes axes are orthogonal i.e. words/terms are statistically independent.
So there are different varieties of model proposed to overcome these limitations. Some of them being Latent Semantic Analysis(LSA), Explicit Semantic Analysis (ESA), Query Expansion .... etc . So in this project we are going to explore some of them.

2 Problem definition

To explore different models that can address different limitations of the Vector Space Model. Compare the evaluation metrics obtained from different model with the Vector Space Model and report the model best suited for Cranfield Dataset. There are different evaluation parameter , in this project our goal is to improve the performance of our search engine on carnfield dataset by make as many evaluation metrics improve as possible.We are considering mainly 5 evaluation metric they are precision, recall, F-Score, MAP and nDCG.

3 Motivation

We see daily in our day today needs we use a search engine like google a lot.So a natural question arises suppose we have our own website or collection of a very large document and we want to search for a query . How do we do that . In assignment we were given Carnfield dataset and we made a toy search engine using Vector Space Model, which has its own limitations as stated earlier.So naturally we would like to improve on those limitations and make our search engine more robust by trying out different models and techniques we studied in the class.So we tried a few of them like Latent Semantic Analysis, Explicit Semantic Analysis , Query Expansion using Wordnet etc to overcome the limitations.

3.1 Motivation for Latent Semantic Analysis

It provides information well beyond the lexical level and reveals semantical relations between the entities of interest.

3.2 Motivation for Explicit Semantic Analysis

It is inspired by the desire to augment text representation with massive amounts of world knowledge. We represent texts as a weighted mixture of a predetermined set of natural concepts, which are defined by humans themselves and can be easily explained. To achieve this aim, we use concepts defined by Wikipedia articles, e.g., COMPUTER SCIENCE, INDIA, or LANGUAGE. An important advantage of our approach is thus the use of vast amounts of highly organized human knowledge encoded in Wikipedia. Furthermore, Wikipedia undergoes constant development so its breadth and depth steadily increase over time. Also working with such scientific field is motivating and significant since many researches try to develop information retrieval systems that could handle documents written in specific language.

4 Background and related work

Many researches and studies focus on improving the search mechanisms used in IR systems in order to satisfy the user defined query as most as the system can. The major part of building an Information Retrieval (IR) system is to understand the contents of documents significantly. So "The more the system able to understand the contents of the documents the more effective will be the retrieval outcomes."
We went through different papers namely "Concept-Based Information Retrieval Using Explicit Semantic Analysis OFER EGOZI, SHAUL MARKOVITCH, and EVGENIY GABRILOVICH, Technion—Israel Institute of Technology" ,

papers shared in class to get the idea how to improve the vector space model.

5 Proposed Methodology

5.1 Latent Semantic Analysis

Latent Semantic Analysis is a fully automatic mathematical/statistical technique for extracting and inferring relations of expected contextual usage of words in passages of discourse. It is not a traditional natural language processing or artificial intelligence program; it uses no humanly constructed dictionaries, knowledge bases, semantic networks, grammars, syntactic parsers, or morphologies, or the like, and takes as its input only raw text parsed into words defined as unique character strings and separated into meaningful passages or samples such as sentences or paragraphs.

The first step is to represent the text as a matrix in which each row stands for a unique word and each column stands for a text passage or other context. Each cell contains the frequency with which the word of its row appears in the passage denoted by its column. Next, the cell entries are subjected to a preliminary transformation, whose details we will describe later, in which each cell frequency is weighted by a function that expresses both the word's importance in the particular passage and the degree to which the word type carries information in the domain of discourse in general.

Next, LSA applies singular value decomposition **(SVD)** to the matrix. This is a form of factor analysis, or more properly the mathematical generalization of which factor analysis is a special case. In SVD, a rectangular matrix is decomposed into the product of three other matrices. One component matrix describes the original row entities as vectors of derived orthogonal factor values, another describes the original column entities in the same way, and the third is a diagonal matrix containing scaling values such that when the three components are matrix-multiplied, the original matrix is reconstructed.

The number of dimensions retained in LSA is an empirical issue. Because the underlying principle is that the original data should not be perfectly regenerated but, rather, an optimal dimensionality should be found that will cause correct induction of underlying relations, the customary factor-analytic approach of choosing a dimensionality that most parsimoniously represent the true variance of the original data is not appropriate.

Finally, the measure of similarity computed in the reduced dimensional space is usually, the cosine between vectors. We did not stop only at LSA rather took the linear combination of the cosine similarities obtained by VSM and LSA to develop hybrid model that performs better than the individual model.

5.2 Explicit Semantic Analysis

Explicit Semantic Analysis (ESA), a novel method that represents the meaning of texts in a high-dimensional space of concepts derived from Wikipedia. We fetched documents from wikipedia that are related to the title of the carnfield docs and use it to represent our document and query matrix.

5.3 Query Expansion Using WordNet

As WordNet is a great resource that captures the relation between words like dog is related to animal, it also gives synonyms, antonyms , parts of speech tag of words.So we capture synonymy of each word in the query and expanded our query then found out cosine similarity between expanded query and the documents.

5.4 VSM with BM25 Formula

BM25 is a bag-of-words retrieval function that ranks a set of documents based on the query terms appearing in each document, regardless of their proximity within the document. It is a family of scoring functions with slightly different components and parameters. One of the most prominent instantiations of the function is as follows.

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

where $f(q_i, D)f(q_i, D)$is $q_iq_i's$ term frequency in the document D, $|D||D|$is the length of the document D in words, and avgdl is the average document length in the text collection from which documents are drawn. k_1k_1 and b are free parameters, usually chosen, in absence of an advanced optimization, as $k_1 \in [1.2, 2.0]k_1 \in [1.2, 2.0]$ and $b = 0.75b = 0.75$.[1]$IDF(q_i)IDF(q_i)$ is the IDF (inverse document frequency) weight of the query term q_iq_i. It is usually computed as:

$IDF(q_i) = \ln\left(\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} + 1\right)$ where N is the total number of documents in the collection, and $n(q_i)n(q_i)$ is the number of documents containing q_iq_i.

There are several interpretations for IDF and slight variations on its formula. In the original BM25 derivation, the IDF component is derived from the Binary Independence Model.

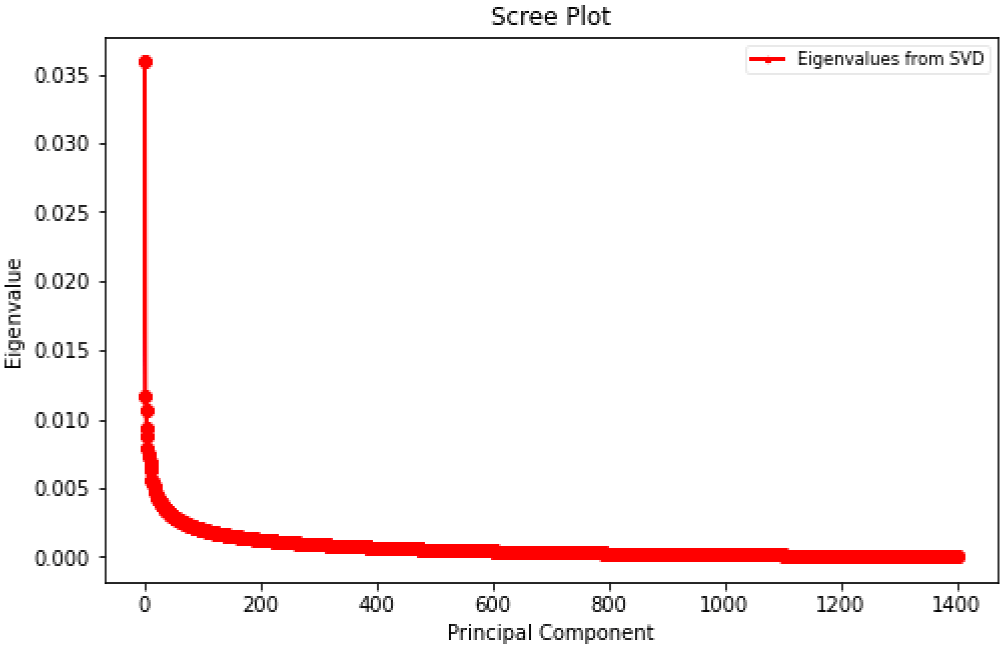
6 Experiments

First we read Queries and Docs from json files and preprocess them. We preprocess the data using punktSentenceSegmenter, pennTreeBankTokenizer, Lematization and stop word removal (english).Then we count the frequency of each word in the document and store them in the dictionary DF_carn. Then we create a vocabulary of all unique words and store them in vocab_carn.Then we go through each word in each doc and craete findout the tf-idf values corresponding to them using create_tf_idf function and then we make a vectorized representation of each document using the function createDocMatrix and store it in D_carn . Similarly we create vectorial representation of the queries using gen_query_vector function and store it in Query_rep. For Vector space model we simply find the cosine similarity between each query and each doc and store them in cosine_smiliratity_values_VSM. Then we get the doc_IDs_ordered_VSM as per their cosine similarity values.Once we have that we go for evaluating the model i.e. calculate different evaluation metrics like precison , recall, Map, nDCG. Once this part is done we go for different models one by one and compare the evaluation metric with that obtained from Vector Space Model.

