CS6370: Natural Language Processing
Project Report

Shashank M Patil
CH18B022
, Shania Mitra
CH18B067

Indian Institute of Technology, Madras

NOTE: The figures and tables mentioned in the explanation are linked to the corresponding Figures and Tables, and can be seen by clicking on the figure/table number appearing in the explanation.

1 Introduction and Problem Definition

Information retrieval (IR) is the process of obtaining information system resources that are relevant to an information need from a collection of those resources. Searches can be based on full-text or other content-based indexing. It is the science of searching for information in a document, searching for documents themselves, and also searching for the metadata that describes data, and for databases of texts, images or sounds.

The cranfield dataset contains 225 queries and 1400 documents and ground truth with relevant documents for each query, with positions of importance, ranging from 1 to 4, 1 being the most important.

The aim of this project is to build an information retrieval system to retrieve the relevant documents from the cranfield dataset for all queries in the dataset and evaluate its performance on 225 queries and 1400 documents using the following metrics: Precision@k, Recall@k, F-score@k, Mean Average Precision@k and nDCG@k, to see how the system fares against ground truth relevances.

2 Background and Related Work

2.1 Vector Space Model

Vector space model is an algebraic model for representing text documents (and any objects, in general) as vectors of identifiers (such as index terms). The term-specific weights in the document vectors are products of local and global parameters. The model is known as the term frequency-inverse document frequency model. The weight vector for document d is

$$\mathbf{v}_d = [w_{1,d}, w_{2,d}, \ldots, w_{N,d}]^T$$

Where,

$$w_{t,d} = \mathrm{tf}_{t,d} \cdot \log \frac{|D|}{|\{d' \in D \mid t \in d'\}|}$$

- $\mathrm{tf}_{t,d}$ is term frequency of term t in document d (a local parameter) $\log \frac{|D|}{|\{d' \in D | t \in d'\}|}$ is inverse document frequency (a global parameter).
$|D|$ is the total number of documents in the document set; $|\{d' \in D \mid t \in d'\}|$ is the number of documents containing the term t.

2.2 Exploration of Cranfield Dataset

In this project, we perform Information Retrieval on the Cranfield Dataset which contains 225 queries and 1400 documents on Aerodynamics. Each query has a list of relevant documents with their corresponding positions, ranging from 1-4. The relevance for each position is calculated as $Relevance = 5 - Position$ such that documents at position 1 have the highest relevance of 4, while the least relevant documents at position 4 have a relevance of 1.

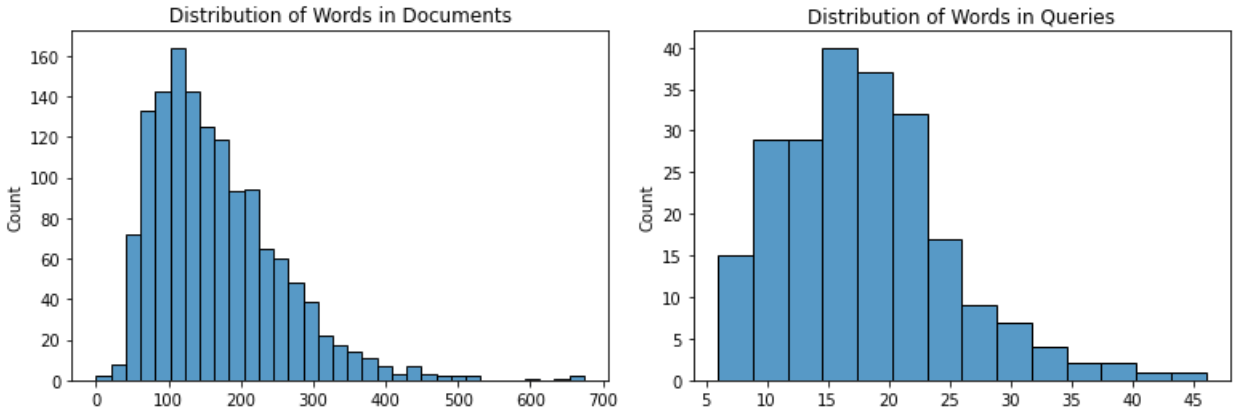

Fig. 1. (a) Distribution of Number of Words in Documents (b) Distribution of Number of Words in Queries of Cranfield Dataset

In Figure 1, it can be seen that the documents in the dataset have a large variation in the number of words, i.e., $0 - 700$. 2 documents, namely, 471 and 995 are empty strings with no words, i.e., no title, author, body or bibliography. The range of number of words in queries is lesser, ranging from $5 - 45$, suggesting that, in general, queries are smaller than the docs, although some may be larger.

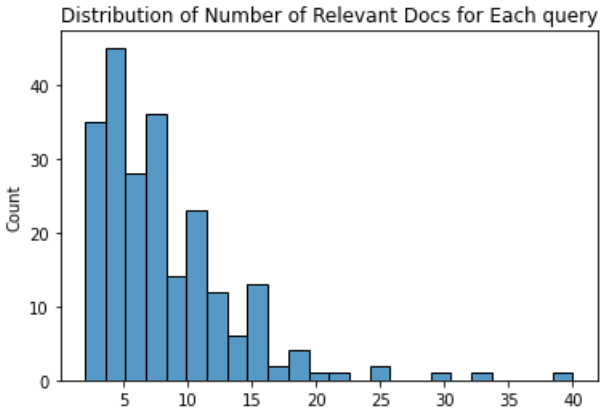

Fig. 2. Distribution of Number of Relevant Documents for the Queries in the dataset

On analysing the ground truth values in Figure 2. we see that most of the queries have around 5-10 relevant documents while very few generic queries have 40 relevant documents.

Figure 3 shows us that some documents such as 629, 572, 1213 are relevant to more queries than others. For example, Document 629 on "second-order effects in laminar boundary layers" is relevant to 9 queries possibly due to it speaking about the effects in boundary layers which is a central topic in Aerodynamics.

There are a total of 3402 words that occur only once among all documents. While most of these words are misspelled and can be used as candidates for misspelled words, many of the words are correctly spelled and are still present only in a single document, as we can observe in Table 2.

3 Motivation

3.1 Augmentation of Tokenization (AugTok)

As highlighted in the proposal,

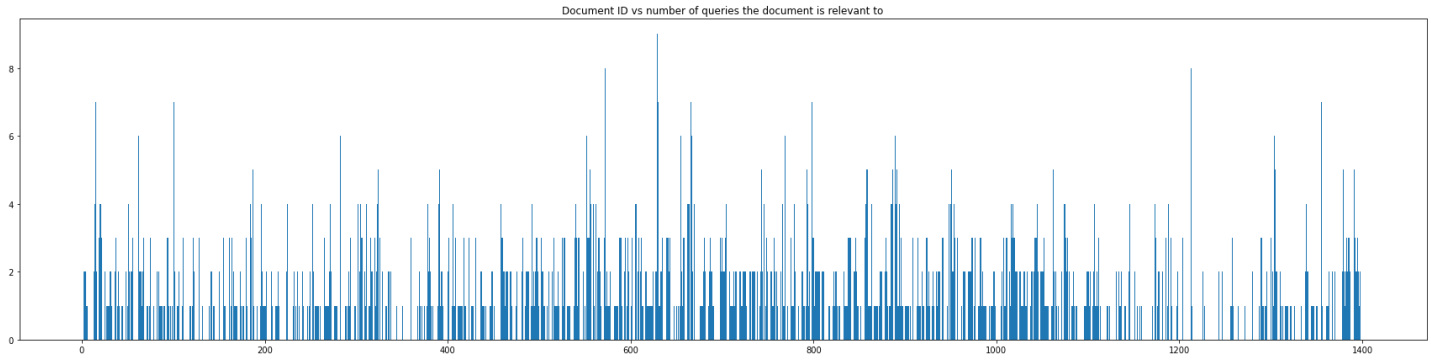

Fig. 3. Number of queries each document is relevant to

token	frequency	token	frequency
affinely	1	flow	2138
lardnert.j	1	pressure	1387
doylem.d.c	1	number	1358
east	1	boundary	1244
sellsc.c.I	1	layer	1190
gothic	1	result	1079
woodleyj.g	1	effect	962
electroform	1	method	909
catheralld	1	theory	900
ob	1	body	876

