CS6370 Team 6 Project report

Durga Sandeep [ME16B125] and Sureddy Abhishek [ME16B166]

IIT Madras
me16b166@smail.iitm.ac.in
me16b125@smail.iitm.ac.in

Abstract. Goal of this project is to build an efficient information retrieval system using Natural Language Processing. The report starts with simple vector space model (VSM) which is considered as the baseline model. We had built two baseline models where the different pre-processing techniques were used. We addressed some of the limitations of vector space model (VSM) through different techniques like Latent Semantic Analysis (LSA), Explicit Semantic Analysis (ESA). Later on this, we also considered query expansion using similar words of the query. We briefly discussed the improvements of the different models and compared with the baseline model using statistical methods. Further on, we discussed about some methods, that can significantly reduce search time like Clustering, LDA(Topic modeling). We compared with without clustering retrieval time using two sample t-test. Finally, tried a method which can improve user experience in the real-time querying like : Query auto-completion (text generation using LSTMs).

Keywords: LDA · Clustering · Query auto-completion · LSA · Query expansion · baseline model · ESA.

1 Introduction

* **Vector Space Model (VSM)**
  Vector Space Model (VSM) represents documents and queries as vectors of numbers. In a basic model, those are the product of term frequencies (word count) and inverse document frequency in the corpus. At the time of retrieval, the documents are ranked by the cosine of the angle between the document vectors and the query vector.
* **Latent Semantic Analysis (LSA)**
  Latent semantic analysis (LSA) is a technique in NLP, 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. This approach takes advantage of implicit higher-order structure in the association of terms with documents. This method is expected to solve the synonymy and polysemy upto an extent.
* **Query Expansion Method**
  Query Expansion Method broadens the query by introducing additional tokens or phrases. The search engine automatically rewrites the query to include them. For example, the query vp marketing becomes vp vice president marketing.
* **Clustering of Documents**
  Here, we cluster the documents using K-means and topic-modeling (LDA). Whenever a query is given, we find the similarity of the query with all cluster centres and retrieve the documents only from that cluster which has high similarity. This way we reduce the number of documents to search for a given query, there by decreases the documents retrieval time.
* **Query auto-completion**
  We use an LSTM based text-generation algorithm to generate the next ‘k‘ context words. Whenever we pass an incomplete query, the algorithm predicts the next k appropriate words based on the training data.

2 Vector Space Model

A vector space model is created using the **T**F-IDF scores of words in the documents. We have implemented two versions of the VSM based on the different pre-processing technqiues applied,

1. **VSM-1** Vector Space Model Version 1
2. **VSM-2** Vector Space Model Version 2

The detailed procedures for each version are provided in the below subsections.

2.1 VSM-1

In this model, we adopted the pre-processing techniques from the NLP assignments given. Following are the steps involved to build the model

- **Data Pre-processing:**
  - **Tokenization :** Making each document into a list of sentences. Making each sentence into a list of words. Here we used "Treebank tokenizer" from nltk package.
  - **Stopword removal :** Stopwords (i.e. most common words that occur in almost all documents, like a,an,the and so on) are removed from this sub list of words. Note tow things here, one is that we can define our own stopwords list based on our corpus and second is that even if we skip this step, the weights corresponding to those stopwords will be reduced using tf-idf method. We do remove the stopwords to avoid the large dimensions (i.e vocabulary size) in VSM model.
  - **Inflection reduction :** Here we used **lemmatization** to reduce the all words to their corresponding root words. we didn't use stemming because, it's a crude way of extracting root word. (benefits of lemmatization over stemming are mentioned here) [1])
  - **TF-IDF matrix :** Its product of term frequency(TF) and inverse document frequency (IDF). A tf-idf matrix is constructed and then each column is normalized to unit length for the documents.

- **Ranking and Evaluation:**
  - **Ranking :** Similar pre-processing steps are to be applied on the queries and the query vectors are obtained. Similarity between the documents and the query are established by cosine similarity. Then we rank based on output similarity scores. Large value of cosine means vectors are in similar direction, small values of cosine means vectos are not in similar direction.
  - **Evaluation :** The ranked documents are evaluated based on different metrics like MeanPrecision@k, MeanRecall@k, MAP@k, nDCG@k, Mean F-Score@k.

2.2 VSM-2

In this model, there are few changes in the data pre-processing techniques of the VSM-1. This subsection provides the methodology adopted and we will also discuss the improvements of VSM-2 over VSM-1. We can clearly see there is an improvement in all the metrics for VSM-2 when compared to VSM-1.

- **Data Pre-processing:**
  - **Convert Upper to Lowercase :** In the given corpus, there are no uppercase letters but the given query can have uppercase letters. So in general, we are making all letters to lowercase. This helps when the dataset has uppercase also. For example - Aeroplane, aeroplane both are same words but if you do not convert them to lowercase, then the VSM model treats both words differently which in turn increase our dimensionality of vector("Curse of dimensionality").
  - **Removing numbers in the text :** Usually we might have words attached with numbers as well. So in this case we are removing all the numbers in the corpus
  - **Removing punctuations, accent marks and other diacritics, extra white spaces :** This is very important. If you do not remove it in the corpus, then the VSM model considers the punctuation as one more word and adds to the dimension. For example - if i have a sentence ["Hi, I am a Robot."] After tokenization it gives ['hi', ',', 'i','am','a','robot','.']. This can lead to a very poor model as it increases the vocabulary size i.e dimension of each vector.
  - **Further Pre-processing :** Same tokenizer and stopwords list as VSM-1, We used lemmatization over stemming like in VSM-1. TF-IDF matrix is constructed for the documents and query. These four pre-processing steps are same as VSM-1.

- **Ranking and Evaluation:**
  - **Ranking :** Similar pre-processing steps are to be applied on the queries and the query vectors are obtained. Similarity between the documents and the query are established by cosine similarity. Then we rank based on output similarity scores.
  - **Evaluation :** The ranked documents are evaluated based on different metrics like MeanPrecision@k, MeanRecall@k, MAP@k, nDCG@k, Mean F-Score@k.

2.3 Results and Evaluation

The comparison table for both the models i.e VSM-1 and VSM-2 is shown in table 1

	VSM-1					VSM-2				
k	**MAP**	**Recall**	**f-score**	**n-DCG**	**Precision**	**MAP**	**Recall**	**f-score**	**n-DCG**	**Precision**
1	0.636	0.11	0.181	0.493	0.636	0.693	0.12	0.193	0.53	0.693
2	0.695	0.18	0.26	0.40	0.538	0.733	0.188	0.266	0.412	0.567
3	0.699	0.23	0.291	0.372	0.467	0.734	0.241	0.302	0.390	0.501
4	0.695	0.264	0.303	0.364	0.411	0.732	0.287	0.328	0.390	0.461
5	0.685	0.294	0.309	0.364	0.372	0.721	0.313	0.328	0.391	0.412
6	0.677	0..323	0.314	0.372	0.343	0.710	0.342	0.331	0.396	0.379
7	0.668	0.347	0.315	0.376	0.320	0.697	0.368	0.334	0.402	0.352
8	0.658	0.364	0.311	0.383	0.301	0.687	0.389	0.330	0.407	0.332
9	0.653	0.382	0.310	0.388	0.285	0.680	0.406	0.325	0.411	0.311
10	0.644	0.40	0.304	0.393	0.267	0.674	0.419	0.318	0.414	0.291

Table 1: Results and Evaluation

From table 1, we can clearly see that Version-2, outperforms Version-1 in all aspects (by 3% to 5%). The vocabulary in Version-1 was around 8000 words, where as in Version-2, it was just 5000 words. This is because of proper text pre-processing which was incorporated in Version-2. We have noticed some punctuation left, without being removed in VSM-1, which results in poor similarity scores, thereby reducing the model performance. So we try to do pre-processing relative to the corpus given.
The distribution statistics of precision, recall, fscore, ndcg of various queries @$k = 7$ for VSM-1 Vs VSM-2 is shown in figures 1, 2, 3 and 4.