6.1 Latent Semantic Analysis

Now we do singular value decomposition of the matrix D_carn i.e. Doc_tf_id which is of shape $(1400, 8839)$. We get three matrices namely U, S, VT where U captures Doc x Doc similarity, V captures term x term similarity and S stores the singular values.Before proceeding further we need to find the K value , for that for that we are using a screen plot of Eigen value from SVD vs Principal Component.

Scree plot: In multivariate statistics, a scree plot is a line plot of the eigenvalues of factors or principal components in an analysis. The scree plot is used to determine the number of factors to retain in an exploratory factor analysis (FA) or principal components to keep in a principal component analysis (PCA). The procedure of finding statistically significant factors or components using a scree plot is also known as a scree test.



From the graph it is clear that K=200 captures sufficient information about the tf-idf matrix. Now we find the doc_rep_k matrix and query_rep_k using the formula doc_rep_k $= U[:,: K], diag(S)[: K,: K]$ and query_rep_k $=Query_rep * VT^{-1}[:,: K] * S^{-1}[: K,: K]$
Then we find the cosine similarity and evaluate our LSA model.
Also we use another method to find K with a goal to maximiza nDCG@5 i.e we

plot nDCG@5 value with different vales of K and check when it is maximized and carry out the above calculation for this K value. From here we get K=500

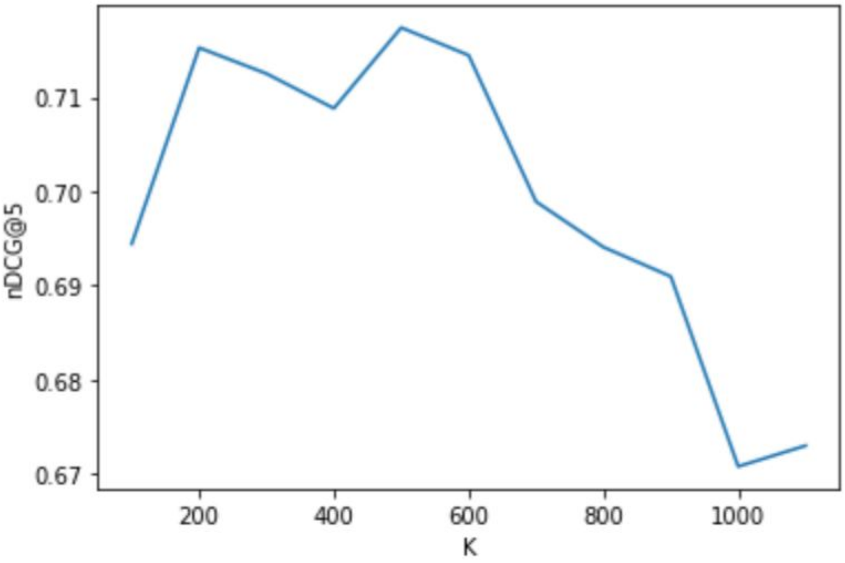


Fig. 1. Finding K for maximizing nDCG@5

As we have both Vector Space Model and LSA ready we compare them through different plots .
Also we go for linear combination of the cosine similarity values obtained from both LSA and VSM to make a hybrid model .
cosine_similarity_values_mix = alpha*cosine_smiliratity_values_VSM + (1-alpha)*cosine_smiliratity_values_LSA
We agin find alpha from the plot between alpha and K .

And again do the comparison of mix model and VSM for K =0.3 obtained from the above graph.

6.2 Explicit Semantic Analysis

The main idea here is to reprsent the documents and query in terms of concepts dervied by mining through wikipedia articles.
The first step remains the same i.e. to create vectorial representation of documents and queries in terms of tf-idf values , dimension of document matrix will be $noofdocXnoofterms$ and that of query matrix will be $noofqueryXnoofterms$.
After that we took we stored title of each doc in the list doc_title and search for wikipedia articles using the title name as a key and stored the summary of top 5

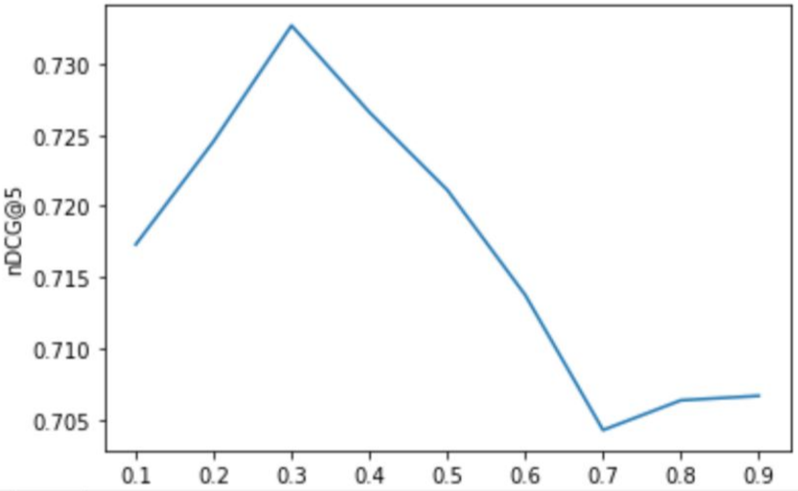


Fig. 2. Finding alpha for maximizing nDCG@5

documents retrieved for each title into the list wikipedia_docs, which takes quite good amount of time.
After that we prepossess the wikipedia_docs create tf-idf matrix for the same.
Next task is to create the ESA_vector for each doc and each query so that similarity between them can be measured. For preparing document esa vector i.e. doc_esa_vector we went through the document one by one , for each document we took one one word ,multiplied the original tf-idf value of the word with the vectorial representation of it in terms of the wikipedia doc that we get from the corresponding columns of the tf-idf of wikipedia matrix, we did it for all the words in the document and added them to get the carnfield doc representation in terms of the wikipedia articles, we did it for all the doc to get doc_esa_vector_matrix. Similarly we found out the query_esa_vector_matrix. Then other steps are simple i.e. find cosine similarities between them and arranging the doc in the decreasing order of cosine similarity between the docs and a particular query.
Then again we go for convex combination of the cosine similarity values obtained from VSM and ESA to arrive at hybrid model. To find the factor of combination we hypertuned alpha with goal to maximize nDCG@5

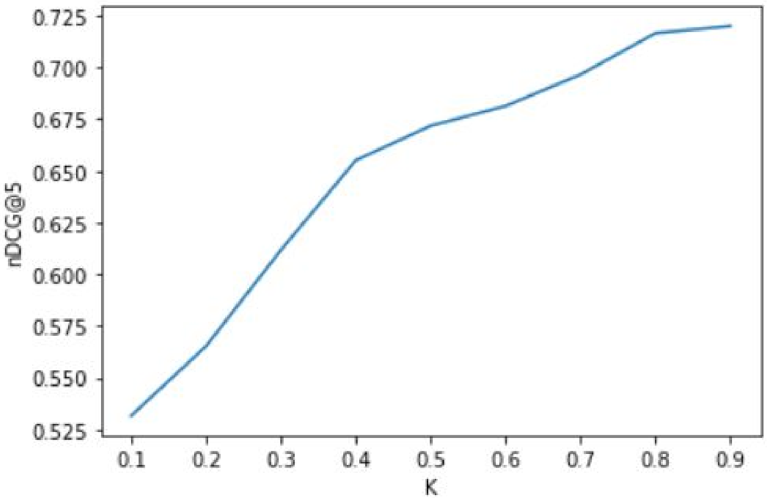


Fig. 3. Finding alpha for maximizing nDCG@5

6.3 Query Expansion Using WordNet

For this model steps till creating tf-idf of doc remains same as we did it for SVM. Once its done. We take each word of a query and find the synonyms of each words and append it to the original query . We do it for all the queries to get the expanded query so that our search engine can handle synonymy problem. Once we have the expanded query we get the vectorial representation of it and proceed as earlier for calculating cosine similarity and then compare the evaluation metrics.
Here also we go for convex combination with VSM as well as we go for SVD along with query expansion with a hope to perform better.