Table 1. (a) Top-10 most rarely occurring words in the documents (b) Top-10 most frequently occurring words in the documents

- All punctuations such as fullstops are considered tokens by the tokenizer
- Some queries and documents have erroneous words like "-dash". For example, Document 8.
- The tokenizer fails to split across hyphens. For example, 'real-gas', 'shock-induced', 'boundary-layer' remain as it is. This causes a problem especially since keywords are matched directly. Some documents and queries contain 'boundary layer', while others contain 'boundary-layer'. This poses a problem since documents containing 'boundary-layer' are not retrieved for queries containing 'boundary layer' and vice versa, which may lead to relevant documents not being retrieved.
- Tokenizer fails to clear out noisy punctuations. For example, "/slip flow/", "/boat-tail/". In case of "/slip flow/", it is split as "/slip", "flow/" and no document with slip or flow is retrieved since, by design, the model considers "slip" and "/slip" as orthogonal dimensions.
- Spelling errors exist in the documents and queries necessitating a spell check pre-processing step. For example, in Document 74, "turbulen coundary" is an incorrect representation of "turbulent boundary

On account of the above mentioned reasons, a pressing need to supplement the exisiting tokenization pipeline, i.e., Punkt Segmenter and PennTreeBank Tokenizer was felt.

3.2 Phrase Extraction

All documents belong to the domain of aerodynamics and hence, if phrases such as "boundary layer", "incompressible flow", etc. , which form real concepts in theory, are extracted and given weightage, it could lead to enhanced concept formation in methods like LSA and could prevent a query or document being assigned to an incorrect or less-relevant article dimension in the article space in ESA. Examples include "shock induced", "thermal stress", "hydrostatic pressure", "heat conduction", "Reynolds number", etc. In case of "thermal stress" and ESA, if left as it is, the document may get assigned to articles such as "thermos flask" or "thermal insulation" due to the presence of "thermal", and to "Trauma" or "Physical stress" due to the presence of "stress", which are clearly irrelevant. However on being used as a single entity, these irrelevant possibilities are ruled out.

Table 2. Examples of words that occur once but are correctly spelt

efficiently	anyone	else	validly	unnecessarily	virtue	determinant
formerly	wildly	empty	reality	collectively	determinant	viscid

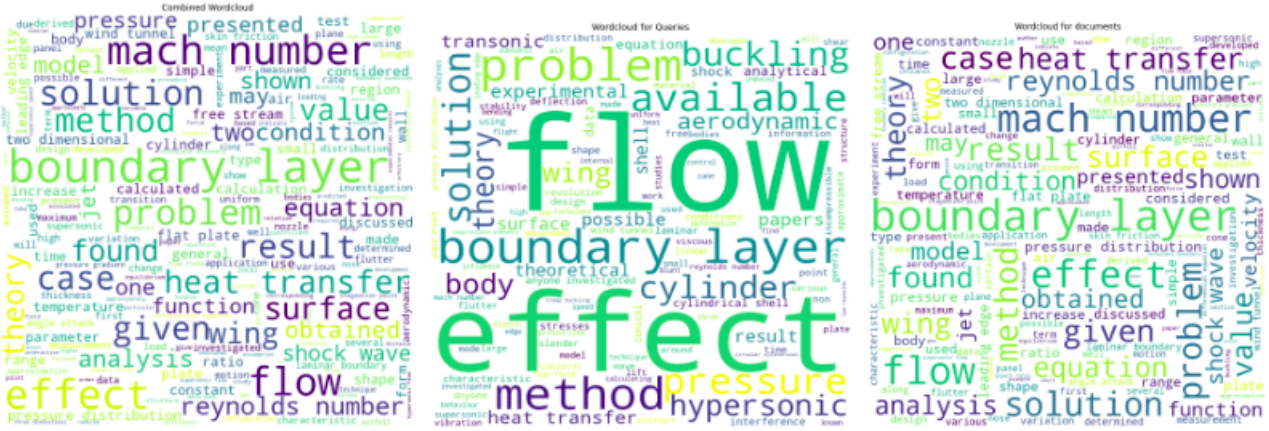

Fig. 4. Wordclouds indicating dominant terms in (a) Documents (b) Queries (c) Combined corpus (Queries + Documents)

3.3 Latent Semantic Analysis (LSA)

The vector space model does not take word-relatedness and overarching concepts into account, and performs only keyword matches which results in poor performance. Latent Semantic Analysis takes into account the semantic similarities between words by nature of its design because a query with terms that do not appear in a document may still end up close to a document in hyperspace. As a result, a query can return documents with terms that are semantically similar to query terms.

3.4 Explicit Semantic Analysis (ESA)

The central theme of all the documents in the cranfield dataset is Aerodynamics and thus, background knowledge on aerodynamics such as wikipedia articles and research papers would help in supplementing the ideas present in the documents. This would help capture semantic similarity instead of performing keyword matches, due to the presence of multiple semantically similar words in the articles/background knowledge used.

4 Methodology and Experiments

4.1 Augmentation of Tokenization

To improve upon the existing tokenization pipeline, functions were introduced to remove extra spaces, all punctuations, extra full stops and split across hyphens. Further two methods of spell-check were implemented:

- Method - 1: Context Insensitive Spellcheck using the SymSpell Compound algorithm which corrects non-dictionary spelling errors and even splits compound words into their individual words. This was done since most of the spelling errors were observed to be non-dictionary words.
- Method - 2: A simple Edit Distance based method wherein all words which occur only once are thought to be spelling errors and are corrected by replacing with a candidate from the remaining tokens (frequency $>$ 2) with the least edit distance

However, these could not be incorporated finally due to the following reasons:

- Method - 1 is extremely computational-resource itensive[1]

[1] Individual Systems crashed on running; Estimated time on Google Colab is 7-9 hours before which runtime disconnects or crashes

– In Method-2 the main assumption, i.e., words which occur only once in all documents are spelling errors, does not hold as we can see in Table 2.

Thus, the final pipeline consists of Punkt and PennTreeBank as before and functions to remove spaces, punctuations and noise words additionally.

4.2 Phrase extraction

To extract phrases of length 2 , we may use PMI or PPMI

$$pmi(x;y) \equiv \log \frac{p(x,y)}{p(x)p(y)} = \log \frac{p(x \mid y)}{p(x)} = \log \frac{p(y \mid x)}{p(y)}$$

PPMI clips all negative PMI values to zero. In this case study, the Phrases module from gensim.models has been used to extract phrases, which uses Normalized Pointwise Mutual Information.

$$npmi(x;y) = \frac{pmi(x;y)}{h(x,y)}$$

Where $h(x,y)$ is the joint self-information, which is estimated as $-\log_2 p(X=x, Y=y)$

Various libraries were tested to extract phrases from all the documents. However, Phrases from gensim.models gave the most apt phrases and was able to detect phrases such as “Boundary Layer”, “Hydrostatic pressure” and “Mach Number”, which form concepts in aerodynamics.