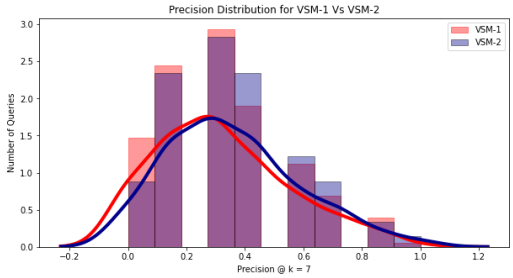


Fig. 1: precision distribution VSM-1 Vs VSM-2 @ k=7

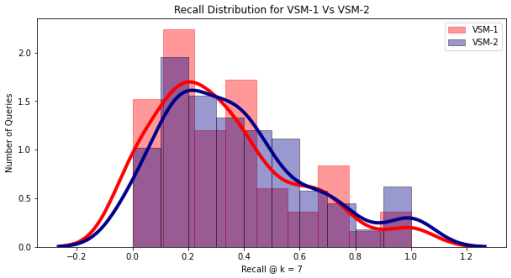


Fig. 2: Recall distribution VSM-1 Vs VSM-2 @ k=7

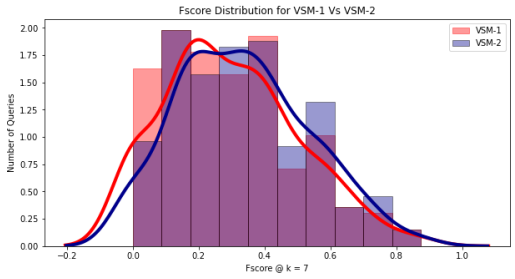


Fig. 3: FScore distribution VSM-1 Vs VSM-2 @ k=7

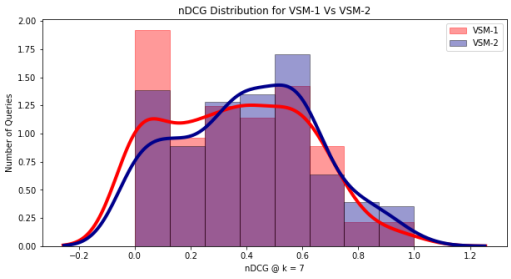
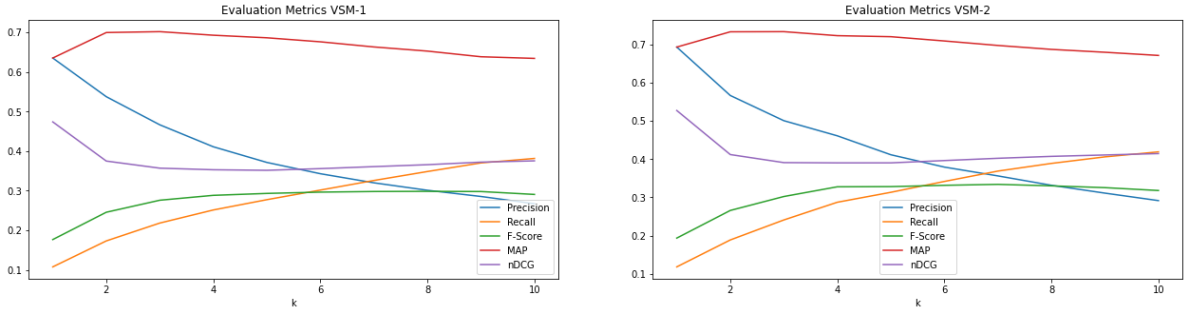


Fig. 4: nDCG distribution VSM-1 Vs VSM-2 @ k=7

Inference : From the above distributions for VSM-1 and VSM-2, we can clearly see that all lower values of metrics (i.e precision, recall, fscore, ndcg) are increased to higher values by the VSM-2 model. So we can say that VSM-2 model performs better than VSM-1 model. Later on, we will also use hypothesis testing to verify our statement i.e VSM-2 and VSM-1 are not performing similarly.

The general trends of the evaluation measures for all models, were as shown in figure 5



(a) VSM-1 (b) VSM-2

Fig. 5: Results and Evaluation

From now onwards, we use VSM-2 as our Baseline model throughout this report.

Few Examples where the retrieved documents are completely irrelevant.

* **query id :** 28
* **query :** "what application has the linear theory design of curved wings ."
* **documents retrieved :** [752, 1075, 762, 674, 921, 1051, 247, 451, 1050, 680]
  Doc 752: "slender not-so-thin wing theory . a method for making an approximate thickness correction to slender thin-wing theory is presented . the method is tested by applying it to cones with rhombic cross-sections and the agreement is found to be good if the cones are not too thick . it is then suggested that the thickness correction to slender thin-wing theory may be applied unchanged to linear thin-wing theory . this suggestion is compared with some experiments on delta wings and it is found that there is considerable improvement over thin-wing theory near the centre line, but that this improvement is not maintained as the wing tips are approached .",
* **relevant documents :** [224, 279, 512]
  Doc 224: "quasi-cylindrical surfaces with prescribed loadings in the linearised theory of supersonic flow . a formula for the velocity field in terms of a given surface distribution of vorticity is applied to points lying on the surface . an equation giving the shape of a quasi circular-cylindrical surface in terms of a prescribed loading is derived . as an example a half ring wing with prescribed loading is discussed ."
  Doc 224: "supersonic drag calculations for a cylindrical shell wing of semicircular cross section combined with a central body of revolution . a semi-circular ring wing with a body of revolution on the axis is studied to find the wave and the vortex drag for various chordwise lift distributions and for three values of a parameter describing the wing geometry . using the wave drag obtained from the chordwise loading that gives the least drag, together with the vortex and skin friction drags, the maximum lift to drag ratio for each wing geometry is computed . compared to the estimates made by lomax and heaslet, somewhat lower drags are found ."
* **Explanation:** The retrieved document by the VSM-2 model has more number of exact matching words to the given query. But for the relevant document, we see there are very few exactly matching words, this is the reason why our VSM-2 model could not retrieve the relevant document.

Few Examples where the retrieved documents are all relevant .

* **query id :** 185
* **query :** "experimental studies on panel flutter ."
* **documents retrieved :** [856, 1008, 766, 857, 859, 858, 391, 948, 658, 864]
* **relevant documents :** [858, 859, 857, 1008, 856, 15, 285, 894, 766, 948]
  Doc 856 : " experimental investigation at mach numbers 3. 0 of the effects of thermal stress and buckling on the flutter of four-bay aluminium alloy panels with length-width ratios of 10 . skin-stiffener aluminum alloy panels consisting of four bays, each bay having a length-width ratio of 10, were tested at a mach number of 3.0 at dynamic pressures ranging from 1,500 psf to 5,000 psf and at stagnation temperatures

from 300 f to 655 f . the panels were restrained by the supporting structure in such a manner that partial thermal expansion of the skins could occur in both the longitudinal and lateral directions . a boundary faired through the experimental flutter points consisted of a flat- panel portion, a buckled- panel portion, and a transition point at the intersection of the two boundaries . in the region where a panel must be flat when flutter occurs, an increase in panel skin temperature (or midplane compressive stress) makes the panel more susceptible to flutter . in the region where a panel must be buckled when flutter occurs, the flutter trend is reversed . this reversal in trend is attributed to the panel postbuckling behavior ."

* **Explanation:** The relevant document contains lot of exactly matching words similar to the query. so we could retrieve the relevant document.

