CS6370: Natural Language Processing

# Project Report

Shashank Patil CH18B022
Shania Mitra CH18B067

---

## 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. Information retrieval 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 degrees of relevance ranging from 1 to 4, 4 being the most relevant. 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.

## Background and Related Work:

- ### Vector Space Model

  **Vector space model** is an algebraic model for rpresenting 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,

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

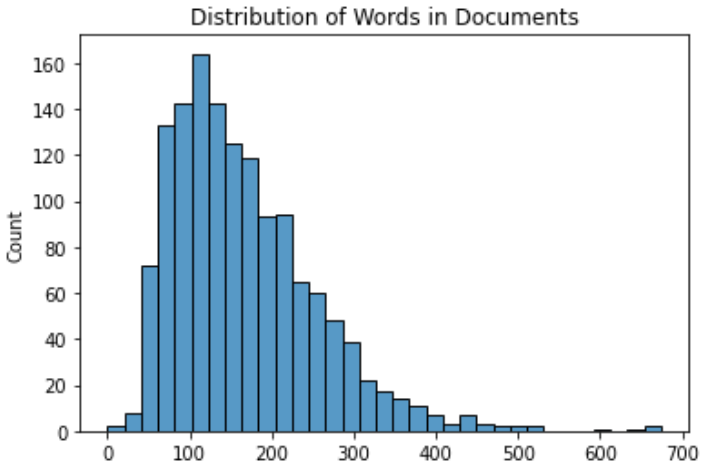
  - $\mathrm{tf}_{t,d}$ is term frequency of term *t* in document *d* (a local parameter)
  - $\log \frac{|D|}{|\{d' \in D \,|\, t \in d'\}|}$ is inverse document frequency (a global parameter).

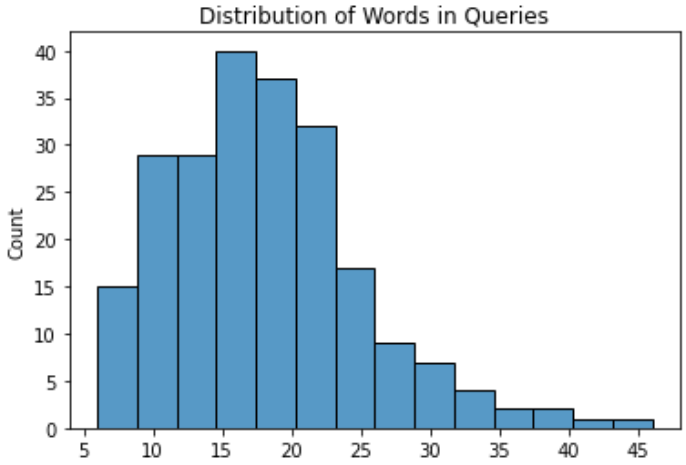
$|D|$ is the total number of documents in the document set; $|\{d' \in D \mid t \in d'\}|$

is the number of documents containing the term *t*.

- Exploration of Cranfield Dataset:

The cranfield dataset contains 225 queries and 1400 documents on Aerodynamics. Each query has a list of relevant documents with their corresponding degrees of relevance.




As can be seen in the above figure, the documents in the cranfield dataset have large variation in the number of words i.e., 0 to 700. Two documents - 471 and 995 are empty strings with no words.
Number of words in queries also ranges from 6 to 50.