Examples of phrases extracted: ’Shock wave’, ’boundary layer’, ’laminar boundary’, ’skin friction’, ’aerodynamic heat’, ’heat conduction’, ‘jet propulsion’, etc.

Finally, we test our models with all three kinds of preprocessing :

– **Augmented Tokenization:** punctuation and extra space removal along with Punkt + PennTreeBank
– **Augmented Tokenization with Phrases:** punctuations, extra spaces removed and extraction of phrases along with Punkt + PennTreeBank
– **Old Preprocessing:** Punkt + PennTreeBank only

to see which performs best.

4.3 Latent Semantic Analysis

Latent Semantic Analysis LSA is based on the mathematical concept of Singular Value Decomposition (SVD) where the term-document tf-idf matrix (X) is decomposed via SVD taking only the first only K dimensions into consideration.

$$X_K = U_K \Sigma_K V_K^T$$

where,
X_K refers to the approximate term-document matrix,
U_K refers to the term-concept matrix
Σ_K refers to the singular values(concept strengths) along the diagonal arranged in descending order,
V_K^T refers to the concept-document matrix

In this project, we explore 3 implementations of Latent Semantic Analysis:

Implementation-A: Query and document similarity are computed in the concept space
Implementation-1: Query and document similarity are computed in the original space but aided by concept space. Part 1: Query Dot Product
Implementation-2: Query and document similarity are computed in the original space but aided by concept space. Part 2: Query calculated as centroid of terms

These are explained in detail below.

Latent Semantic Implementation-A: The query vector is expressed in the concept space as follows,

$$\hat{\mathbf{q}}_j = \Sigma_K^{-1} U_K^T \mathbf{q}_j$$

where,
$\hat{\mathbf{q}}_j$ is the query vector in concept space
$\mathbf{q}_j$ is the query vector in term space

Similarly, the document vector can be expressed in concept space.

$$\hat{\mathbf{d}}_j = \Sigma_K^{-1} U_K^T \mathbf{d}_j$$

$\hat{\mathbf{d}}_j$ is the document vector in concept space
$\mathbf{q}_j$ is the document vector in term space

We then calculate the cosine similarity between a query and the document in the concept to find the relevant documents.

Latent Semantic Implementation - 1 We know that,

$$X_K = U_K \Sigma_K V_K^T$$

The cosine-similarity of the given query vector, (q) in the term space with the documents can be calculated as follows,

$$q^\top \cdot X_K = q^\top \cdot \left(U_K \cdot \Sigma_K \cdot V_k^T\right) = \left(q^\top \cdot U_K \cdot \Sigma_K\right) \cdot V_K^T$$

where $\left(q^\top \cdot U_K \cdot \Sigma_K\right)$ can be interpreted as the query mapping in the concept space.

Latent Semantic Implementation - 2 In the this implementation, query representation in concept space is calculated by taking the centroid of the representations of all the terms of the query in concept space.

Experiments To determine the best possible preprocessing and implementation of LSA and to determine the optimal K value, we run experiments for 9 cases

Case-1: LSA Implementation-A Augmented Tokenization
Case-2: LSA Implementation-A Augmented Tokenization With Phrases
Case-3: LSA Implementation-A Old Preprocessing
Case-4: LSA Implementation-1 Augmented Tokenization
Case-5: LSA Implementation-1 Augmented Tokenization With Phrases
Case-6: LSA Implementation-1 Old Preprocessing
Case-7: LSA Implementation-2 Augmented Tokenization
Case-8: LSA Implementation-2 Augmented Tokenization With Phrases
Case-9: LSA Implementation-2 Old Preprocessing

Determining Optimal K and Optimal Preprocessing K values are experimented with in the range of 300 to 1000 in steps of 50 To determine the optimal value of K and the best preprocessing, we plot the values of MAP@10 and nDCG@10 for all 9 cases, in Figure 5.

Using both MAP and nDCG curves, we can see that Implementation-1 is better than Implementation-2 as well as Implementaton-A since it has higher MAP and nDCG values for the same K. Further, we see that Augmentation with phrases performs better than both Augmentation and Old Preprocessing in terms of nDCG. Moreover, the variation in MAP for tokenization with phrases is lower than that of without phrases for the range of Ks considered. Thus, it can be concluded that Implementation-1 + Augmentation with Phrases performs the best. For this case, from experiments we observed that K=600 is optimal for Implementation-1 in terms having the maximum MAP@4 and at MAP@10. [MAP@4 is considered since we observe that among k = 1 to 10, maximum MAP occurs at k=3 or 4]. This is confirmed by plotting all evaluation metrics for all 9 cases for

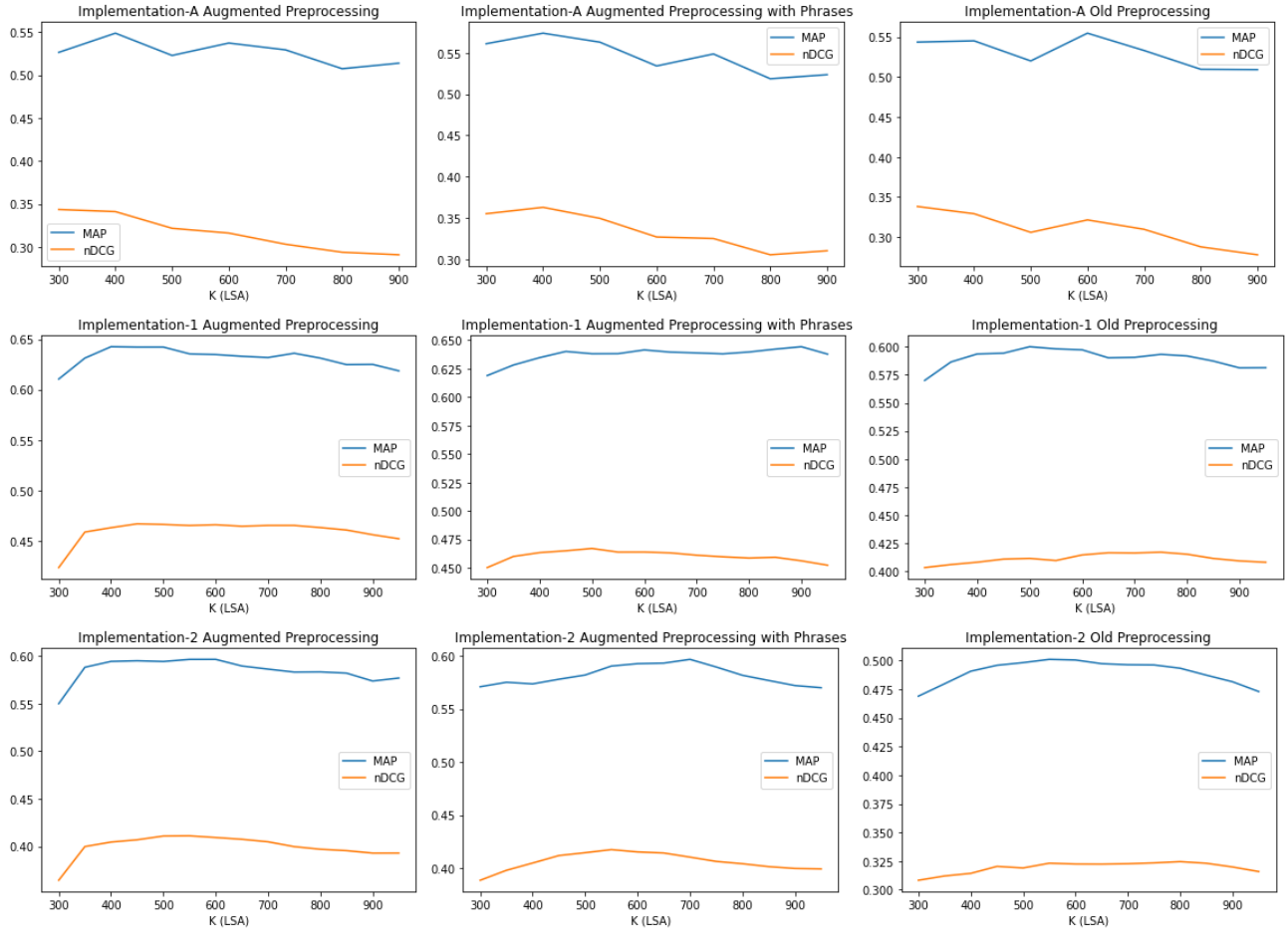

Fig. 5. Plots for MAP@10 and nDCG@10 for all 9 cases for K ranging from 300 to 900

K=300 to 1000 in steps of 50. Thus, further analysis and comparison of all algorithms is carried out using K = 600. Figure 6, shows the plots of all metrics for each of the 9 cases with K = 600 for ranks (k) from 1 to 10.