2.4 Observations

* Precision (Mean precision) decreases and Recall increases as k increases, this implies that more relevant documents are retrieved in the starting (i.e. most of the top most retrieved documents are relevant).
* F-score increases and remained almost constant. (this implies a stable trade-off balance between precision and recall. It was low in the starting because of low recall values.)
* MAP is more smooth curve with less variation compared to precision curve. It decreased slowly compared to precision curve.
* For nDCG it decreased and became saturated. (It was implemented by changing relevance scores for query positions 1 to 4 -> 3 to 0 (i.e. 1 –> 3, 2 –> 2, 3 –> 1, 4 –> 0)). nDCG curve shows that the IR system could perform better (because ideal nDCG equal to 1) it shows more or less constant variation as k is increased

2.5 Hypothesis Testing

* **Null Hypothesis:** VSM-1 and VSM-2 performs similar in terms of precision, recall, f-score, n-DCG.
* **Alternate Hypothesis :** VSM-1 and VSM-2 doesn't perform similar in terms of precision, recall, f-score, n-DCG.
* **Approach :** For this task, we observed the precision, recall, f-score and n-DCG for VSM-1 and VSM-2 on given 225 queries in the cranfield queries. Then, we apply two sample two tailed t-test to evaluate the hypothesis on each individual metrics.
  Result of t-test:

Metric	t-statistic	p-value
Precision	-1.69241	0.0912622
Recall	-1.76362	0.0784774
FScore	-1.90723	0.0571297
nDCG	-1.75331	0.0802332

* **Conclusion :** All the p-values are much significant (i.e. greater than 0.05). So, we cannot reject the null hypothesis. We conclude that VSM-1 and VSM-2 performs almost similar, for a given query.

2.6 Limitations of Vector Space Model :

The following are limitations of vector space model, which we are trying to address :

* **Curse of Dimensionality :** Long documents are poorly represented because they have poor similarity values (a small scalar product and a large dimensionality)
* Vector space model assumes the orthogonality of dimensions (words), but it is not true in general.
* Semantic sensitivity. Documents with similar context but different term vocabulary won't be associated.
* The order in which the terms appear in the document is lost in the vector space representation.
* The document vectors are highly sparse and this leads to poor similarity values between documents and also between query and documents.

From the next section onwards we will try to address these limitations.

3 Document Representation & Orthogonality(limitation 1 & 2):

Similar words occur in similar documents, similar documents have similar words, to break this circularity, we use the latent variables called concepts. So, we map the documents to concept space (analogical to projection on eigen vectors) to find similarity between the documents using higher order associations.This also helps us in reducing the dimension of vector representation for all documents by using LSA for better representation in terms of space(storage of huge vectors).LSA solves the problem of orthogonality of dimensions to an extent i.e mitigates the problem of identifying synonymy(merge the dimensions associated with terms that have similar meanings), by capturing the higher order associations between the words.

3.1 Latent Semantic Analysis (LSA)

LSA assumes there exists a hidden semantic structure in the data that is partially obscured by the randomness of word choice with respect to retrieval.Singular value decomposition is a statistical technique used to identify the latent structure and remove the arbitrary noise in the data. [2]
There could be various reasons for these approximations (applying LSA on term-document matrix):

1. The original term-document matrix is presumed too large for the computing resources; in this case, the approximated low rank matrix is interpreted as an approximation.
2. The original term-document matrix is presumed noisy: for example, anecdotal instances of terms are to be eliminated. From this point of view, the approximated matrix is interpreted as a de-noisified matrix (a better matrix than the original)
3. The original term-document matrix is presumed overly sparse relative to the "true" term-document matrix. That is, the original matrix lists only the words actually in each document, whereas we might be interested in all words related to each document—generally a much larger set due to synonymy.

LSA Versions : We have implemented two versions of LSA :

1. LSA (corpus : cranfield docs) on TF-IDF matrix. LSA version-1
2. LSA (corpus : cranfield docs & brown corpus) on TF-IDF matrix. LSA version-2

Procedure :

1. Construct a term document matrix using TF-IDF
2. Applying SVD on the term document matrix to get the k-rank approximation of term-document matrix where k is a hyperparameter which we will tune using "gridsearchcv" in sklearn learnt from the data. k is nothing but the number of concepts, and eigen vectors in svd trasnformations implies the concept space.
3. Now we transform our query and documents into the concept space. cosine similarity is used to find the similarity between query and the documents. Ranking is done based on the similarity scores

3.2 LSA Hyperparameter Tuning

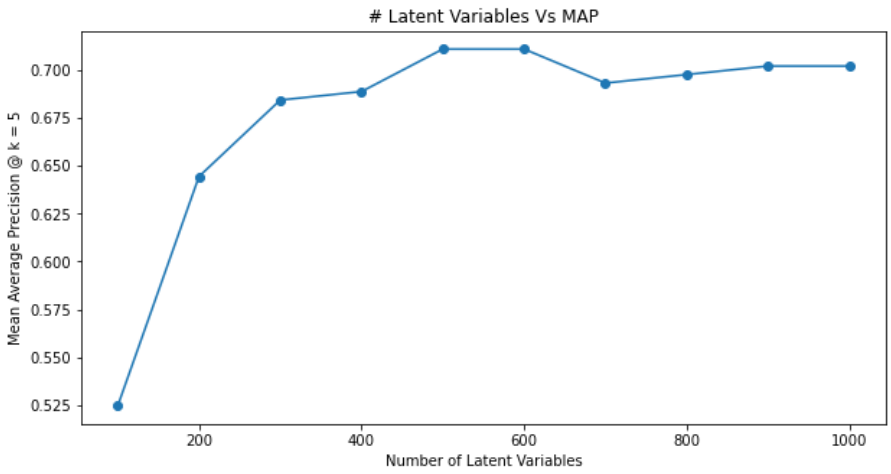


Fig. 6: No. of Latent Variables Vs MAP

The best value of evaluation measures(MAP, Recall, nDCG,...) were found around $k = 500$ (latent dimensions.). See figure 6

3.3 VSM-2 Vs LSA-500

The comparison of LSA-1(without brown corpus) with VSM-2 is shown in the table 2

	LSA with 500 latent dimensions					Basemodel				
k	**Precision**	**Recall**	**f-score**	**MAP**	**n-DCG**	**Precision**	**Recall**	**f-score**	**MAP**	**n-DCG**
1	0.707	0.119	0.195	0.707	0.532	0.693	0.12	0.193	0.693	0.53
2	0.578	0.19	0.269	0.738	0.414	0.567	0.188	0.266	0.733	0.412
3	0.513	0.244	0.307	0.741	0.394	0.501	0.241	0.302	0.734	0.39
4	0.471	0.294	0.335	0.733	0.397	0.461	0.287	0.328	0.732	0.39
5	0.418	0.317	0.333	0.728	0.395	0.412	0.313	0.328	0.721	0.391
6	0.383	0.346	0.335	0.718	0.4	0.379	0.342	0.331	0.71	0.396
7	0.363	0.378	0.342	0.701	0.409	0.352	0.368	0.334	0.697	0.402
8	0.34	0.399	0.339	0.691	0.413	0.332	0.389	0.33	0.687	0.407
9	0.32	0.418	0.336	0.679	0.417	0.311	0.406	0.325	0.68	0.411
10	0.302	0.433	0.33	0.67	0.422	0.291	0.419	0.318	0.674	0.414

Table 2: LSA-1 Vs VSM-2

The comparison of LSA-2 (LSA with large corpus) is shown in the table [3]