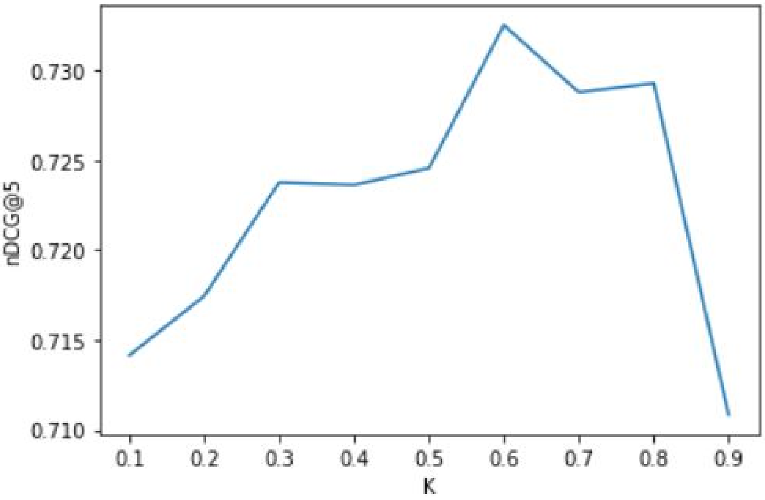


Fig. 4. Finding alpha for maximizing nDCG@5

From the graph we get alpha =0.65
We take linear combination of the VSM and Query_Expansion_with_LSA with alpha being 0.65 and compare the different evaluation metrics.

6.4 VSM with BM25 Formula

We did not go for implementing this formula from scratch rather used the gensim library and got our work done.

7 Results

In the below results we compare vector space model(shaded in yellow) with new model.

7.1 Latent Semantic Analysis

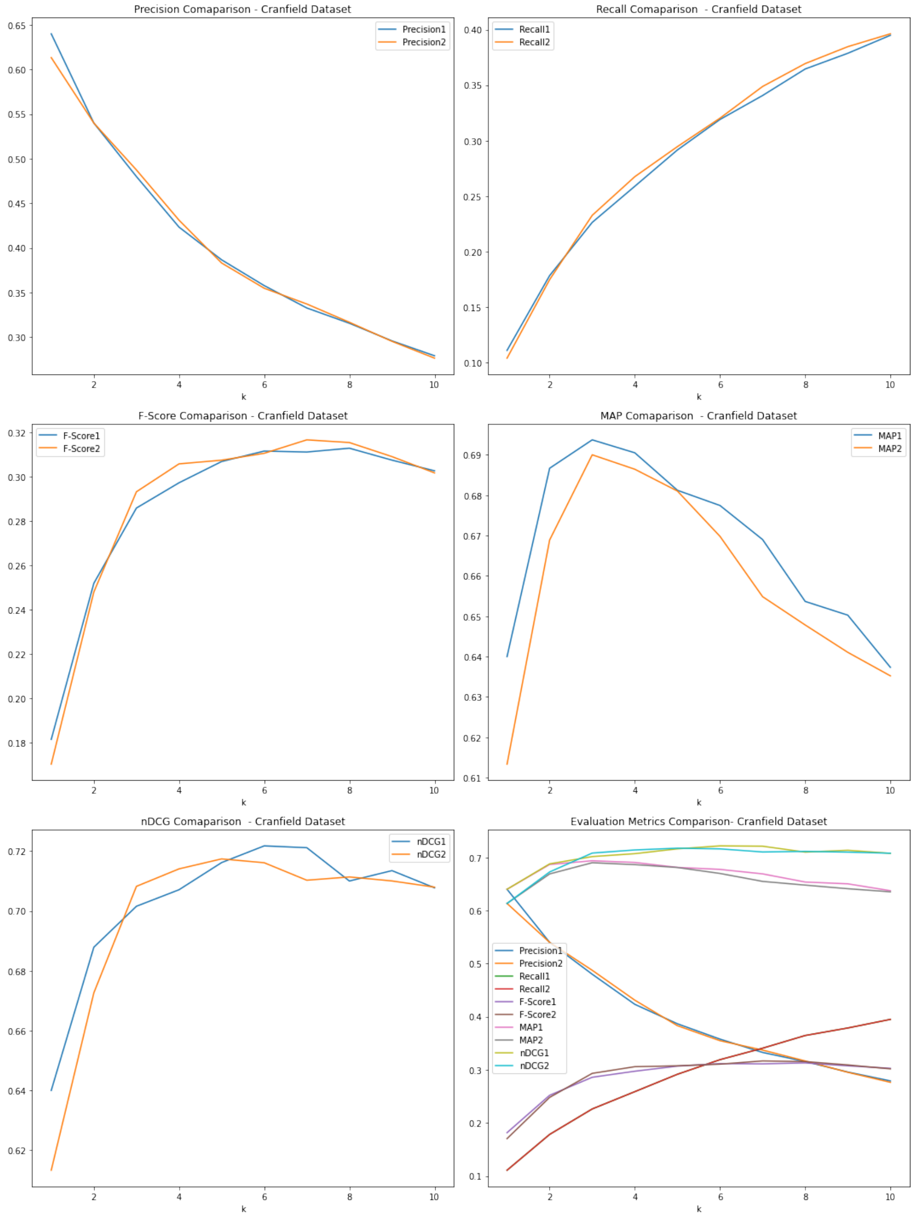
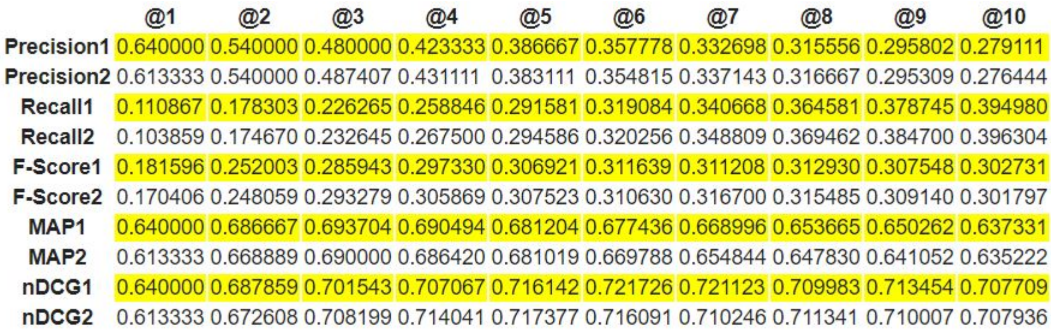


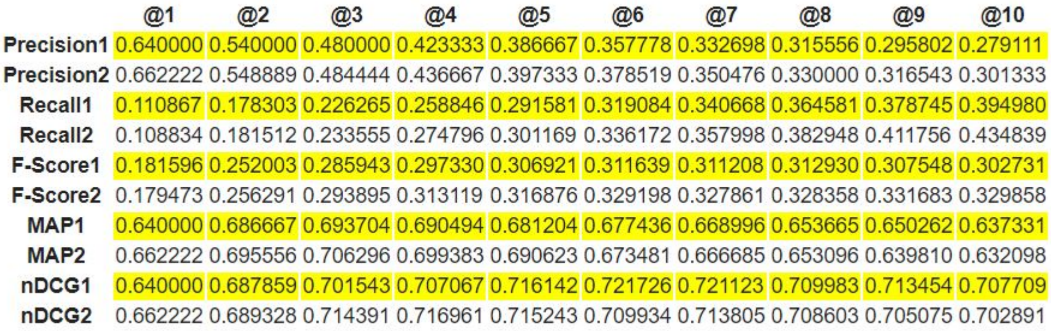
Fig. 5. Comparison between VSM and LSA @ K =500



	@1	@2	@3	@4	@5	@6	@7	@8	@9	@10
Precision1	0.640000	0.540000	0.480000	0.423333	0.386667	0.357778	0.332698	0.315556	0.295802	0.279111
Precision2	0.613333	0.540000	0.487407	0.431111	0.383111	0.354815	0.337143	0.316667	0.295309	0.276444
Recall1	0.110867	0.178303	0.226265	0.258846	0.291581	0.319084	0.340668	0.364581	0.378745	0.394980
Recall2	0.103859	0.174670	0.232645	0.267500	0.294586	0.320256	0.348809	0.369462	0.384700	0.396304
F-Score1	0.181596	0.252003	0.285943	0.297330	0.306921	0.311639	0.311208	0.312930	0.307548	0.302731
F-Score2	0.170406	0.248059	0.293279	0.305869	0.307523	0.310630	0.316700	0.315485	0.309140	0.301797
MAP1	0.640000	0.686667	0.693704	0.690494	0.681204	0.677436	0.668996	0.653665	0.650262	0.637331
MAP2	0.613333	0.668889	0.690000	0.686420	0.681019	0.669788	0.654844	0.647830	0.641052	0.635222
nDCG1	0.640000	0.687859	0.701543	0.707067	0.716142	0.721726	0.721123	0.709983	0.713454	0.707709
nDCG2	0.613333	0.672608	0.708199	0.714041	0.717377	0.716091	0.710246	0.711341	0.710007	0.707936