From the plots for K = 600, we confirm that Implementation-1 outperforms Implementation-1 and old preprocessing in both cases performs poorly. It appears that augmented preprocessing performs similar to augmented preprocessing with bigrams.

However, from Tables 3 and 4, we see the following: nDCG@10 for Augmented Tokenization with Phrases is 0.002 less than that of Augmented Tokenization whereas MAP@10 for Augmented Tokenization with Phrases is 0.01 more than that of Augmented Tokenization. Hence, K=600 with Augmented Tokenization with Phrases is taken as optimal and considered for further analysis.

From Tables 3 and 5, it is clear that Implementation-1 does far better than Implementation-A in terms of all the metrics at all ranks. Considering MAP@10 for Implementation-A and Implementation-1, we see that there is 11 % absolute difference in performance [A: 0.53, 1: 0.64]

Since Implementation-A computes similarity between documents and queries in the lower dimensional concept space, it can be said that the concepts being formed by LSA are not fully representing the essence of the queries/documents. This may be due to polysemous words being present in them.

Precision-Recall Curves for K = 600 for Each of the 9 Cases In Figure 7, in all 9 cases, we see that the area under precision-recall curves for Latent Semantic Analysis is higher than that of Vector Space Model which clearly indicates the superiority of LSA over VSM.

Evaluation Metrics for Optimal Case In this project, the following evalution metrics have been considered:

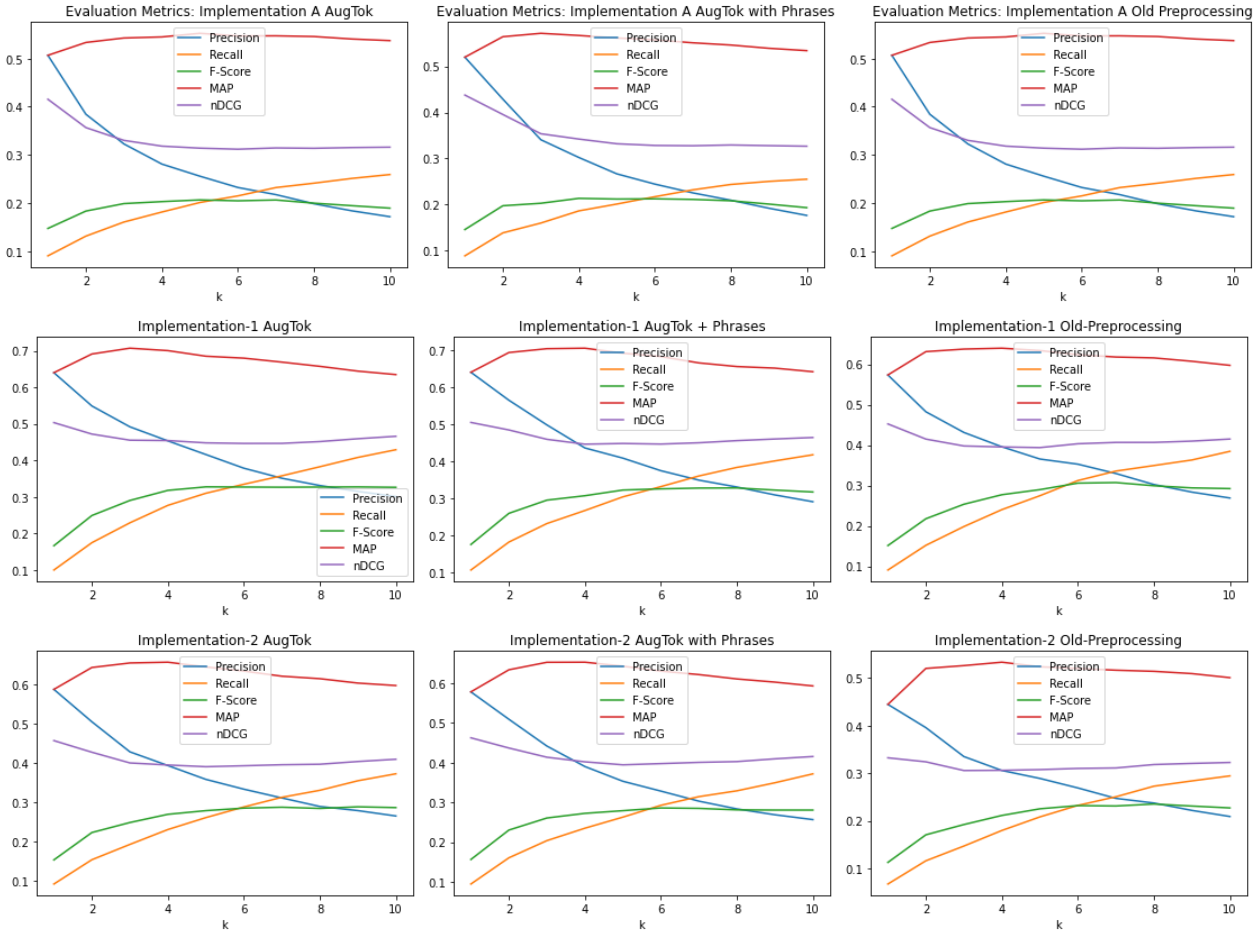

Fig. 6. Plots for all evaluation metrics for each of the 9 cases for K=600

- Precision@k
- Recall@k
- F-score@k
- nDCG@k
- MAP@k

MAP@k and nDCG@k are highly suited for search engine tasks. In our case, since relevance measurements are present, nDCG@k would have been an ideal choice. However, from all the plots above, we observe that every system has a poorer nDCG than MAP at all k. Further, the range of variation of nDCG is seen to be lower in our case. Thus, we give more preference to the MAP@k metric.

For k we consider all ranks until 10 and all ranks until 100. However more preference is given to MAP@10 since we assume that the user is most interested in the top-10 results and may not be willing to go through the top-100 results. The plots for the optimal case, i.e., Implementation-1 with Augmented Tokenization with Phrases and K=600 can be seen in Figure 8.

4.4 Explicit Semantic Analysis

From the LSA analysis we understand that augmented tokenization with phrases extracted works better. In ESA the concepts which are wikipedia articles, are more intuitive and since they are related to aerodynamics we expect articles with the phrases such as “boundary-layer” to be more relevant to an aerodynamics query than articles with just “boundary” or just “layer”. Hence, we proceed with augmented tokenization with phrases.