	LSA with brown corpus (500 latent dim)					Basemodel				
k	**Precision**	**Recall**	**f-score**	**MAP**	**n-DCG**	**Precision**	**Recall**	**f-score**	**MAP**	**n-DCG**
1	0.702	0.117	0.192	0.702	0.532	0.693	0.12	0.193	0.693	0.53
2	0.573	0.186	0.264	0.733	0.414	0.567	0.188	0.266	0.733	0.412
3	0.511	0.244	0.307	0.738	0.393	0.501	0.241	0.302	0.734	0.39
4	0.458	0.285	0.325	0.733	0.392	0.461	0.287	0.328	0.732	0.39
5	0.42	0.319	0.334	0.716	0.392	0.412	0.313	0.328	0.721	0.391
6	0.379	0.341	0.331	0.71	0.393	0.379	0.342	0.331	0.71	0.396
7	0.357	0.372	0.336	0.696	0.403	0.352	0.368	0.334	0.697	0.402
8	0.336	0.393	0.334	0.684	0.409	0.332	0.389	0.33	0.687	0.407
9	0.315	0.409	0.329	0.673	0.414	0.311	0.406	0.325	0.68	0.411
10	0.298	0.427	0.325	0.662	0.418	0.291	0.419	0.318	0.674	0.414

Table 3: LSA-2 Vs VSM-2

The general trends of evaluation metrics for LSA-500 is shown in figure 7

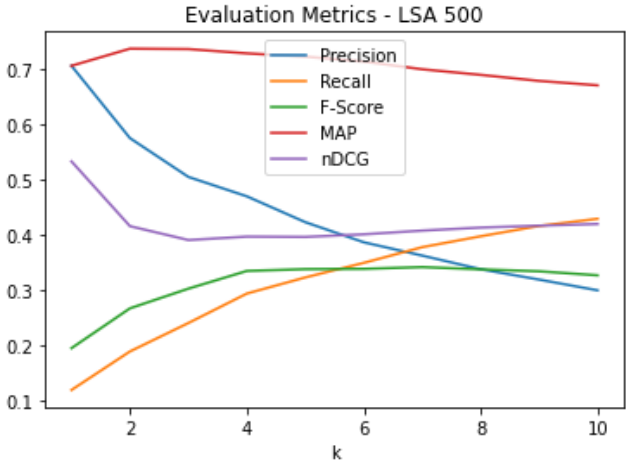


Fig. 7: Evaluation metrics trend of LSA-500

The distribution statistics of precision for LSA-500 and its comparison with VSM-2 is shown in figures 8, 9, 10 and 11

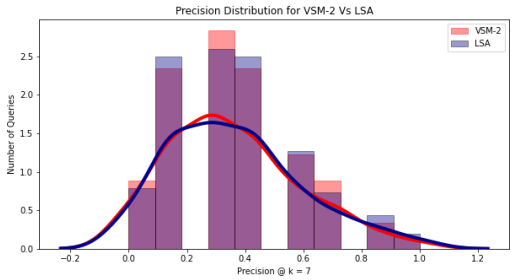


Fig. 8: precision distribution LSA-500 Vs VSM-2 @ k=7

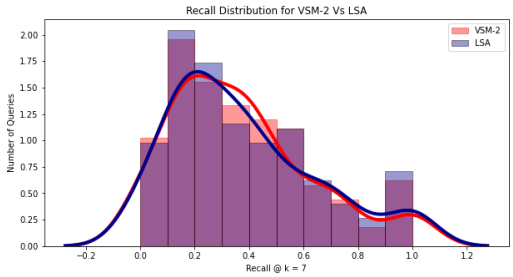


Fig. 9: Recall distribution LSA-500 Vs VSM-2 @ k=7

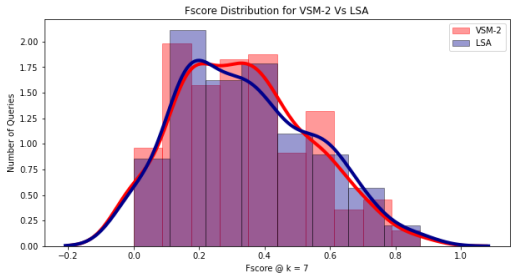


Fig. 10: FScore distribution LSA-500 Vs VSM-2 @ k=7

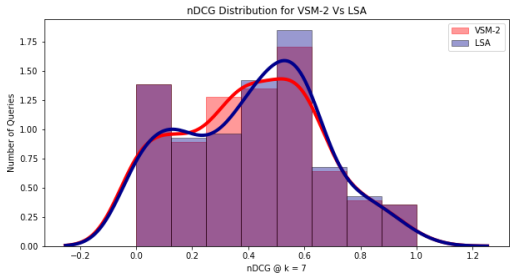


Fig. 11: nDCG distribution LSA-500 Vs VSM-2 @ k=7

3.4 Observations

From the distributional graphs, we observe that LSA shifts the distribution slightly towards right. This implies that Document Representation is improved using Latent Semantic Analysis.
Few Examples where the LSA performed better than baseline model

* **query id :** 109
* **query 174 :** 'panels subjected to aerodynamic heating .'
* **relevant documents :** [860, 861, 606, 980, 12, 766]
* **documents retrieved by VSM 2:** [1008, 859, 658, 857, 856, 391, 627, 766, 858, 948]
* **documents retrieved by LSA-500 :** [859,1008,658,857,856,766, 858]
  LSA precision: 0.142
  Doc 766 : "experimental investigation at mach number of 3. 0 of effects of thermal stress and buckling on flutter characteristics of flat single-bay panels of length-width ratio 0. 96 . flat, single-bay, skin stiffener panels with length-width ratios of 0.96 were tested at a mach number of 3.0, at dynamic pressures ranging from 1,500 to stagnation temperatures from 300 f to effects of thermal stress and buckling on the flutter of such panels . the panels supporting structure allowed partial thermal expansion of the skins in both the longitudinal and lateral directions . panel skin material and skin thickness were varied . a boundary faired through the experimental flutter points consisted of a flat-panel portion, a buckled-panel portion, and a transition point, at the intersection of the two boundaries, where a panel is most susceptible to flutter . the flutter region consisted of two fairly distinct sections, a large-amplitude flutter region and a small-amplitude flutter region . the results show that an increase in panel skin temperature flutter . the flutter trend for buckled panels is reversed . use of a modified temperature parameter, which approximately accounts for the effects of differential pressure and variations in panel skin material and skin thickness, reduced the scatter in the data which resulted when these effects were neglected . the results are compared with an exact theory for clamped panels for the condition of zero midplane stress . in addition, a two-mode /transtability/ solution for clamped panels is compared with the experimentally determined transition point ."
* **query id :** 184
* **query 174 :** 'work on small-oscillation re-entry motions .'

* **relevant documents :** [32, 67, 715, 717, 716, 499, 1379, 639]
* **documents retrieved by VSM 2:** [207, 281, 1113, 515, 1348, 164, 917, 1346, 715, 658]
* **documents retrieved by LSA-500 :** [207,1113,515,281,715,639,164,917]
  LSA precision: 0.285
  Doc 715 : "motion of a ballistic missile angularly misaligned with the flight path upon entering the atmosphere and its effect upon aerodynamic heating, aerodynamic loads and miss distance . an analysis is given of the oscillating motion of a ballistic missile which upon entering the atmosphere is angularly misaligned with respect to the flight path . the history of the motion for some example missiles is discussed from the point of view of the effect of the motion on the aerodynamic heating and loading . the miss distance at the target due to misalignment and to small accidental trim angles is treated . the stability problem is also discussed for the case where the missile is tumbling prior to atmospheric entry .",
  Doc 639: "analytical study of the tumbling motions of vehicles entering planetary atmospheres . the tumbling motion of vehicles entering planetary atmospheres is analyzed . a differential equation governing the tumbling motion, its arrest, and the subsequent oscillatory motion is obtained and identified as the equation for the fifth painleve transcendant . an approximate analytical solution for the transcendant is derived . comparisons with results obtained from numerical integration of the exact equations of motion indicate that the solution for the angle-of-attack history is sufficiently accurate to be of practical use ."
* **explanation :** LSA performed better because it captured higher order associations between the words, for example, in doc 766, the word thermal and temperature are related to heating in the query. As a result, it identified the relevant document, But in VSM-2, it can't retrieve document based on higher order association.