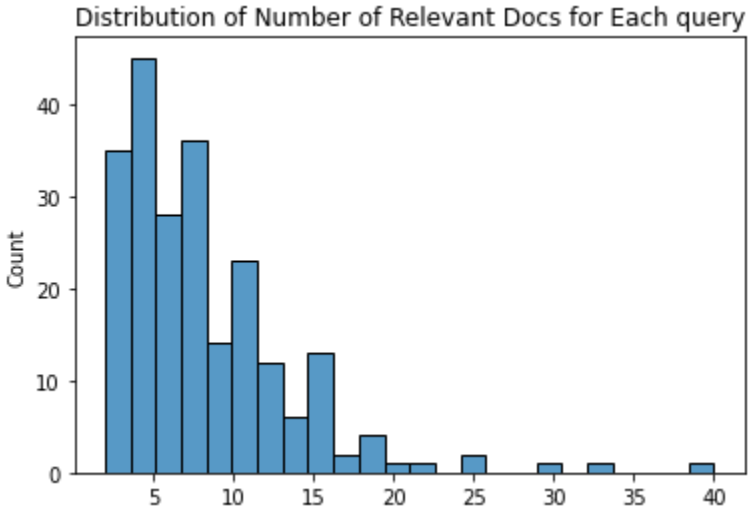
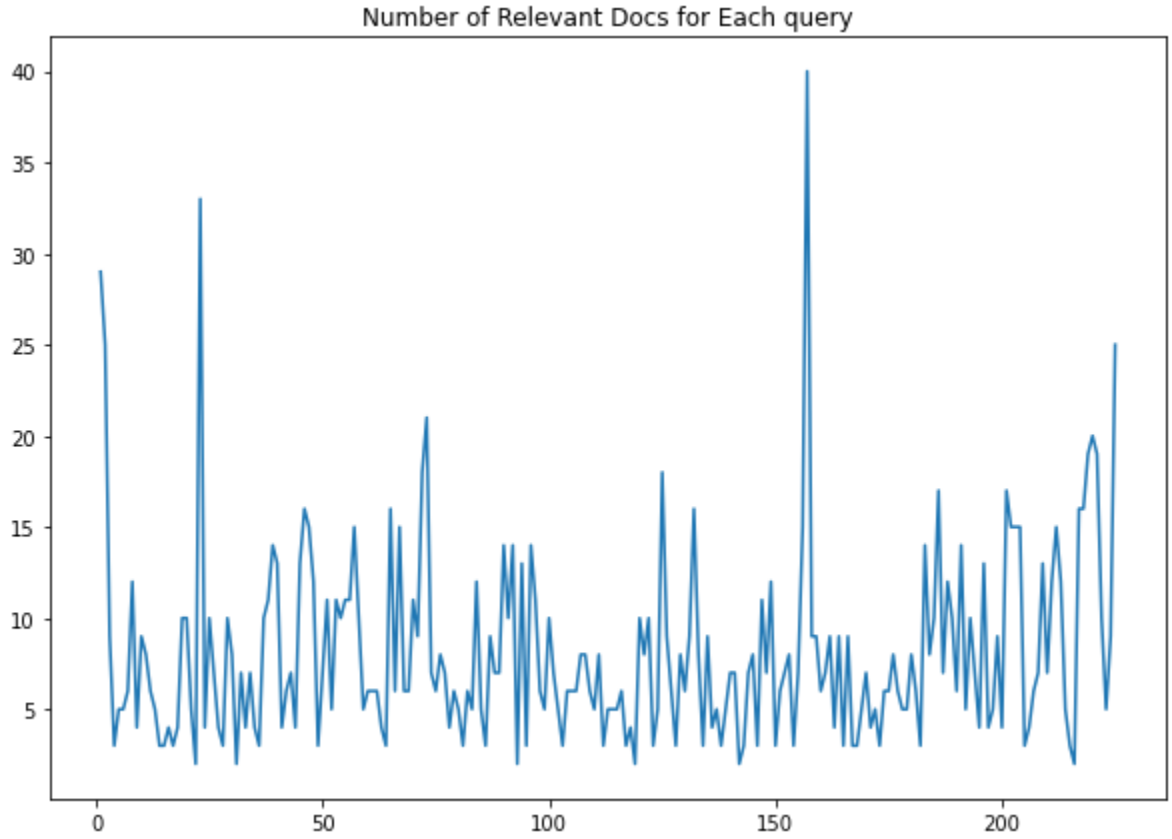
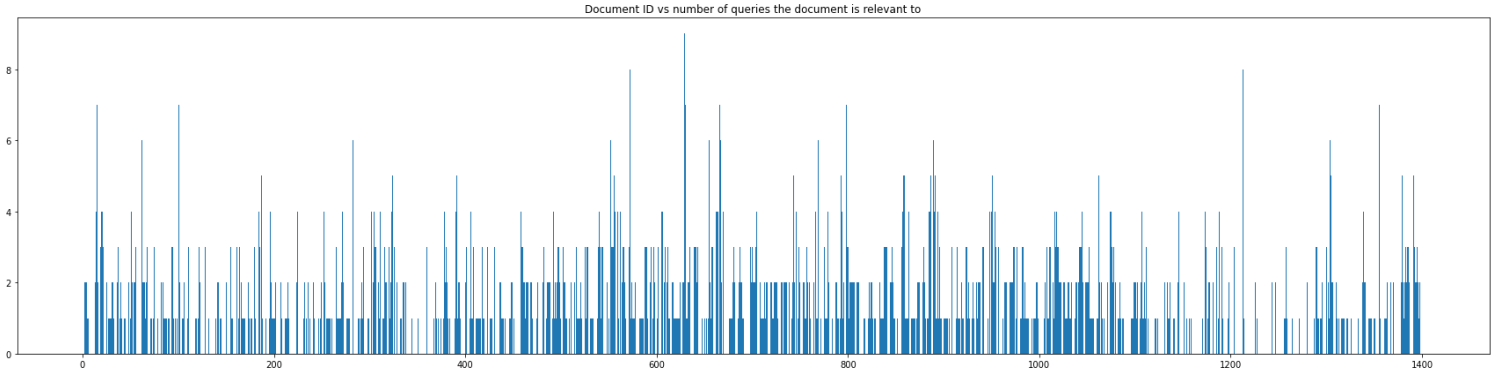