	Precision@k	Recall@k	Fscore@k	nDCG@k	MAP@k
1	0.640000	0.105724	0.174696	0.504444	0.640000
2	0.564444	0.181253	0.258860	0.484135	0.693333
3	0.497778	0.231316	0.294519	0.458853	0.703704
4	0.435556	0.266460	0.306471	0.445725	0.705062
5	0.408000	0.303437	0.321840	0.447648	0.691556
6	0.374074	0.331124	0.325077	0.446037	0.682417
7	0.348571	0.359788	0.327160	0.449535	0.665299
8	0.330000	0.383004	0.327510	0.455301	0.655450
9	0.308642	0.400545	0.321748	0.459691	0.651159
10	0.290222	0.417082	0.316548	0.463760	0.641464

Table 3. Values of all metrics for k = 1 to 10 for K = 600 for Implementation 1 Augmented Tokenization with Phrases

	Precision@k	Recall@k	Fscore@k	nDCG@k	MAP@k
1	0.640000	0.099860	0.166439	0.503333	0.640000
2	0.548889	0.174737	0.249068	0.471980	0.691111
3	0.491852	0.229089	0.290265	0.455249	0.707037
4	0.453333	0.276858	0.317921	0.454219	0.700370
5	0.416000	0.309996	0.327497	0.447914	0.685049
6	0.378519	0.334634	0.326999	0.446347	0.679767
7	0.351111	0.357736	0.326502	0.446512	0.668879
8	0.331111	0.382564	0.326970	0.451452	0.657336
9	0.315062	0.408042	0.327315	0.459191	0.644239
10	0.300889	0.429127	0.326300	0.465942	0.634765

Table 4. Values of all metrics for k = 1 to 10 for K = 600 for Implementation 1 Augmented Tokenization

Procedure

1. To obtain the articles PetScan was used with “Aerodynamics” as the topic. Setting depth = 1 we get 500 articles, setting depth = 3 we get 3343 articles.
2. After obtaining the names of articles, wikipedia articles corresponding to these names are extracted. For each article, the content of the page along with links and backlinks are combined to form the body of the article.
3. This body is then fed through the preprocessing pipeline. Following this a term-article matrix is generated. The row headings are the terms or phrases present in all the articles and the column headings are the names of the articles, which form concepts in ESA. The entries in this matrix are the TF-IDF values.
4. Following this, all documents and queries are projected to the article space. For this, all tokens in the document/query and their count of occurrences were calculated. Then for each term/phrase their corresponding tf-idf vector was looked up from the term-article matrix and weighted according to the number of occurrences in that document/query. This allows us to represent each document and query as a vector in the article space. If a termi in a doc or query is not present in the term-article matrix it is ignored. If no terms

	Precision@k	Recall@k	Fscore@k	nDCG@k	MAP@k
1	0.520000	0.088631	0.145729	0.437778	0.520000
2	0.428889	0.138550	0.197448	0.395580	0.564444
3	0.340741	0.159884	0.202837	0.353779	0.571852
4	0.302222	0.186267	0.213474	0.342221	0.567160
5	0.266667	0.201135	0.211993	0.332027	0.561660
6	0.244444	0.217034	0.212456	0.328370	0.557558
7	0.225397	0.231951	0.211178	0.327721	0.551247
8	0.209444	0.243638	0.208286	0.329368	0.546383
9	0.192099	0.250307	0.201073	0.327941	0.539316
10	0.176444	0.255041	0.193280	0.326699	0.534199

Table 5. Values of all metrics for k = 1 to 10 for K = 600 for Implementation A Augmented Tokenization with Phrases

in a doc or query match with terms in the term-article matrix, the doc or query is represented as a zero vector in the article space.

5. To get the relevant documents for each query, we take the cosine similarity of the query with all documents and rank them in descending order of similarity.

Experiment-1: ESA with 500 Articles Examples from the 500 articles used can be seen in Table 6.

Aerodynamics	Standard conditions for temperature and pressure
Anemometer	Wing
Fluid dynamics	Area rule
Jet engine	Thrust
Lift (force)	Wind tunnel
Laminar flow	Windmill
Mach number	Ames Research Center

Table 6. Examples of articles from the set of 500 articles used

Experiment-2: ESA with 3343 Articles Examples from the 3343 articles used:

In Table 8 , we see an example of the articles in this set of 3343 articles, obtained from PetScan with the topic as Aerodynamics and depth = 3. These do not appear to be very relevant to the topic of Aerodynamics. For example, 'Overview of gun laws by nation' and 'gunpowder' are concepts that do not occur in the Cranfield Documents.

4.5 Evaluation Metrics for both Experiments

From Figure 12, it can be observed that the area under the P-R curve for ESA-500 is lesser than the area under the P-R curve for ESA-3343. Thus, we can say that ESA-3343 outperforms ESA-500.