3.5 Limitations of LSA:

* The most important limitation of LSA is the extreme computational effort behind SVD. The computational complexity of this algorithm is $O(n^2 * k^3)$, where n is the number of documents plus the number of terms, and k is the number of embedded latent dimensions.
* The interpretability of the model is also lost to an extent.
* It partially solves Synonymy (words with similar meaning), but not Polysemy (same word with different meaning)

3.6 Hypothesis Testing LSA vs VSM-2

* **Null Hypothesis:** LSA-500 and VSM-2 performs similar in terms of precision, recall, f-score, n-DCG.
* **Alternate Hypothesis :** LSA-500 and VSM-2 doesn't perform similar in terms of precision, recall, f-score, n-DCG.
* **Approach :** For this task, we observed the precision, recall, f-score and n-DCG for LSA-500 and VSM-2 on given 225 queries in the cranfield queries. Then, we apply two sample two tailed t-test to evaluate the hypothesis on each individual metrics.
  Result of t-test:

Metric	**t-statistic**	**p-value**
Precision	-0.176353	0.860097
Recall	-0.252374	0.80086
FScore	-0.263295	0.792445
nDCG	-0.29344	0.769322

* **Conclusion :** All the p-values are much significant (i.e. greater than 0.05). So, we cannot reject the null hypothesis. We conclude that LSA-500 and VSM-2 performs almost similar, for a given query.

4 Addressing Semantic Sensitivity(Limitation 3):

Different words can have the same meaning called synonyms. So using Query Expansion we would like to get the similar words using word2vec.

4.1 Query Expansion(QE) model

Query Expansion broadens the query by introducing additional tokens or phrases. The search engine automatically rewrites the query to include them. For example, the query vp marketing becomes **(vp OR "vice president")** marketing.The implementation details are :

1. We train a **continuous bag of words(CBOW)**[5] model on the corpus and produce vector representation of all words in the corpus. These word vectors are also known as word embeddings.
2. For every word in the query, we'll take the top most similar word (cosine similarity), matching with the query word and append it to the query.
   This can be illustrated for a word as follows in figure 12

```
[282]:  1 res = np.array(model.wv.most_similar(positive=["good"]))[:,0].tolist()
        2 res

[282]: ['agreement',
        'result',
        'experimental',
        'comparison',
        'theoretical',
        'data',
        'compare',
        'satisfactory',
        'prediction',
        'reasonably']
```

Fig. 12: query expansion model predicting similar words to "good".

3. Now we take the query's vector representation and rank the documents as per cosine similarity values.

Note : Here size of corpus is very small to train good CBOW model, so we didn't get the improvement in the results. But this can be very helpful if we have good amount of data to train on.

4.2 VSM-2 vs QE

The comparison of Query Expansion model is shown in the table [4]

	Query expansion model					**Base model 2**				
k	**Precision**	**Recall**	**f-score**	**MAP**	**n-DCG**	**Precision**	**Recall**	**f-score**	**MAP**	**n-DCG**
1	0.568	0.0968	0.158	0.568	0.432	0.693	0.12	0.193	0.693	0.53
2	0.449	0.145	0.207	0.609	0.333	0.567	0.188	0.266	0.733	0.412
3	0.394	0.188	0.24	0.622	0.319	0.501	0.241	0.302	0.734	0.39
4	0.358	0.220	0.258	0.621	0.317	0.461	0.287	0.328	0.732	0.39
5	0.334	0.253	0.266	0.616	0.321	0.412	0.313	0.328	0.721	0.391
6	0.312	0.281	0.272	0.605	0.328	0.379	0.342	0.331	0.71	0.396
7	0.29	0.303	0.273	0.598	0.33	0.352	0.368	0.334	0.697	0.402
8	0.275	0.322	0.274	0.585	0.333	0.332	0.389	0.33	0.687	0.407
9	0.262	0.340	0.275	0.578	0.340	0.311	0.406	0.325	0.68	0.411
10	0.247	0.345	0.270	0.570	0.345	0.291	0.419	0.318	0.674	0.414

Table 4: Query expansion model Vs Basemodel 2

The general trends of evaluation metrics for QE is shown in figure 13

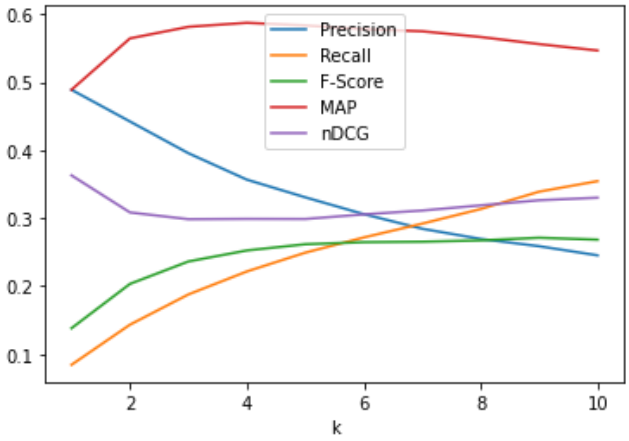


Fig. 13: Evaluation metrics trend of QE

The distribution statistics of recall for QE (Query Expansion) and its comparison with VSM-2 is shown in figures 15, 14, 17, 16

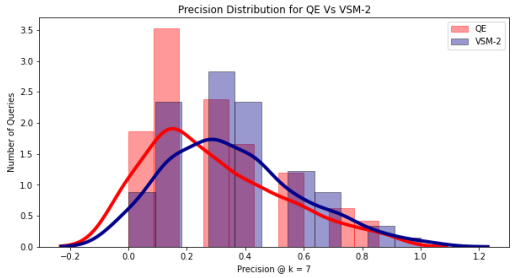


Fig. 14: precision distribution QE Vs VSM-2 @ k=7

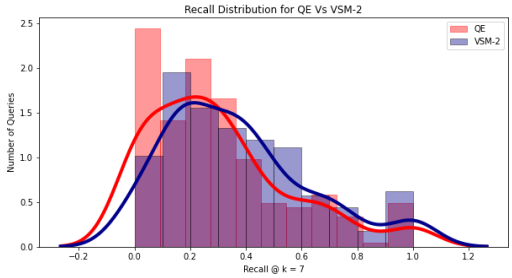


Fig. 15: Recall distribution QE Vs VSM-2 @ k=7

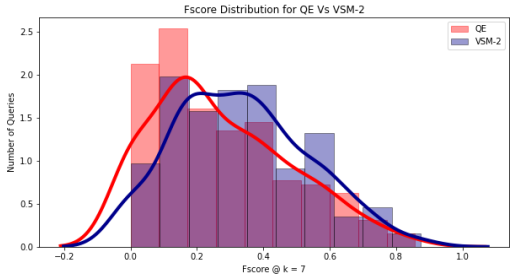


Fig. 16: FScore distribution QE Vs VSM-2 @ k=7

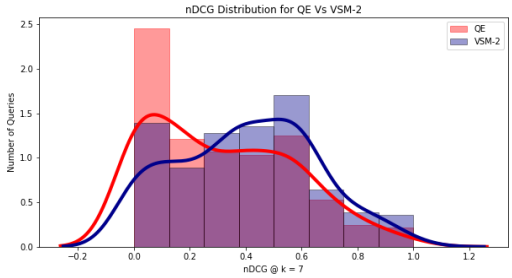


Fig. 17: nDCG distribution QE Vs VSM-2 @ k=7

4.3 Observations

From the distributional graphs, we observe that the distribution of QE shifted towards left (QE model performed badly), one of the main reason might be because of poor word2vec representation (very small training data). Few Examples where query expansion model performed well compared to VSM-2

* **query id :** 85
* **preprocessed query :** parameter seriously influence natural transition laminar turbulent flow model wind tunnel.