Fig. 6. Table for evaluation Metric

From the above table and graphs and table it is very clear that LSA Model is performing little better wrt precision, recall and F-Score and bad for MAP and nDCG .
Overall its good.
Now lets see for K =200



	@1	@2	@3	@4	@5	@6	@7	@8	@9	@10
Precision1	0.640000	0.540000	0.480000	0.423333	0.386667	0.357778	0.332698	0.315556	0.295802	0.279111
Precision2	0.662222	0.548889	0.484444	0.436667	0.397333	0.378519	0.350476	0.330000	0.316543	0.301333
Recall1	0.110867	0.178303	0.226265	0.258846	0.291581	0.319084	0.340668	0.364581	0.378745	0.394980
Recall2	0.108834	0.181512	0.233555	0.274796	0.301169	0.336172	0.357998	0.382948	0.411756	0.434839
F-Score1	0.181596	0.252003	0.285943	0.297330	0.306921	0.311639	0.311208	0.312930	0.307548	0.302731
F-Score2	0.179473	0.256291	0.293895	0.313119	0.316876	0.329198	0.327861	0.328358	0.331683	0.329858
MAP1	0.640000	0.686667	0.693704	0.690494	0.681204	0.677436	0.668996	0.653665	0.650262	0.637331
MAP2	0.662222	0.695556	0.706296	0.699383	0.690623	0.673481	0.666685	0.653096	0.639810	0.632098
nDCG1	0.640000	0.687859	0.701543	0.707067	0.716142	0.721726	0.721123	0.709983	0.713454	0.707709
nDCG2	0.662222	0.689328	0.714391	0.716961	0.715243	0.709934	0.713805	0.708603	0.705075	0.702891

Fig. 7. Table for evaluation Metric

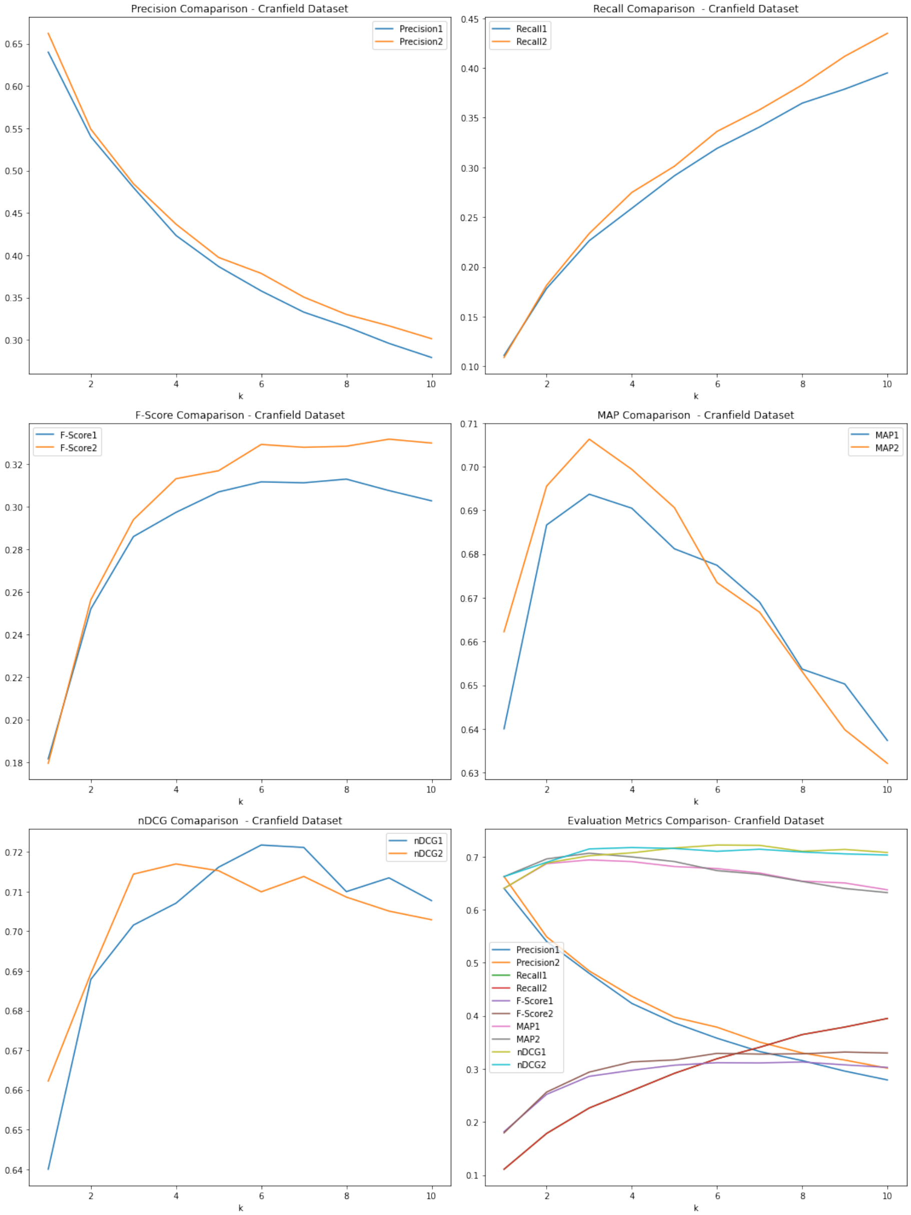
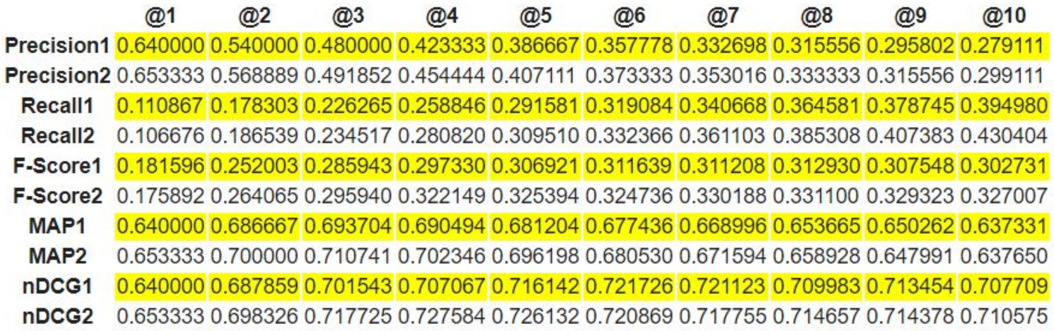


Fig. 8. Comparison between VSM and LSA @ K =200

From the graphs and table it is clear that for K = 200 LSA perform better in terms of precision, recall, F-Score, MAP . And wrt nDCG also it is better than K =500

7.2 Linear Combination of Vector Space Model and Latent Semantic Analysis



	@1	@2	@3	@4	@5	@6	@7	@8	@9	@10
Precision1	0.640000	0.540000	0.480000	0.423333	0.386667	0.357778	0.332698	0.315556	0.295802	0.279111
Precision2	0.653333	0.568889	0.491852	0.454444	0.407111	0.373333	0.353016	0.333333	0.315556	0.299111
Recall1	0.110867	0.178303	0.226265	0.258846	0.291581	0.319084	0.340668	0.364581	0.378745	0.394980
Recall2	0.106676	0.186539	0.234517	0.280820	0.309510	0.332366	0.361103	0.385308	0.407383	0.430404
F-Score1	0.181596	0.252003	0.285943	0.297330	0.306921	0.311639	0.311208	0.312930	0.307548	0.302731
F-Score2	0.175892	0.264065	0.295940	0.322149	0.325394	0.324736	0.330188	0.331100	0.329323	0.327007
MAP1	0.640000	0.686667	0.693704	0.690494	0.681204	0.677436	0.668996	0.653665	0.650262	0.637331
MAP2	0.653333	0.700000	0.710741	0.702346	0.696198	0.680530	0.671594	0.658928	0.647991	0.637650
nDCG1	0.640000	0.687859	0.701543	0.707067	0.716142	0.721726	0.721123	0.709983	0.713454	0.707709
nDCG2	0.653333	0.698326	0.717725	0.727584	0.726132	0.720869	0.717755	0.714657	0.714378	0.710575