Figure 2

On analysing the ground truth values, we see that most of the queries have around 5-10 relevant documents while some generic queries have 40 relevant documents.




The above figure tells us that some queries have larger number of relevant queries than others




The above figure 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.

| token | frequency |
|---|---|
| affinely | 1 |
| lardnert.j | 1 |
| doylem.d.c | 1 |
| east | 1 |
| sellsc.c.l | 1 |
| gothic | 1 |
| woodleyj.g | 1 |
| electroform | 1 |
| catheralld | 1 |
| ob | 1 |

| token | frequency |
|---|---|
| flow | 2138 |
| pressure | 1387 |
| number | 1358 |
| boundary | 1244 |
| layer | 1190 |
| result | 1079 |
| effect | 962 |
| method | 909 |
| theory | 900 |
| body | 876 |

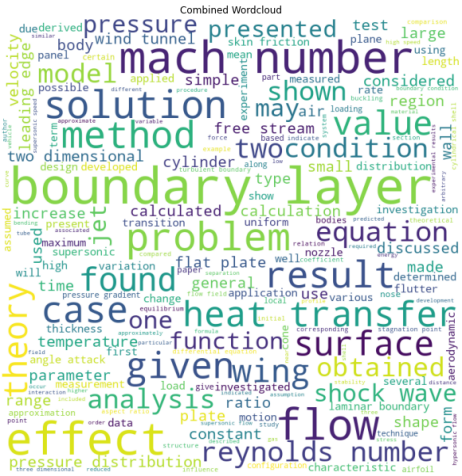
Figure: (a) Top-10 most rarely occurring words in the documents (b) Top-10 most frequently occurring words in the documents

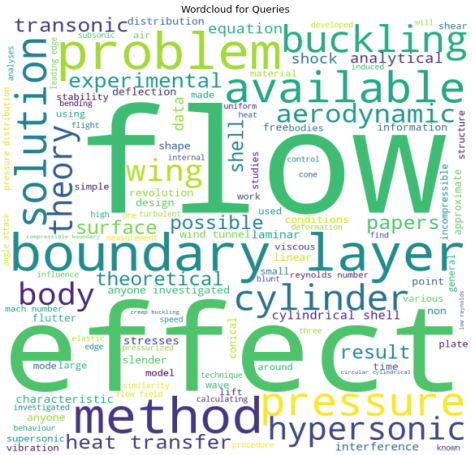
There are a total of 3402 words that occur only once among all documents. While most of these words are misspelt and can be used as candidates for misspelled words, many of the words are correctly spelled and are still present only in a single document. Hence, this method of spell check cannot be carried out.

Words that occur once, but are correctly spelt (examples). A total of 3402 words occur only once

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

WordClouds indicating dominant terms in documents and queries:





## Motivation

- Extracting Phrases

All documents belong to the domain of aerodynamics, thus, phrases such as “boundary layer”, “incompressible flow”, etc. must be extracted and given weightage, since these form real concepts in theory. PPMI (Positive Pointwise Mutual Information) used to extract phrases such as “shock induced”, “thermal stress”, “hydrostatic pressure”, “heat conduction”, “Reynolds number”, etc. Countings of occurrences and Co-occurrences of words in a text corpus can be used to approximate the probabilities appearing in the PPMI calculation.

- LSA

The vector space model does not take word-relatedness and overarching concepts into account, performing only keyword matches which results in poor performance.
It 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.

- 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.