* **Expanded Query :** dependence vas delay strengthen roughness layer injection dimensional cover tunnel wind parameter seriously influence natural transition laminar turbulent flow model wind tunnel.
* **Words Added :** ['dependence', 'vas', 'delay', 'strengthen', 'roughness', 'layer', 'injection', 'dimensional', 'cover', 'tunnel']
  VSM Recall : 0.0 QE Recall : 0.2
* **documents retrieved by QE:** [710, 80, 314, 40, 7, 1153, 1155, 1211, 431, 96]
* **relevant documents :** [608, 406, 606, 710, 546]
  Doc 710: "the smallest height of roughness capable of affecting boundary-layer transition . an investigation was made to determine the smallest size of isolated roughness that will affect transition in a laminar-boundary layer . critical heights for three types of roughness were found in a low-speed wind tunnel . the types were /1/ two-dimensional spanwise wires, /2/ three-dimensional discs, and /3/ a sandpaper type . in addition to type of roughness , test variables included the location of roughness , pressure distribution, degree of tunnel turbulence, and length of natural laminar flow . the most satisfactory correlation parameter was found to be the roughness reynolds number, based on the height of roughness and flow properties at this height . the value of this critical reynolds number was found to be substantially independent of all test variables except the shape of roughness . this parameter also correlates well other published data on critical roughness in low-speed flow . the value of the roughness reynolds number necessary to move transition forward to the roughness itself was also determined for the three types of roughness and was found to be approximately constant for a given type of roughness . an investigation of the limited amount of available data on critical roughness in supersonic flow indicates that the effects of roughness may still be correlated by the roughness reynolds number . the value of this reynolds number depends primarily on the mach number at the top of the roughness . when this mach number is greater than 1.0, the roughness reynolds number based on conditions behind a shock is probably the characteristic parameter .",
* **query id :** 184
* **preprocessed query :** work small oscillation entry motion.
* **Expanded Query :** limitation upon estimation atmosphere linearize work small oscillation entry motion
* **Words Added :** ['limitation', 'upon', 'estimation', 'atmosphere', 'linearize']
  VSM Recall : 0.0 QE Recall : 0.25
* **documents retrieved by QE:** [207, 715, 944, 1348, 982, 164, 917, 1345, 716, 639]
* **relevant documents :** [32, 67, 715, 717, 716, 499, 1379, 639] **Doc 715 :** "motion of a ballistic missile angularly misaligned with the flight path upon entering the atmosphere and its effect upon aerodynamic heating, aerodynamic loads and miss distance . an analysis is given of the oscillating motion of a ballistic missile which upon entering the atmosphere is angularly misaligned with respect to the flight path . The history of the motion for some example missiles is discussed from the point of view of the effect of the motion on the aerodynamic heating and loading . the miss distance at the target due to misalignment and to small accidental trim angles is treated . the stability problem is also discussed for the case where the missile is tumbling prior to atmospheric entry .",

  Doc 716: "study of the oscillatory motion of manned vehicles entering the earth's atmosphere . an analysis is made of the oscillatory motion of vehicles which traverse arbitrarily prescribed trajectories through the atmosphere . expressions for the oscillatory motion are derived as continuous functions of the properties of the trajectory . results are applied to a study of the oscillatory behavior of re-entry vehicles which have decelerations that remain within limits of human tolerance . it is found that a deficiency of aerodynamic damping for such vehicles may have more serious consequences than it does for comparable ballistic missiles ."
* **Explanation:** Because of the additionally added synonyms (or co-occuring words), we were able to capture the relevant documents which cannot be identified by the VSM-2 model. For the query: 85, QE model retrieved doc 710, which is a relevant document. This is because, the words added by the QE model roughness, tunnel, dimensional are helping in matching with the relevant document.

4.4 Hypothesis Testing VSM-2 Vs QE

* **Null Hypothesis:** QE and VSM performs similar in terms of precision, recall, f-score, n-DCG.

* **Alternate Hypothesis :** QE and VSM doesn't perform similar in terms of precision, recall, f-score, n-DCG.

* **Approach :** For this task, we observed the precision, recall, f-score and n-DCG for LSA-500 and VSM on 225 queries in the cranfield queries. Then, we apply two-tailed t-test to evaluate the hypothesis on each

individual metrics.
Result of t-test:

Metric	t-statistic	p-value
Precision	-3.111	0.0019836
Recall	-2.75348	0.00613633
FScore	-3.18545	0.0015464
nDCG	-3.56525	0.000402662

∗ **Conclusion :** We reject the null hypothesis. We conclude that QE and VSM-2 do not perform similarly for a given query. From distributions, we can say that QE model performed slightly better for few queries and worse for other queries.

5 Clustering Methods to reduce Retrieval time

As the number of documents to search increases, the time taken to retrieve relevant documents to a particular query also increases linearly. So we came up with two methods to reduce the search time required to retrieve relevant documents :

1. KMeans Clustering
2. Topic modeling using LDA.

5.1 K-means Clustering

Clustering is an un-supervised method of grouping data. Here, we use K-means clustering[3]. (an application of EM algorithm). The detailed steps of implementation are :

1. Apply KMeans clustering on the TF-IDF matrix of documents, and cluster them into 'k' clusters. The value of k can be found using elbow plot. The best value was found to be around 6. See figure [18] and [19] :

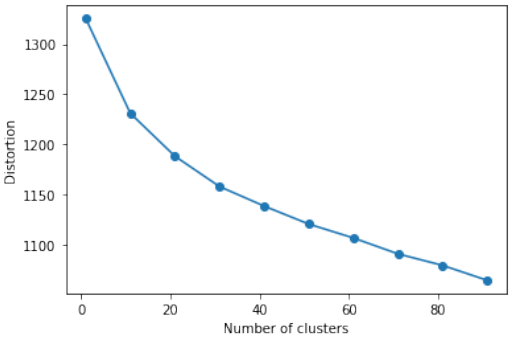


Fig. 18: Elbow plot to find optimal clusters

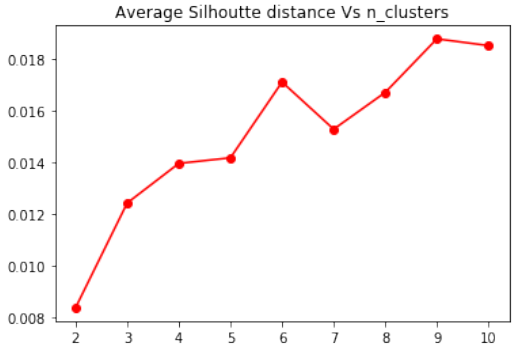


Fig. 19: Avg Silhoutte distance

2. Now for a given query, we find the cosine similarity between the query and k cluster centres. Based on the cosine similarity scores, we decide on which cluster the query belongs to.
3. Now, we take cosine similarity of a query with all documents in that cluster and rank those documents.

5.2 Kmeans Vs VSM-2

Comparison interms of Retrieval Time :
Comparing Clustering model with Base model, in terms of average time of retrieval is shown in table [7]
We see that, the **retrieval time is reduced by a factor of 3**. This seems to be a good improvement.
Comparison interms of evaluation metrics
The comparison of clustering model with base line is shown in table [6] :

There is a decrement in all evaluation measures by approximately 0.1. This trade-off is expected, because, we have reduced our search space. However, if speed of retrieval is more important, then one can use this method.