Fig. 9. Comparison between VSM(model1) and Hybrid model(model2)

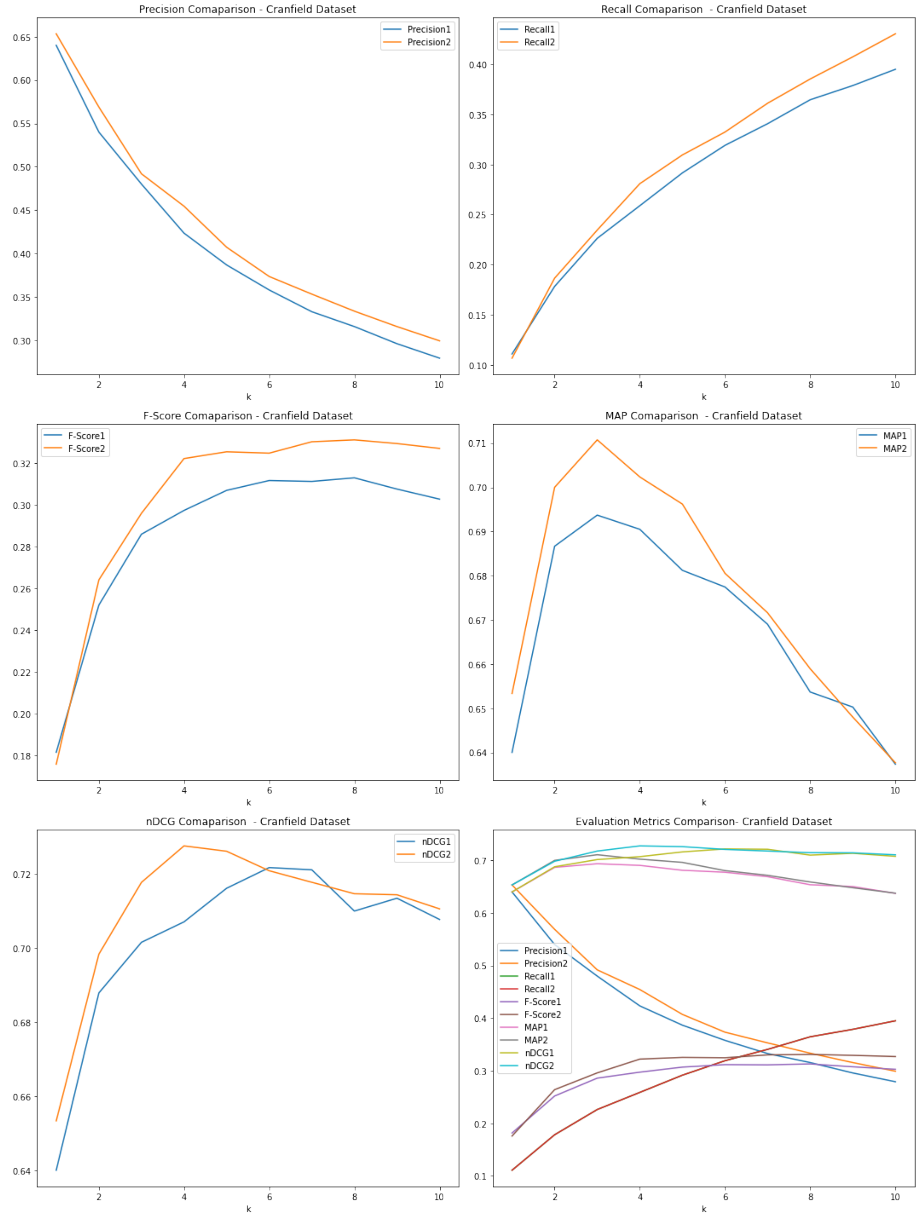


Fig. 10. Comparison between Hybrid Model and VSM

From the above table and graphs and table it is very clear that Hybrid Model is performing better wrt all the evaluation metrics as it has both the advantage of LSA and VSM.Linear combination of the cosine similarties is taken at alpha =0.3

7.3 Explicit Semantic Analysis

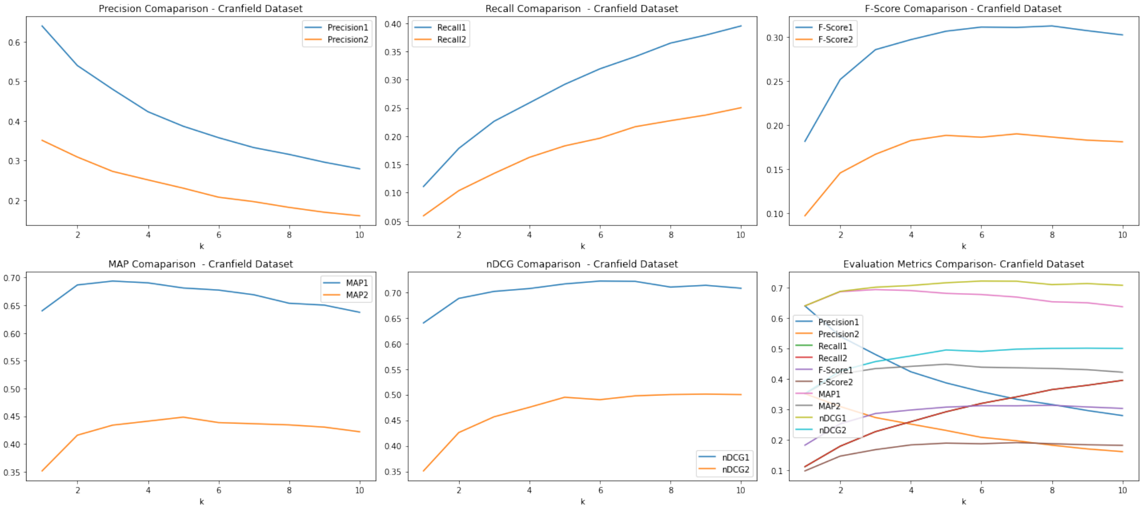
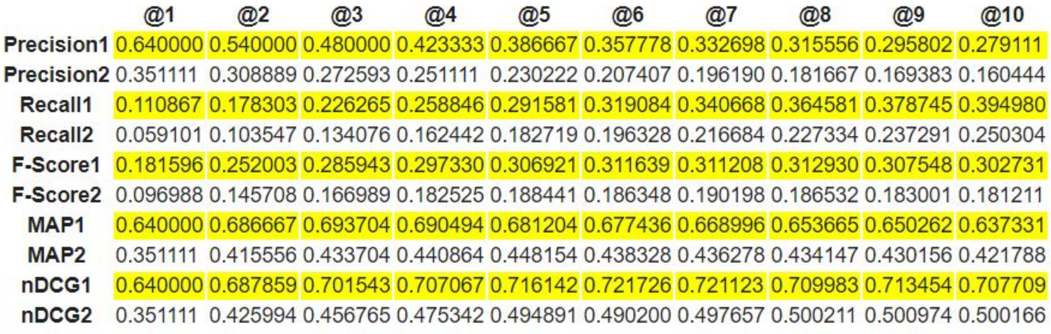


Fig. 11. Comparison between ESA Model and VSM



	@1	@2	@3	@4	@5	@6	@7	@8	@9	@10
Precision1	0.640000	0.540000	0.480000	0.423333	0.386667	0.357778	0.332698	0.315556	0.295802	0.279111
Precision2	0.351111	0.308889	0.272593	0.251111	0.230222	0.207407	0.196190	0.181667	0.169383	0.160444
Recall1	0.110867	0.178303	0.226265	0.258846	0.291581	0.319084	0.340668	0.364581	0.378745	0.394980
Recall2	0.059101	0.103547	0.134076	0.162442	0.182719	0.196328	0.216684	0.227334	0.237291	0.250304
F-Score1	0.181596	0.252003	0.285943	0.297330	0.306921	0.311639	0.311208	0.312930	0.307548	0.302731
F-Score2	0.096988	0.145708	0.166989	0.182525	0.188441	0.186348	0.190198	0.186532	0.183001	0.181211
MAP1	0.640000	0.686667	0.693704	0.690494	0.681204	0.677436	0.668996	0.653665	0.650262	0.637331
MAP2	0.351111	0.415556	0.433704	0.440864	0.448154	0.438328	0.436278	0.434147	0.430156	0.421788
nDCG1	0.640000	0.687859	0.701543	0.707067	0.716142	0.721726	0.721123	0.709983	0.713454	0.707709
nDCG2	0.351111	0.425994	0.456765	0.475342	0.494891	0.490200	0.497657	0.500211	0.500974	0.500166

Fig. 12. Table of Comparison between ESA Model and VSM

From the above graph and table it is clear that ESA very bad with respect to all evealuation parameter considered here. This is because Cranfield is a relatively smaller dataset , so the doc collected from wikipedia is not of much relevance to the doc present in the dataset, so the concepts are not captured properly.