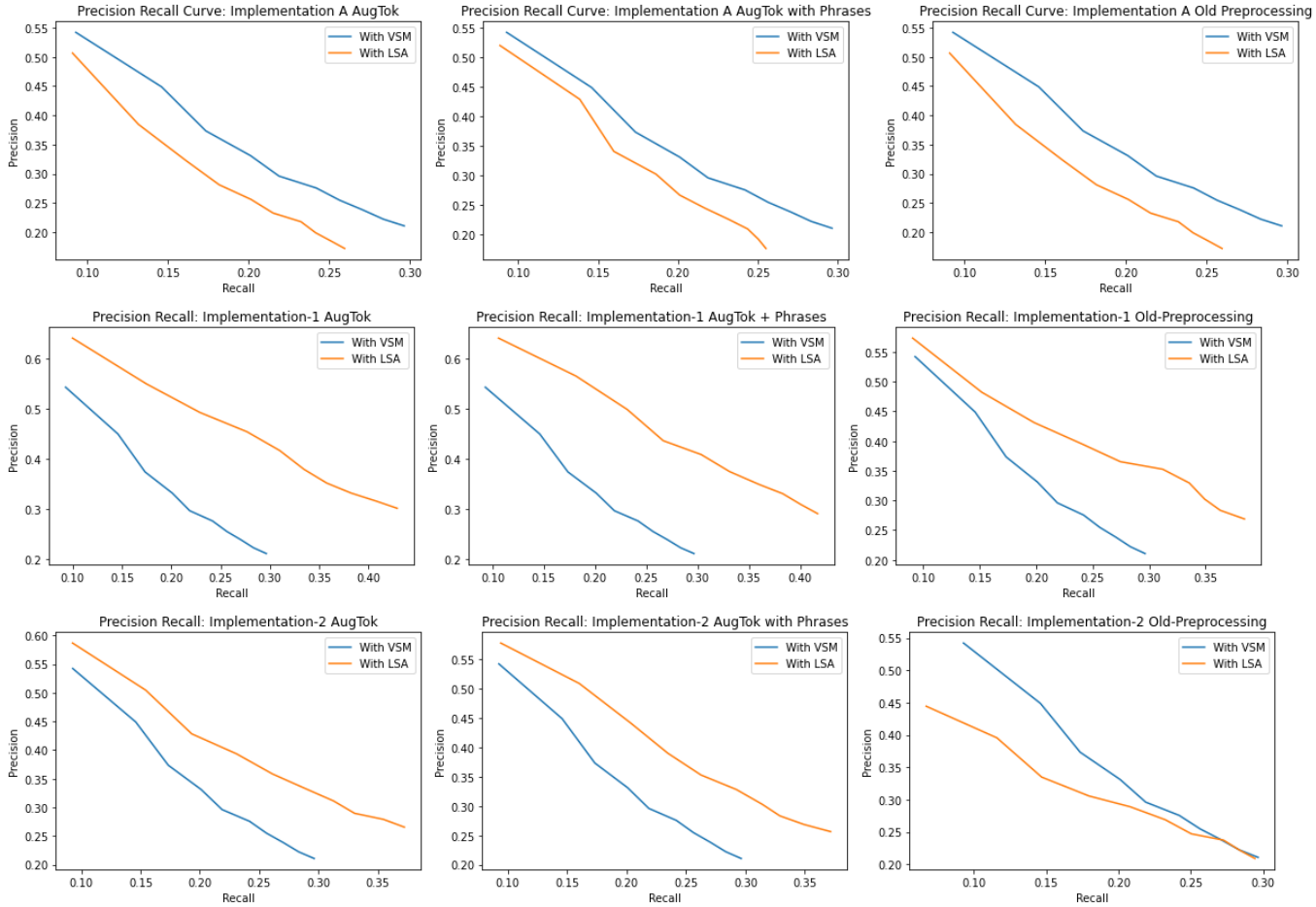

Fig. 7. Precision-Recall Curves for K = 600 for each of the 9 Cases

5 Results and Analysis

From Table 10 we see that among all the LSA-A implementations, AugTok and Old Preprocessing have the same values for all metrics at rank 10, while AugTok with phrases performs better than the others with respect to all metrics. This tells us that (a) removing punctuations and spaces in this case did not enhance concept formation (b) using phrases as basic units like terms in LSA enhances concept formation and helps in bringing out concepts that capture the essence of the queries and documents.

Among all LSA-2 implementations we see that all metrics apart from nDCG@10 are better for AugTok without phrases, while nDCG@10 is higher for AugTok with Phrases. From this we deduce that the usage of phrases helps in retrieving documents of higher relevance in the top ranks.

Among the 2 ESA implementations, it is observed that the one with 3343 articles (more background knowledge) fares better than the one with only 500 articles with respect to all metrics at rank 10, even though in the 3343 article set, the new articles (apart from the old 500) introduced many seemingly irrelevant concepts.

Further, as discussed earlier, nDCG@10 for LSA Implementation-1 Augmented Tokenization with Phrases is 0.002 less than that of Augmented Tokenization whereas MAP@10 for Augmented Tokenization with Phrases is 0.01 more than that of Augmented Tokenization. As discussed in Section 4.3.7, more preference is given to MAP.

Thus, finally, LSA Implementation-1 with Augmented Tokenization + Phrases is better than all the other algorithms in the above mentioned table, at Information Retrieval on Cranfield dataset with respect to evaluation metric MAP@10 under the assumptions that a) phrases detected using PPMI represent concepts better than the individual terms forming the phrase b) terms in the documents are not polysemous c) there are no spelling errors that are dictionary words (only non-dictionary spelling errors).

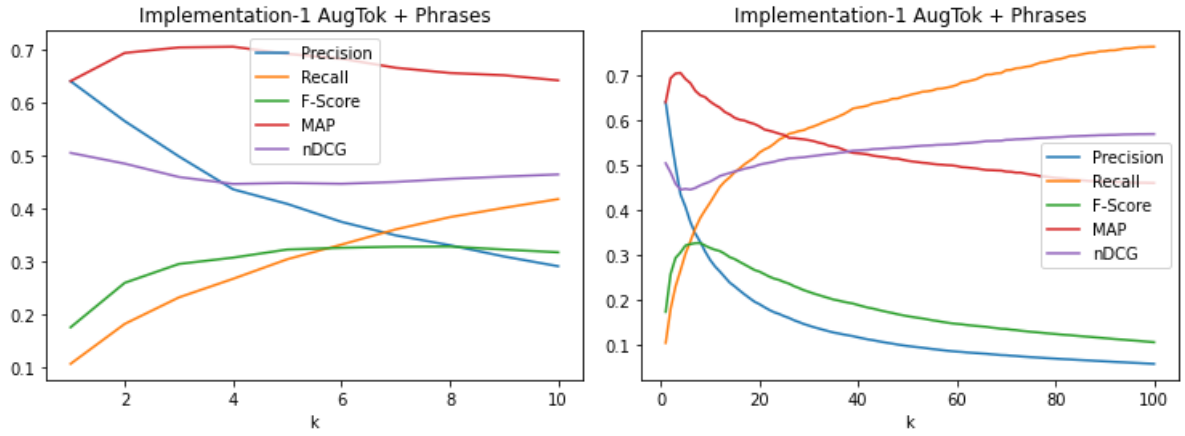

Fig. 8. Evaluation Metrics Plot for (a) k=1-10 (b) k=1 to 100 or Implementation-1 AugTok+Phrases K=600

	Aerodynamics	Anemometer	Fluid dynamics	Jet engine	Lift (force)	Laminar flow	Mach number
aerodynamic	33.444298	1.320170	6.160792	2.200283	8.361075	0.880113	2.640339
greek	18.741394	15.617828	3.123566	6.247131	0.000000	0.000000	0.000000
άήρ	5.521461	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
aero	7.223837	0.000000	0.000000	16.855619	0.000000	0.000000	0.000000
air	16.253217	7.170537	32.984470	37.764828	38.242864	5.736430	14.819110

Fig. 9. Portion of TF-IDF Term Article Matrix used for ESA with 500 Articles

5.1 Analysis of LSA vs VSM

In this section, we compare the results of LSA based and basic VSM based IR systems. We deep-dive into the query-wise performance of the two models for 3 cases: Queries that LSA performed well on but VSM performed poorly Queries that VSM performed well on but LSA performed poorly Queries on which both LSA and VSM performed poorly We quantify the performance by finding the following metrics at k=10 for each query: Precision@k, Recall@k, F-Score@k, nDCG@k, AveragePrecision@k for each of the models. For each model, a list of queries that the model performed well as well as poorly on is collated. Poor performance is characterised by all the metrics mentioned above being zero for that query, while good performance is identified by nDCG@10>0.5 and AP@k>0.5 for the query.

Case 1: Queries that LSA performed well on but VSM performed poorly For each query id, top documents predicted by each of the methods (LSA and VSM) and respective ground truths for that query are provided

Query-ID 9

papers on internal /slip flow/ heat transfer studies

1. LSA Predicted: 22, 21, 550, 571
2. VSM Predicted: 102 846 45
3. Ground Truth: 534, 21, 22, 550

The top-4 documents predicted by LSA have the following titles:

	Precision@k	Recall@k	Fscore@k	nDCG@k	MAP@k
1	0.235556	0.039540	0.065010	0.175556	0.235556
2	0.140000	0.044425	0.063266	0.133281	0.246667
3	0.128889	0.063222	0.078737	0.131228	0.262963
4	0.111111	0.074379	0.082349	0.126826	0.269506
5	0.101333	0.081794	0.083265	0.125163	0.267691
6	0.097037	0.090343	0.085951	0.126827	0.268599
7	0.092063	0.099381	0.088088	0.128179	0.270538
8	0.087778	0.106879	0.089129	0.130819	0.267365
9	0.082469	0.114358	0.088584	0.132737	0.264957
10	0.079556	0.121406	0.088884	0.135696	0.262832

Table 7. Table of all metrics for k=1 to 10 for ESA with 500 Articles

Aerodynamics	Kite
Anemometer	Faster-than-light
Fluid dynamics	Frederick Abel
BASE jumping	Firearm
Extravehicular activity	Overview of gun laws by nation
Fin	Gunpowder

Table 8. Example of articles in set of 3343 articles

1. 21: heat transfer in slip flow
2. 22: slip-flow heat transfer to a flat plate
3. 550: laminar heat transfer in tubes under slip-flow conditions
4. 571: heat transfer to flat plate in high temperature rarefied ultra-high mach number flow

Here we see that the augmented tokenization of the docs and queries in specifically the removal of slashes and hyphens has resulted in retrieval of the relevant documents which was not the case in our previous VSM model. This clearly establishes the importance of the Augmented Tokenization procedure. Upon careful observation of the retrieved documents by LSA, it is likely that the concepts such as 'heat transfer' and 'slip flow' are extracted which has led to the efficient retrieval of the documents when similarity is calculated upon expressing both the document and query in concept space. This would not be possible with VSM as it is mainly based on keyword matching. Further, the fact that we used the extracted phrases such as "heat transfer" as terms in the term-document matrix may have aided in the concept formation of heat transfer.