	Clustering, k = 6	**Base model**
Avg retrieval time	3.7 ms	11 ms

Table 5: Retrieval time comparison

	Clustering, k = 6					**Base model 2**				
k	**Precision**	**Recall**	**f-score**	**MAP**	**n-DCG**	**Precision**	**Recall**	**f-score**	**MAP**	**n-DCG**
1	0.573	0.093	0.154	0.573	0.412	0.693	0.12	0.193	0.693	0.53
2	0.458	0.145	0.207	0.609	0.318	0.567	0.188	0.266	0.733	0.412
3	0.407	0.188	0.24	0.611	0.302	0.501	0.241	0.302	0.734	0.39
4	0.369	0.222	0.258	0.602	0.299	0.461	0.287	0.328	0.732	0.39
5	0.34	0.249	0.266	0.597	0.303	0.412	0.313	0.328	0.721	0.391
6	0.307	0.267	0.264	0.594	0.304	0.379	0.342	0.331	0.71	0.396
7	0.29	0.291	0.268	0.583	0.31	0.352	0.368	0.334	0.697	0.402
8	0.27	0.305	0.265	0.576	0.313	0.332	0.389	0.33	0.687	0.407
9	0.253	0.319	0.261	0.57	0.317	0.311	0.406	0.325	0.68	0.411
10	0.237	0.328	0.255	0.562	0.319	0.291	0.419	0.318	0.674	0.414

Table 6: KMeans Clustering model Vs VSM-2

5.3 Hypothesis Testing

Null Hypothesis: LSA Mean Retrieval time(efficacy) with and without clustering technique is same.
Alternate Hypothesis : LSA Mean Retrieval time(efficacy) with and without clustering technique is not same.

Approach : For this task, we observed the retrieval time for LSA with and without clustering on 200 queries in the cranfield queries. Then, we apply two-tailed t-test to evaluate the hypothesis.
The retrieval time for LSA with and with out clustering of 200 queries is shown in figure 20
Result of t-test: we got a $t - value = 18.24$ and $p - value = 1.47e - 45$ which is approximately 0. which is

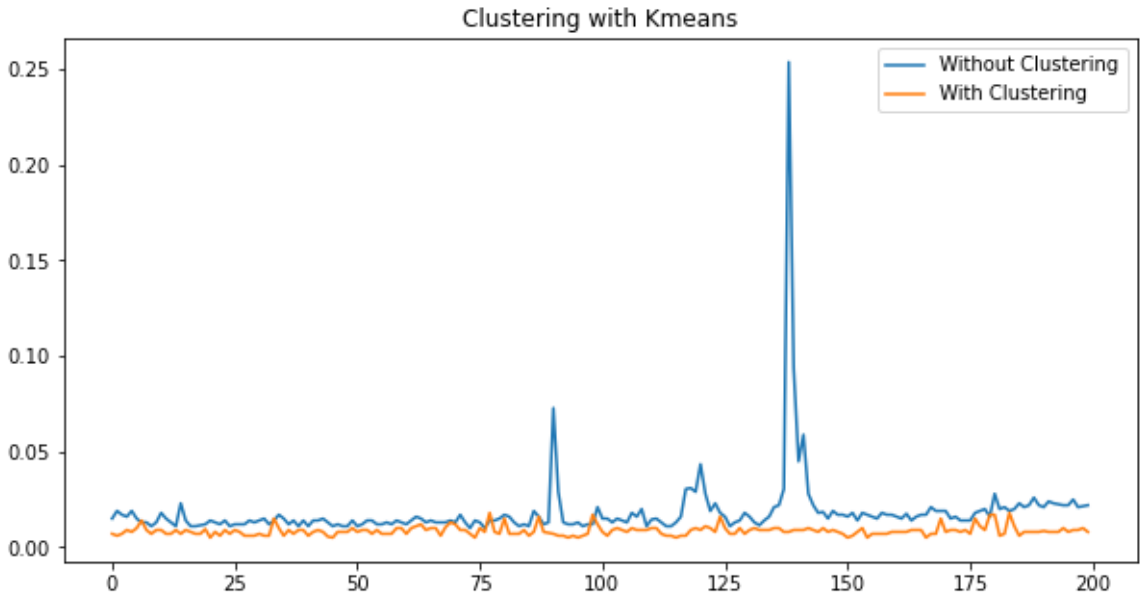


Fig. 20: Kmeans Vs VSM-2 Retrieval time

much less than significance level ($\alpha = 0.05$). So, we reject the Null Hypothesis.
Conclusion : We conclude that Clustering method reduces the retrieval time of documents, for a query.
Note: From the graph, it is clear that the clustering performs better than the without clustering.

5.4 Topic modeling using LDA

Topic modeling is an unsupervised machine learning technique that automatically analyzes text data to determine cluster words for a set of documents. More about this can be found here[6].

under lying assumptions of LDA (Latent Dirichlet Allocation) is same as that of LSA.(i.e. similar topics make use of similar words). The main difference between LSA and LDA is that LDA assumes that the distribution of topics in a document and the distribution of words in topics are Dirichlet distributions.
There are two hyper-parameters that control document and topic similarity, known as alpha and beta, respectively. A low value of alpha will assign fewer topics to each document whereas a high value of alpha will have the opposite effect. A low value of beta will use fewer words to model a topic whereas a high value will use more words, thus making topics more similar between them. The output of algorithm is a vector that contains the coverage of every topic for the document being modeled. For information on computing these probabilities, we refer to the original LDA paper[7]. Model implementations steps are:

1. Documents are grouped, based on similar topics.
2. Now, we will compute the centroids of all topics. (centroid based classifier[8]). Then we compute cosine similarity of these centroids with given query vector.
3. Only the documents belonging to that topic would be considered for ranking.

5.5 LDA Vs VSM-2

Comparison interms of Time of retrieval: Comparing LDA model with Base model, in terms of average time of retrieval is shown in table [7]

	topics, k = 6	**Base model**
Avg retrieval time	8 ms	11 ms

Table 7: Retrieval time comparison

We see that, the **retrieval time is reduced but not significantly**. Note that there is less trade-off when we used Kmeans clustering than LDA. This might be because of not having enough data to approximate the alpha and beta parameters of probability distribution. The high retrieval time(compared to Kmeans) for LDA can be because of containing many documents in a single topic.
comparison interms of evaluation metrics:
Comparing the LDA model, with baseline model in table [8] :

	LDA model					**Base model 2**				
k	**Precision**	**Recall**	**f-score**	**MAP**	**n-DCG**	**Precision**	**Recall**	**f-score**	**MAP**	**n-DCG**
1	0.484	0.085	0.139	0.484	0.338	0.693	0.12	0.193	0.693	0.53
2	0.407	0.133	0.188	0.547	0.277	0.567	0.188	0.266	0.733	0.412
3	0.367	0.178	0.223	0.564	0.271	0.501	0.241	0.302	0.734	0.39
4	0.337	0.209	0.239	0.567	0.276	0.461	0.287	0.328	0.732	0.39
5	0.307	0.237	0.247	0.562	0.275	0.412	0.313	0.328	0.721	0.391
6	0.279	0.257	0.247	0.551	0.278	0.379	0.342	0.331	0.71	0.396
7	0.255	0.27	0.241	0.546	0.282	0.352	0.368	0.334	0.697	0.402
8	0.24	0.286	0.241	0.542	0.285	0.332	0.389	0.33	0.687	0.407
9	0.229	0.302	0.24	0.536	0.29	0.311	0.406	0.325	0.68	0.411
10	0.215	0.312	0.235	0.533	0.295	0.291	0.419	0.318	0.674	0.414

Table 8: LDA model Vs Baseline

5.6 Hypothesis Testing

* **Null Hypothesis:** LSA Mean Retrieval time(efficacy) and LDA mean retrieval time is same.

* **Alternate Hypothesis :** LSA Mean Retrieval time(efficacy) and LDA mean retrieval time is not same.