7.4 Hybrid Model of ESA and VSM at alpha =0.9

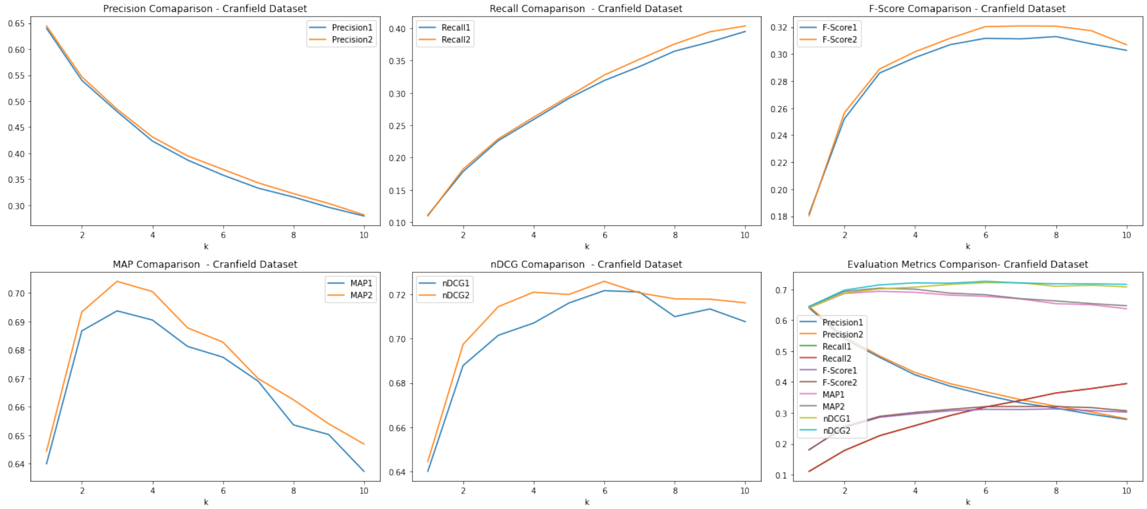
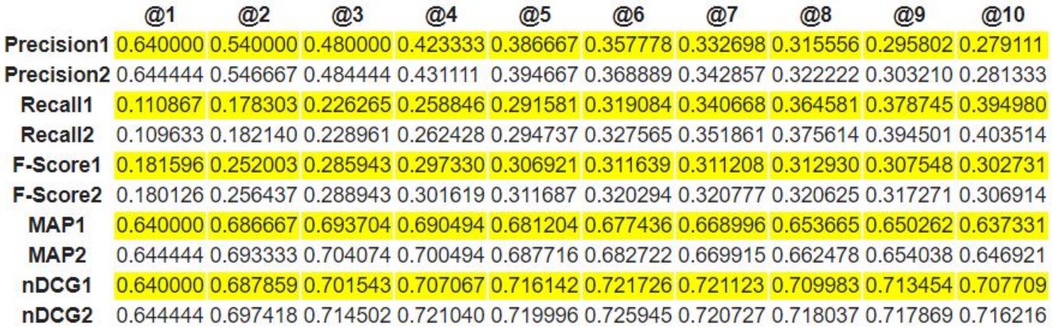


Fig. 13. Comparison between ESA-VSM Hybrid Model and VSM



	@1	@2	@3	@4	@5	@6	@7	@8	@9	@10
Precision1	0.640000	0.540000	0.480000	0.423333	0.386667	0.357778	0.332698	0.315556	0.295802	0.279111
Precision2	0.644444	0.546667	0.484444	0.431111	0.394667	0.368889	0.342857	0.322222	0.303210	0.281333
Recall1	0.110867	0.178303	0.226265	0.258846	0.291581	0.319084	0.340668	0.364581	0.378745	0.394980
Recall2	0.109633	0.182140	0.228961	0.262428	0.294737	0.327565	0.351861	0.375614	0.394501	0.403514
F-Score1	0.181596	0.252003	0.285943	0.297330	0.306921	0.311639	0.311208	0.312930	0.307548	0.302731
F-Score2	0.180126	0.256437	0.288943	0.301619	0.311687	0.320294	0.320777	0.320625	0.317271	0.306914
MAP1	0.640000	0.686667	0.693704	0.690494	0.681204	0.677436	0.668996	0.653665	0.650262	0.637331
MAP2	0.644444	0.693333	0.704074	0.700494	0.687716	0.682722	0.669915	0.662478	0.654038	0.646921
nDCG1	0.640000	0.687859	0.701543	0.707067	0.716142	0.721726	0.721123	0.709983	0.713454	0.707709
nDCG2	0.644444	0.697418	0.714502	0.721040	0.719996	0.725945	0.720727	0.718037	0.717869	0.716216

Fig. 14. Table of Comparison between ESA-VSM Hybrid Model and VSM

From the above table and graphs and table it is very clear that ESA-VSM Hybrid Model is performing better wrt all the evaluation metrics as it has both the advantage of ESA and VSM.Linear combination of the cosine similarties is taken at alpha =0.9

7.5 Query Exapansion Using WordNet

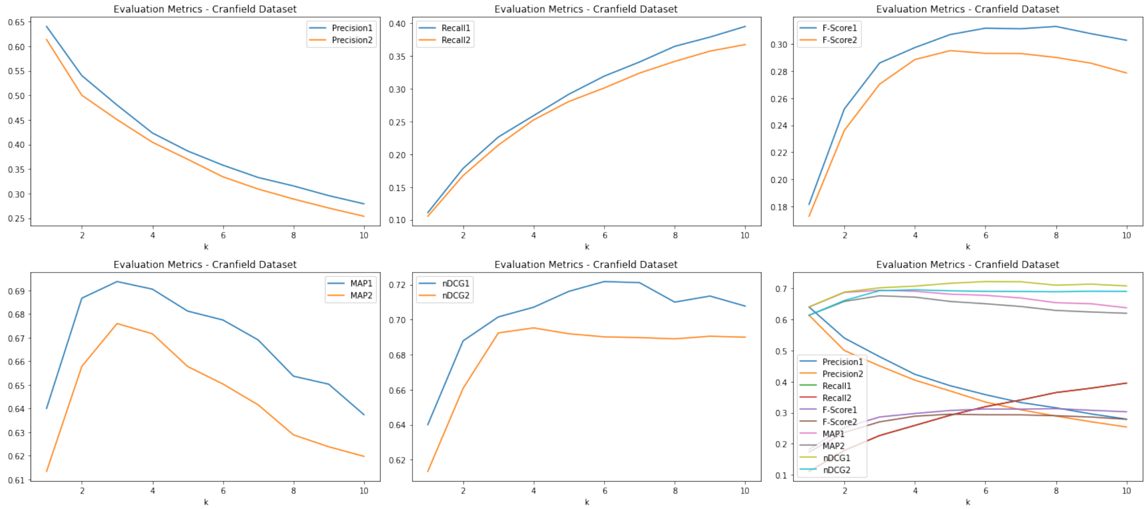
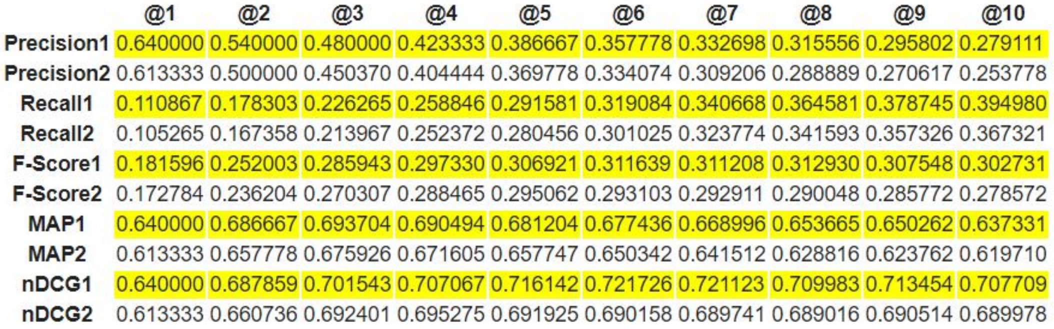