While 3 of the 4 ground truths have been predicted by LSA in the top-4, one of the documents, with ID 534: consideration of energy separation for laminar slip flow in a circular tube, was not retrieved in the top-10. The possible reason for this may be the absence of the concept heat transfer or other concepts like slip flow with low concept strengths, i.e., small singular values after SVD decomposition.

Further, we see that the VSM model predicts documents such as 846: the vibration of thin cylindrical shells under internal pressure which is purely based on keyword(internal) matching and the retrieved document is not relevant.

Case 2: Queries that VSM performed well on but LSA performed poorly Queries which provided good results (AP $\sim$ 1) with VSM and significantly poorer results (nDCG < 0.1 and AP < 0.1) with LSA were not found.

	Apollo program	Apollo 12	Apollo 14	Apollo 15	Apollo 16	Apollo 17	Aeronautics
special	5.353112	0.000000	0.000000	3.568741	0.000000	0.000000	1.784371
message	8.385266	5.590177	5.590177	5.590177	8.385266	5.590177	0.000000
urgent	5.350376	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
need	6.465855	2.586342	5.172684	9.052197	0.000000	2.586342	1.293171
now	3.527845	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

Fig. 10. Portion of TF-IDF Term Article Matrix used for ESA with 500 Articles

	Precision@k	Recall@k	Fscore@k	nDCG@k	MAP@k
1	0.302222	0.046971	0.078298	0.218889	0.302222
2	0.262222	0.080847	0.116169	0.220840	0.364444
3	0.222222	0.099069	0.127425	0.205461	0.381111
4	0.203333	0.121218	0.140337	0.205832	0.385062
5	0.187556	0.137775	0.146593	0.203925	0.387210
6	0.169630	0.148475	0.146159	0.201293	0.388381
7	0.157460	0.161733	0.147361	0.203171	0.378976
8	0.152222	0.177466	0.151363	0.208390	0.372218
9	0.141728	0.186576	0.148877	0.209784	0.371400
10	0.134667	0.198890	0.148697	0.213318	0.365541

Table 9. Table of all metrics for k=1 to 10 for ESA with 500 Articles

Case 3: Queries on which both LSA and VSM performed poorly In these cases both LSA and VSM have given undesirable results but possibly due to different reasons.

Query-ID 207

how do large changes in new mass ratio quantitatively affect wing-flutter boundaries

1. LSA Predicted: 433, 1062, 924
2. VSM Predicted: 365 1185 655
3. Ground Truth: 1290, 1338, 1339, 1340, 1341, 879

The documents predicted by LSA have the following IDs and titles:

- 1062: an experimental and theoretical investigation of second-order wing-body interference at high mach number
- 924: a method for calculating the lift and centre of pressure of wing-body-tail combinations at subsonic, transonic speeds

The above documents may have been retrieved by LSA due to the presence of concepts representing 'wing body' + 'tail' + 'lift' etc. It is also important to note that the document, containing terms semantically similar to those in the query, i.e. large, such as 'high' have also been retrieved. But the ground truth documents include, '...measured and calculated subsonic and transonic flutter characteristics ... calculation of flutter characteristics for finite-span swept unswept wings. . . .' etc.

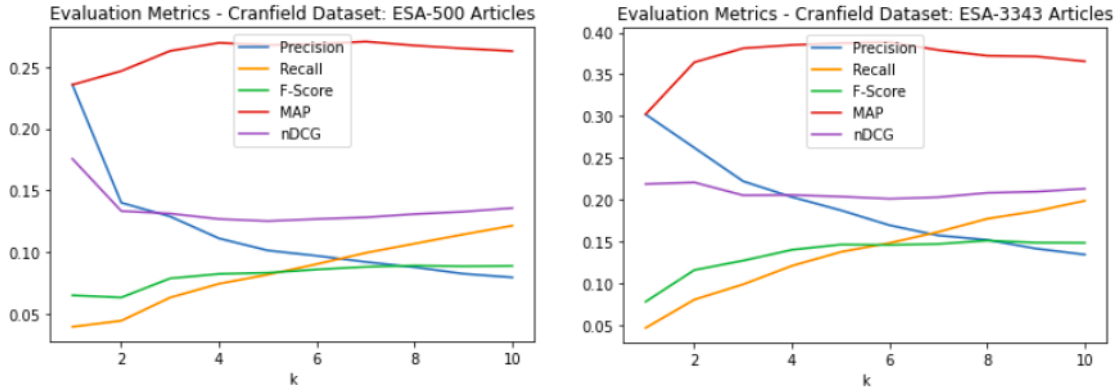

Fig. 11. Plot of all metrics for k=1 to 10 for ESA with (a) 500 articles (b) 3343 articles

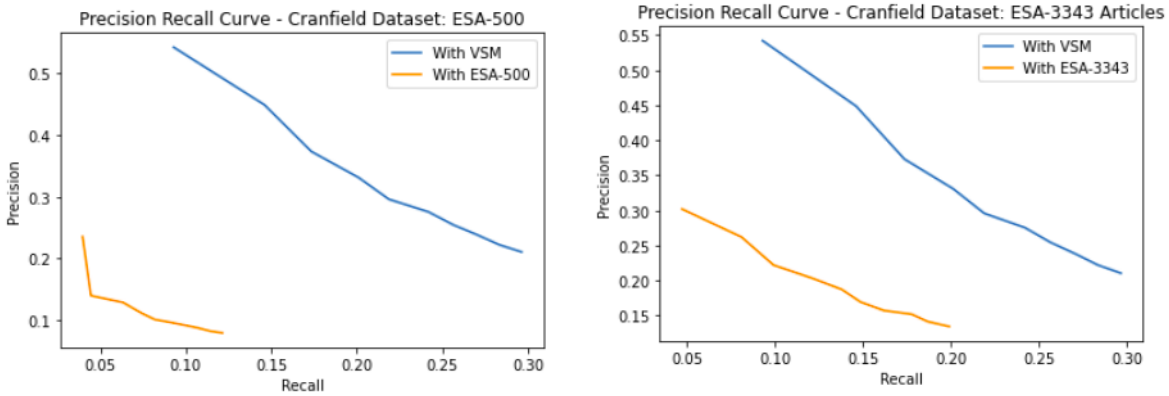

Fig. 12. Precision-Recall Curves for k=1 to 10 for ESA with (a) 500 articles (b) 3343 articles

Though these documents contain the term 'wing' in them, the term 'flutter' may not be well incorporated, i.e., may have a small component, in the concept containing the 'wing body' + 'lift' etc.

The documents predicted by LSA have the following IDs and titles:

- 365: the homogeneous boundary layer at an axisymmetric stagnation point with large rates of injection
- 1185: '....one pair connects the surface mass transfer rate and surface concentration of injected gas ...' , '...in the presence of mass transfer .....'

The above documents may have been retrieved by VSM due to the presence of keywords such as 'large' and 'mass' in the documents, even though they do not not cater to the context.