* **Approach :** For this task, we observed the retrieval time for LSA and topic modeling (topics = 6) on 200 queries in the cranfield queries. Then, we apply two-tailed t-test to evaluate the hypothesis.
  The retrieval time for LSA with and with out clustering of 200 queries is shown in figure 21

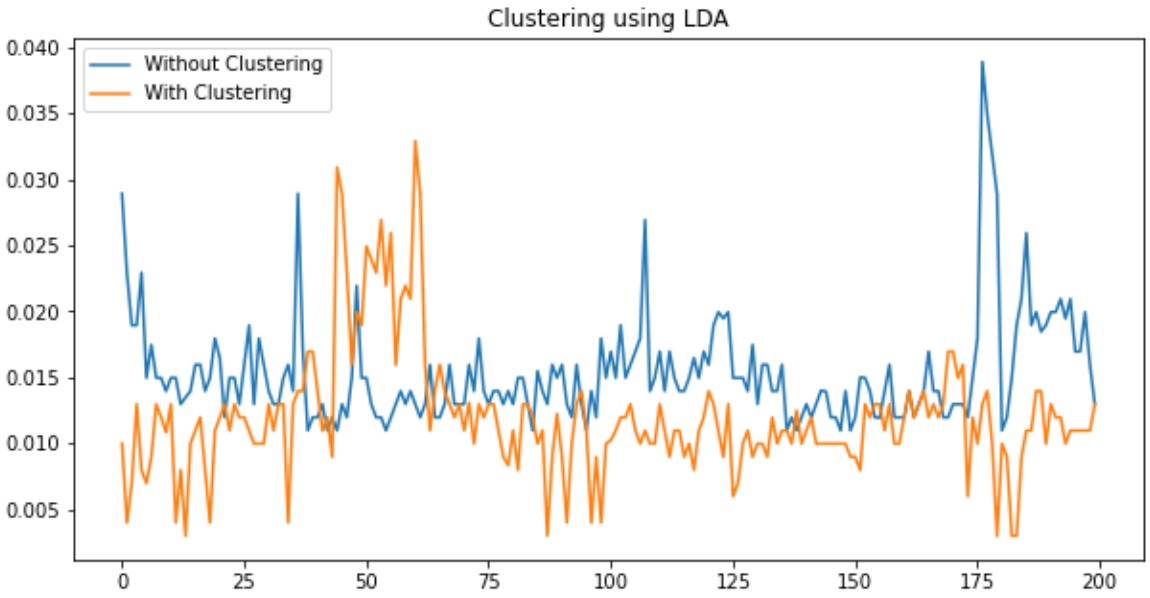


Fig. 21: LDA vs VSM-2 Retrieval time

Result of t-test: we got a $t - value = 7.54$ and $p - value = 2.74e - 11$ which is approximately 0. which is much less than significance level ($\alpha = 0.05$). So, we reject the Null Hypothesis.
Conclusion : We conclude that LDA method reduces the retrieval time of documents, for a query.
Note: From the graph, it is clear that the LDA performs better than LSA in terms of retrieval time.

6 Query Auto-Completion

To improve user interaction with search engine. we can model incomplete queries, or this can be further incorporated with browsers, so that they can provide auto-completion of a query that we type, as shown in figure22

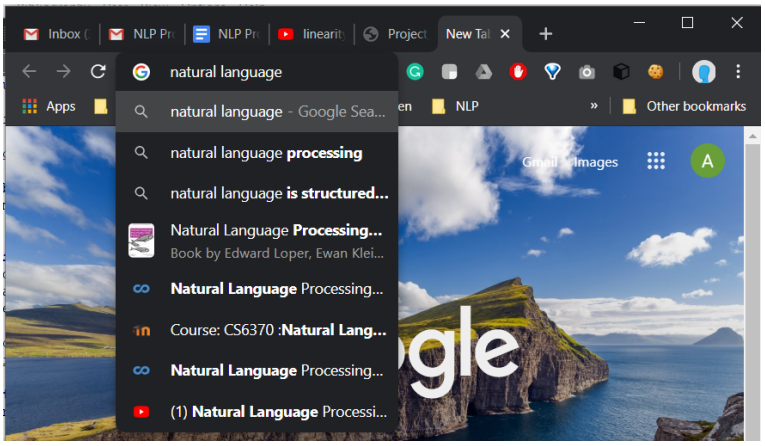


Fig. 22: query completion integrated with browser.

6.1 Query Completion model

The details are given below:

1. A shallow neural network model was built, using an Embedding layer with 16 dimensions (each word), followed by an LSTM layer, to take context into account, followed by a Dense layer and an output layer, with softmax (to predict the next word).

2. All the sentences in the queries are preprocessed,like we take, say 5 continuous words, in all sentences and use then as train input sentences, and the next word as train output sentence.
   This can be illustrated by the example: "when constructing aeroelastic models of heated high speed aircraft". Here the set of train sentences constructed would be : ['when constructing aeroelastic models of', 'constructing aeroelastic models of heated', 'aeroelastic models of heated high',...], corresponding labels would be ['heated', 'high', 'speed',...].
3. Once the model is trained, we can pass an incomplete query and get the next 'n' words.

This is illustrated as shown in figure23 we predict next 2 words

```
In [53]:  1 complete_query("experimental studies of creep",2)
Out[53]: 'experimental studies of creep buckling either'
```

Fig. 23: query completion model illustration.

if we try to predict more words, the quality of words deteriorates, below, in figure24 we show, the result of predicting next 10 words.

```
In [14]:  1 complete_query("experimental studies of creep",10)
Out[14]: 'experimental studies of creep buckling or must be developed to the fundamental three speed'
```

Fig. 24: query completion model illustration with more next words.

We couldn't find a way, to compare with a base line model. Only manual interpretation of some queries were considered to evaluate this model.

7 Summary:

In summary, we built two baseline models which are Baseline-1 and Baseline-2. Here Baseline-1 and Baseline-2 are differ by their pre-processing data techniques but both are vector space models. Because of extensive pre-processing techniques in Baseline-2, our dimensions in vector space model reduced and from the results, we could clearly say that Baseline-2 performed little better than the Baseline-1 model. Then we applied Latent Semantic analysis(LSA) on both pre-processed data to get better representation of term-document matrix.

LSA applied on extensive pre-processed data gives the best result among all the models. Here LSA improved the rank ordering of the retrieved documents(section 3) which in-turn improved the model. In section 4, we introduced new method called "Query Expansion Model" where we try to add the top-k similar words to the query, use this expanded query as the input to the IR system. Surprisingly, this did not improved the efficacy. This may be because of small training corpus.

Till section 5, we try to improve model effectiveness/efficacy(precision, recall,..). As we know, information retrieval(IR) systems are evaluated based on Efficiency also i.e time and space usage, latency. In Section 5, we try to reduce the retrieval time of documents, so we used Document clustering techniques and Observed a significant drop in retrieval time, at the cost of reduced recall and precision, which was expected.(trade-off)

Our best model is dependent on the exact query words, sometimes they may not be in the corpus. So, we have implemented a Query auto-completion in Section 6 to help the user to finish the query appropriately. Note, this auto-completion feature is helpful in the real-world deployment but not when you have a set of test queries to evaluate your system.

References

1. Stemming Vs Lemmatization, which is better ?
2. Latent semantic analysis, wikipedia
3. KMeans clusering reference
4. Billel Aklouche1, Ibrahim Bounhas1, Yahya Slimani1, Query Expansion Based on NLP and Word Embeddings
5. https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa
6. https://monkeylearn.com/blog/introduction-to-topic-modeling/
7. David M. Blei, Andrew Y.Ng, Michael I. Jordan, Latent Dirchlet Allocation, Journal of Machine Learning Research 3 (2003) 993-1022
8. Centroid based classifier

Thanking Note: We thank prof. Sutanu Chakraborti sir for giving this unique opportunity to work on a real world NLP problem. This project really helped us to explore in the domain of information retrieval systems and also created a great interest in further exploring in the domain of NLP. We thank TAs Monisha J and Devi G for their constant guidance in the project and clearing course doubts.