Fig. 15. Comparison between VSM with Query Expansion and VSM



	@1	@2	@3	@4	@5	@6	@7	@8	@9	@10
Precision1	0.640000	0.540000	0.480000	0.423333	0.386667	0.357778	0.332698	0.315556	0.295802	0.279111
Precision2	0.613333	0.500000	0.450370	0.404444	0.369778	0.334074	0.309206	0.288889	0.270617	0.253778
Recall1	0.110867	0.178303	0.226265	0.258846	0.291581	0.319084	0.340668	0.364581	0.378745	0.394980
Recall2	0.105265	0.167358	0.213967	0.252372	0.280456	0.301025	0.323774	0.341593	0.357326	0.367321
F-Score1	0.181596	0.252003	0.285943	0.297330	0.306921	0.311639	0.311208	0.312930	0.307548	0.302731
F-Score2	0.172784	0.236204	0.270307	0.288465	0.295062	0.293103	0.292911	0.290048	0.285772	0.278572
MAP1	0.640000	0.686667	0.693704	0.690494	0.681204	0.677436	0.668996	0.653665	0.650262	0.637331
MAP2	0.613333	0.657778	0.675926	0.671605	0.657747	0.650342	0.641512	0.628816	0.623762	0.619710
nDCG1	0.640000	0.687859	0.701543	0.707067	0.716142	0.721726	0.721123	0.709983	0.713454	0.707709
nDCG2	0.613333	0.660736	0.692401	0.695275	0.691925	0.690158	0.689741	0.689016	0.690514	0.689978

Fig. 16. Table of Comparison between VSM with Query Expansion and VSM

From the above graph and table it is clear that VSM with Query Expansion is performing littlebad with respect to all evaluation parameter considered here. This is because number of queries in the carnfield data set is quite less , so there is not so much problem of synonymy , like if somewhere the word ”plane” is used then there are very less places where ”jet” is used to refer plane.However for custom query Query Expansion will perform generally better.

7.6 Query Exapansion Using WordNet along with LSA

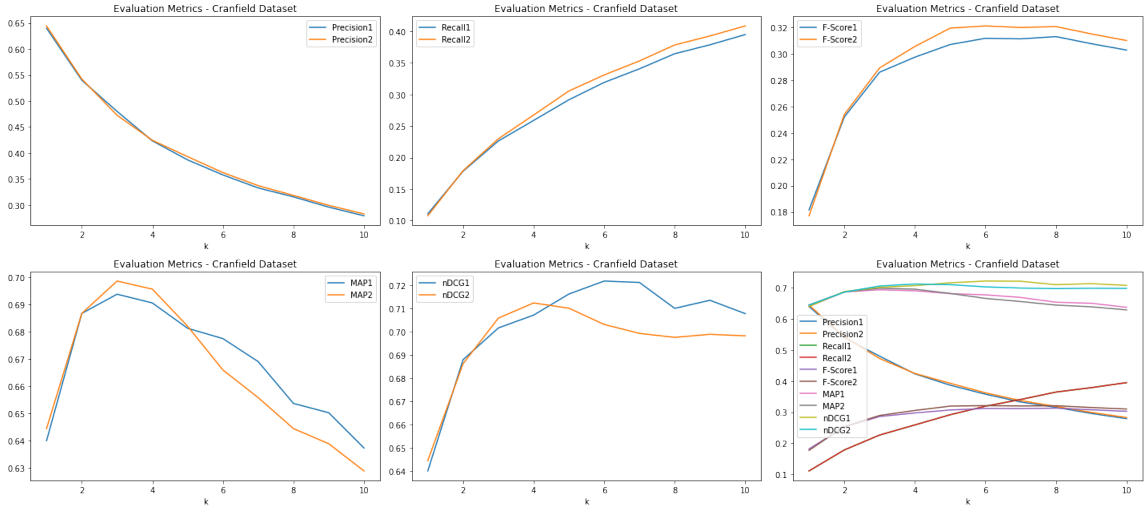
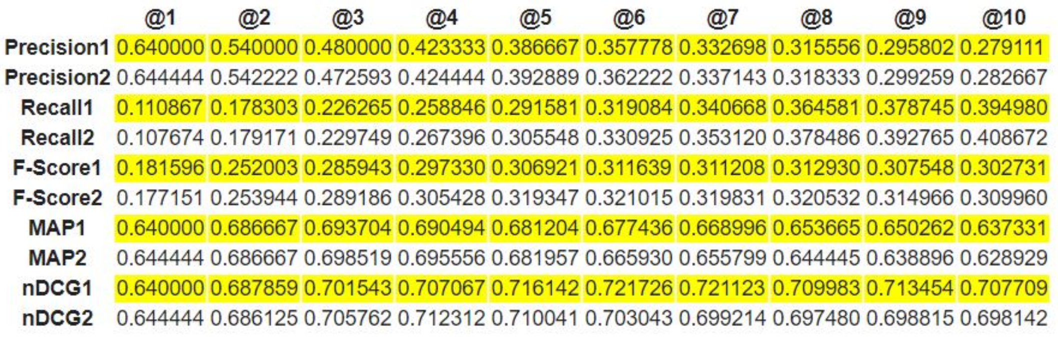


Fig. 17. Comparison between Query Expansion with LSA and VSM



	@1	@2	@3	@4	@5	@6	@7	@8	@9	@10
Precision1	0.640000	0.540000	0.480000	0.423333	0.386667	0.357778	0.332698	0.315556	0.295802	0.279111
Precision2	0.644444	0.542222	0.472593	0.424444	0.392889	0.362222	0.337143	0.318333	0.299259	0.282667
Recall1	0.110867	0.178303	0.226265	0.258846	0.291581	0.319084	0.340668	0.364581	0.378745	0.394980
Recall2	0.107674	0.179171	0.229749	0.267396	0.305548	0.330925	0.353120	0.378486	0.392765	0.408672
F-Score1	0.181596	0.252003	0.285943	0.297330	0.306921	0.311639	0.311208	0.312930	0.307548	0.302731
F-Score2	0.177151	0.253944	0.289186	0.305428	0.319347	0.321015	0.319831	0.320532	0.314966	0.309960
MAP1	0.640000	0.686667	0.693704	0.690494	0.681204	0.677436	0.668996	0.653665	0.650262	0.637331
MAP2	0.644444	0.686667	0.698519	0.695556	0.681957	0.665930	0.655799	0.644445	0.638896	0.628929
nDCG1	0.640000	0.687859	0.701543	0.707067	0.716142	0.721726	0.721123	0.709983	0.713454	0.707709
nDCG2	0.644444	0.686125	0.705762	0.712312	0.710041	0.703043	0.699214	0.697480	0.698815	0.698142

Fig. 18. Table of Comparison between Query Expansion with LSA and VSM

Clearly it is performing better than LSA for precision, recall,and FScore and little poor for other evalaution metric.Here i used K =200 for LSA which can also be fine tuned to make the model perform more better.

7.7 Convex combination of Query expansion with LSA and VSM

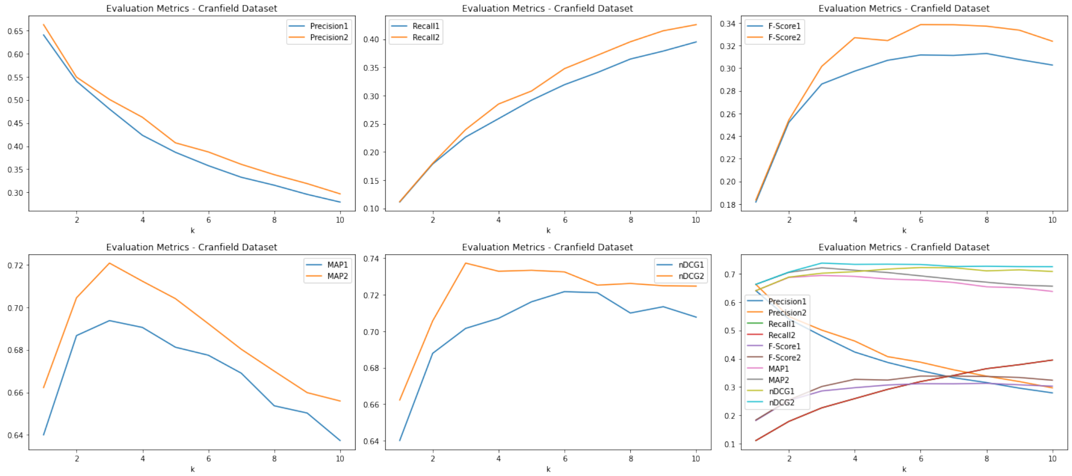
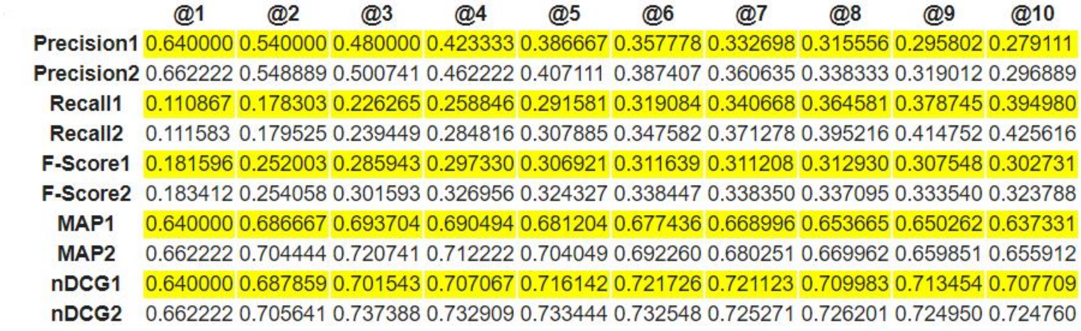