5.2 Analysis of ESA vs VSM

Case 1: Queries that ESA performed well on but VSM performed poorly

Query-ID 88

how does a satellite orbit contract under the action of air drag in an atmosphere in which the scale height varies with altitude

1. ESA Predicted: 617, 615, 483
2. Top-3 wikipedia articles- 'Drag (physics)', 'Ventilation (architecture)', 'External ballistics'
3. VSM Predicted: 162, 449, 314
4. Ground Truth: 613, 614, 615, 616, 617, 618, 548

Using K = 600	Precision@10	Recall@10	F-score@10	nDCG@10	MAP@10
LSA - A AugTok	0.172000	0.259445	0.189747	0.316151	0.537161
LSA - A AugTok +Phrases	0.176444	0.255041	0.193280	0.326699	0.534199
LSA - A Old Preprocessing	0.172000	0.259445	0.189747	0.316151	0.537161
LSA - 1 AugTok	**0.300889**	**0.429127**	**0.326300**	**0.465942**	0.634765
LSA -1 AugTok +Phrases	0.29022	0.417082	0.316548	0.463760	**0.64146**
LSA -1 Old Preprocessing	0.268889	0.384412	0.292167	0.414756	0.597179
LSA - 2 AugTok	0.265333	0.372482	0.286636	0.409327	0.596444
LSA - 2 AugTok +Phrases	0.256444	0.371558	0.280402	0.415171	0.592721
LSA -2 Old Preprocessing	0.208889	0.294219	0.226925	0.322411	0.500575
ESA-500 Articles	0.079556	0.121406	0.088884	0.135696	0.262832
ESA-3343 Articles	0.134667	0.198890	0.148697	0.213318	0.365541

Table 10. Table of all evaluation metrics at Rank-10 for all Algorithms and their variants

Top-3 wikipedia articles considered in the analysis are the ones with the highest TF-IDF values in the query vector mapped in the concept space, i.e., the query has the largest projections in these dimensions.

The top-3 documents retrieved by ESA have the following titles:

- 617: determination of upper-atmosphere air density profile from satellite observations
- 615: the contraction of satellite orbits under the influence of air drag
- 483: stagnation point shock detachment distance for flow around spheres and cylinder

We observe that since the query when expressed in article space mainly corresponds to the articles with titles 'Drag (physics)', 'Ventilation (architecture)', 'External ballistics', the documents retrieved also correspond to these articles. Therefore this background knowledge helped in the retrieval of the relevant documents.

The top-3 documents retrieved by VSM have the following titles:

- 162: nearly circular transfer trajectories for descending satellites
- 449: interaction of a charged satellite with the ionosphere
- 314: simplified method for determination of the critical height of distributed roughness particles for boundary layer transition at mach numbers from 0 to 5.

We see that VSM has retrieved documents just by keyword matching like satellite, height etc. which are not relevant.

Case 2: Queries that VSM performed well on but ESA performed poorly

Query-ID 128

has anyone programmed a pump design method for a high-speed digital computer

1. ESA Predicted: 543, 1360, 1293
2. Top-3 wikipedia articles- 'Speed of sound' , 'Sweep theory', 'Jet engine'
3. VSM Predicted: 945, 1063, 988

4. Ground Truth: 985, 990, 945

The top-3 documents predicted by ESA have the following IDs and titles:

- 543: the stacking of compressor stage characteristics to give an overall compressor performance map
- 1360: simplified analysis of general instability of stiffened shells in pure bending
- 1293: design of stiffened cylinders in axial compression

As only 500 articles on aerodynamics were used as background knowledge, lack of enough background knowledge could be the reason for the misinterpreting the query as relating to the articles on 'jet engine', 'speed of sound' due to the presence of the keyword 'speed' and 'pump'. Further, due to the detection of 'Jet Engine' as an important article, documents with the word compressor may have been detected as relevant.

Since, no articles on 'impellers' or 'pump design' are a part of the background knowledge, most of the terms were either ignored or mapped to irrelevant articles, thus resulting in improper retrieval.

The top-3 documents predicted by ESA have the following IDs and titles:

- 945: method for design of pump impellers using a high speed digital computer
- 1063: on obtaining solutions to the navier-stokes equations with high speed digital computers
- 988: nonviscous flow through a pump impeller on a blade to-blade surface of revolution

VSM has provided relevant retrieved upon matching the query vector containing keywords such as pump, design and computer with the document vectors expressed in term space, since in this case, the keywords that contain the essence of the query, i.e., 'pump design', 'high speed digital computer' are directly available in the documents.

The ground truth:

- 945: method for design of pump impellers using a high speed digital computer
- 985: a rapid approximate method for the design of hub shroud profiles of centrifugal impellers of given blade shape
- 990: a rapid approximate method for determining velocity distribution on impeller blades of centrifugal compressors

Case 3: Queries on which both ESA and VSM performed poorly

Query-ID - 141

what analytical solutions are available for stresses in edge-loaded shells of revolution

1. ESA Predicted: 424, 736, 735
2. Top-3 wikipedia articles- 'Navier–Stokes equations' 'Wind power' 'Generation on the Wind'
3. VSM Predicted: 930, 890, 889

The top-3 documents retrieved by ESA have the following IDs and titles:

- 424: cantilever plate with concentrated edge load
- 736: the bending of a wedge shaped plate
- 735: the bending of uniformly loaded sectorial plates with clamped edges

The main idea being communicated in the query is the need for solutions to calculate stresses in revolving shells with weights on the edges. Among the ESA retrieved docs, in 424 we see: "....the method of finite differences.....cantilever plate which bears a concentrated load at the longitudinal free edge". Thus, ESA gets the overarching concept behind the query right, i.e., it provides a method of solution for stresses, as well as takes care of the edge-loaded criteria. However, it gets the object wrong, i.e., it provides the solution for a a thin plate instead of a shell. This clearly tells us that, ESA is successful in grasping concepts and semantic similarity beyond what is given in the documents. As far as the top

The top-3 documents retrieved by VSM have the following IDs and titles:

- 930: general theory of large deflections of thin shells with special applications to conical shells
- 890: comments on 'thermal buckling of clamped cylindrical shells

– 889: a simplified method of elastic stability analysis for thin cylindrical shells

Among the VSM retrieved docs, in 930, we see that the document talks about the theory behind deflections in conical shells and not about how to solve for the stresses in the shell, which was the intent behind the query. Further, the document mentions "A general theory is developed for the case of large deflections but with rotations of the elements negligible compared to unity". The query specifically asks for revolving/rotating shells, but the document contains theory for a case of no rotation. Further, the document mentions a solution for the calculation of deflection, "the problem can be reduced to the solution of two fourth-order partial differential equations in a stress function and the deflection normal to the shell." The document provides the 'solution' for deflection using a certain 'stress' function while the query wants the 'solution' to calculate 'stress'. The two keywords get mapped out of context by VSM and hence the document gets retrieved.

Ground Truth:

– 954: analysis of stress at several junctions in pressurized shells
– 1042: on transverse vibrations of thin, shallow elastic shells
– 1039: on transverse vibrations of thin, shallow elastic shells

In case of Document 954, the algorithms may grasped the "part of" relation of 'edges' and 'several junctions', i.e., when several junctions are being analysed for stress, edge analysis would also be included.

6 Conclusion and Future Direction

LSA Implementation-1 with Augmented Tokenization + Phrases is better than all the other algorithms at Information Retrieval on Cranfield dataset with respect to evaluation metric MAP@10 under the assumptions that a) phrases detected using NPMI represent concepts better than the individual terms forming the phrase b) terms in the documents are not polysemous c) there are no spelling errors that are dictionary words (only non-dictionary spelling errors)

Spell check can be added to the tokenization pipeline which could not be added due to computational resource constraints. ESA with larger relevant background knowledge can provide significantly better results. However this could not be carried out due to resource constraints since the number of terms increases largely with increasing the number of articles, requiring systems with greater RAM to store and perform computations on the matrices.