Fig. 19. Comparison Convex combination of Query expansion with LSA and VSM



	@1	@2	@3	@4	@5	@6	@7	@8	@9	@10
Precision1	0.640000	0.540000	0.480000	0.423333	0.386667	0.357778	0.332698	0.315556	0.295802	0.279111
Precision2	0.662222	0.548889	0.500741	0.462222	0.407111	0.387407	0.360635	0.338333	0.319012	0.296889
Recall1	0.110867	0.178303	0.226265	0.258846	0.291581	0.319084	0.340668	0.364581	0.378745	0.394980
Recall2	0.111583	0.179525	0.239449	0.284816	0.307885	0.347582	0.371278	0.395216	0.414752	0.425616
F-Score1	0.181596	0.252003	0.285943	0.297330	0.306921	0.311639	0.311208	0.312930	0.307548	0.302731
F-Score2	0.183412	0.254058	0.301593	0.326956	0.324327	0.338447	0.338350	0.337095	0.333540	0.323788
MAP1	0.640000	0.686667	0.693704	0.690494	0.681204	0.677436	0.668996	0.653665	0.650262	0.637331
MAP2	0.662222	0.704444	0.720741	0.712222	0.704049	0.692260	0.680251	0.669962	0.659851	0.655912
nDCG1	0.640000	0.687859	0.701543	0.707067	0.716142	0.721726	0.721123	0.709983	0.713454	0.707709
nDCG2	0.662222	0.705641	0.737388	0.732909	0.733444	0.732548	0.725271	0.726201	0.724950	0.724760

Fig. 20. Table of Convex combination of Query expansion with LSA and VSM

Clearly it is performing much much better than VSM wrt all the evaluation metric ,It is perfroming better than all other model we tried on carnfield data set.Even the value of K and corresponding alpha can be further hypertuned to get more better performance.

7.8 VSM with BM25 Formula

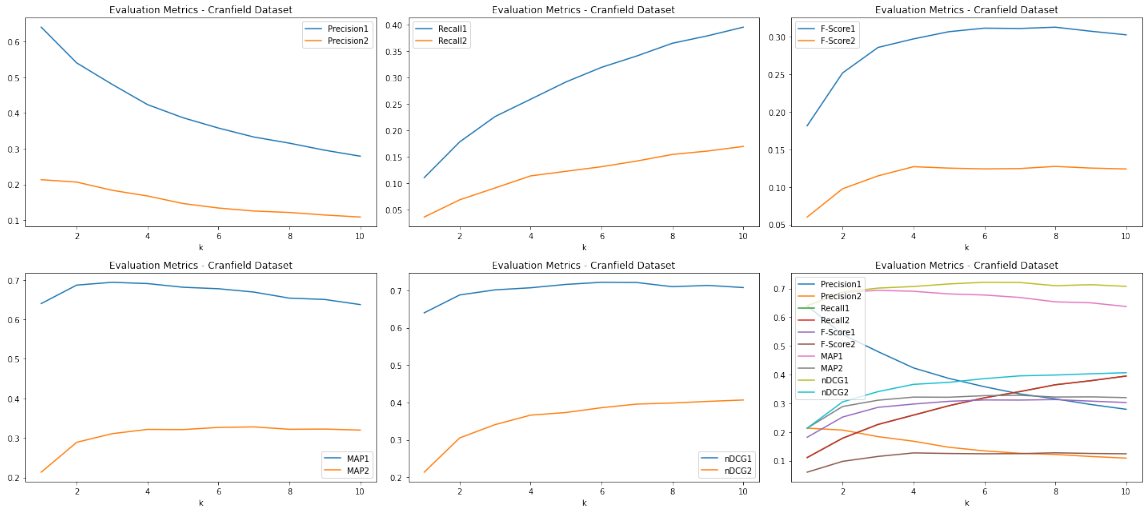
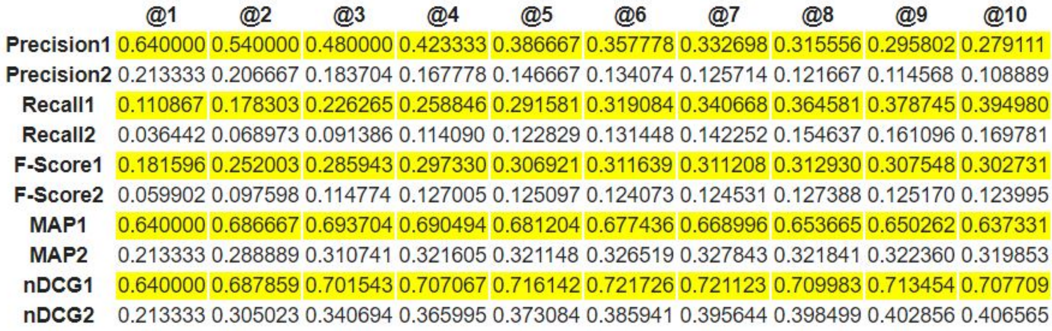


Fig. 21. Comparison VSM with BM25 Formula and VSM



	@1	@2	@3	@4	@5	@6	@7	@8	@9	@10
Precision1	0.640000	0.540000	0.480000	0.423333	0.386667	0.357778	0.332698	0.315556	0.295802	0.279111
Precision2	0.213333	0.206667	0.183704	0.167778	0.146667	0.134074	0.125714	0.121667	0.114568	0.108889
Recall1	0.110867	0.178303	0.226265	0.258846	0.291581	0.319084	0.340668	0.364581	0.378745	0.394980
Recall2	0.036442	0.068973	0.091386	0.114090	0.122829	0.131448	0.142252	0.154637	0.161096	0.169781
F-Score1	0.181596	0.252003	0.285943	0.297330	0.306921	0.311639	0.311208	0.312930	0.307548	0.302731
F-Score2	0.059902	0.097598	0.114774	0.127005	0.125097	0.124073	0.124531	0.127388	0.125170	0.123995
MAP1	0.640000	0.686667	0.693704	0.690494	0.681204	0.677436	0.668996	0.653665	0.650262	0.637331
MAP2	0.213333	0.288889	0.310741	0.321605	0.321148	0.326519	0.327843	0.321841	0.322360	0.319853
nDCG1	0.640000	0.687859	0.701543	0.707067	0.716142	0.721726	0.721123	0.709983	0.713454	0.707709
nDCG2	0.213333	0.305023	0.340694	0.365995	0.373084	0.385941	0.395644	0.398499	0.402856	0.406565

Fig. 22. Table of comparison between VSM with BM25 Formula and VSM

Clearly it is performing worse than all other model i come accross so for.

8 Conclusion

We tried almost more than 25 combination of different models with different hyperparameter. We shortslisted 8 among them and after a detailed comparison is done for those model we arrive at the following conclusion.
(1)Hybrid Model that combination of VSM along with some other model like LSA/ESA or Query Expansion is performing better than the individual models on carnfield dataset.
(2)Query expansion with WordNet with LSA applied to it outperformed all other models . When compared with Vector Space Model Precision@10 increased by 6.36% recall@10 increased by 7.75% F-Score@10 increased by 6.95% MAP@10 increased by 2.91% and nDCG@10 increased by 2.41%.
(3)For small dataset like carnfield ESA alone is not wise to perform as it takes quite good amount of time to fetch the articles from wikipedia without any improvement in the performance.
(4)Cosine similarity measure performed well than BM25 similarity measure on Vector Space Model for Carfield Data set .
(5) Query Expansion is more useful when we apply LSA because then only it is able to capture synonyms and hidden relation in a much better way.