## Methodology and Experiments

- Augmentation of Tokenization

  As highlighted in the proposal,
  ● 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

To improve upon this, functions were introduced to remove extra spaces, all punctuations, extra full stops and split across hyphens. Further two methods of spell-check were implemented but could not be incorporated into the final pipeline due to the lack of computational resources.

Further, 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.

We test our models with all three kinds of preprocessing -

- Augmented Tokenization
- Augmented Tokenization with phrases
- Old preprocessing (Punkt + PennTreeBank only), to see which performs best.

## ● Phrase Extraction

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

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

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.

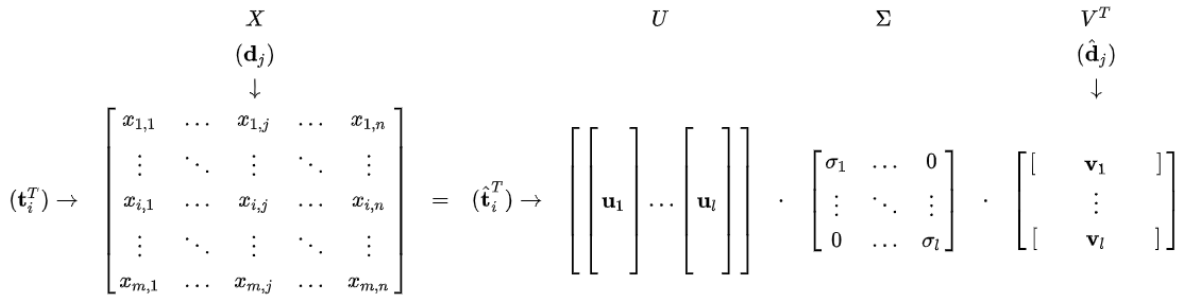
$$\operatorname{npmi}(x;y) = \frac{\operatorname{pmi}(x;y)}{h(x,y)}$$

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

Examples of phrases extracted:
'Shock_wave', 'boundary_layer', 'laminar_boundary', 'skin_friction', 'aerodynamic_heat', 'heat_conduction', ‘jet_propulsion’, etc.

- LSA

LSA is based on the mathematical foundation of Singular Value Decomposition(SVD)
SVD of the derm-document tf-idf matrix will give the following,



After considering a lower dimensional space, let's say k,

The term document can be expressed in the lower-dimensional space. We write this approximation as:

$$X_k = U_k \Sigma_k V_k^T$$

Where Uk can be interpreted as the term-concept matrix, sigma k as the concept strengths and Vk refers to the concept-document matrix.

We experimented with two different implementations of LSA where the main difference is how the representation of the query is calculated in the concept space.

In the first implementation, query is calculated using the the formula:

$$\hat{\mathbf{q}} = \Sigma_k^{-1} U_k^T \mathbf{q}$$

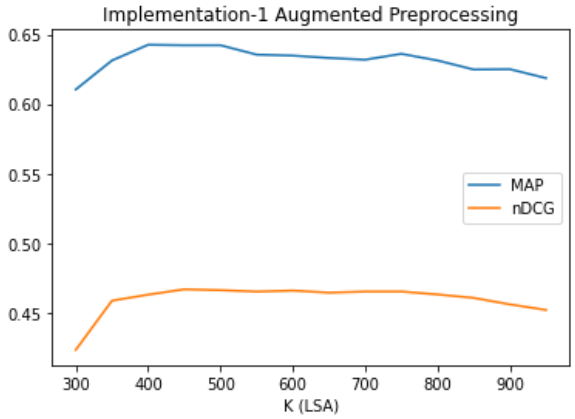
,

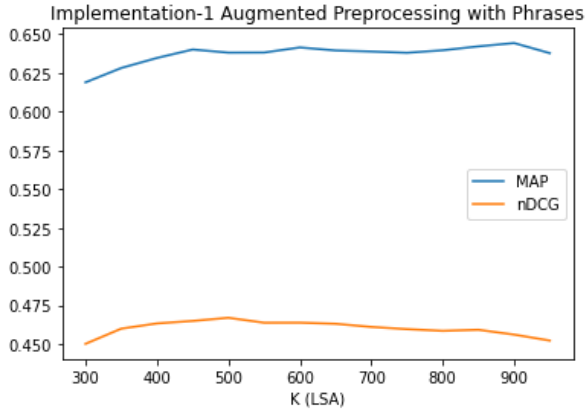
Where q_cap is the representation in the vector space.

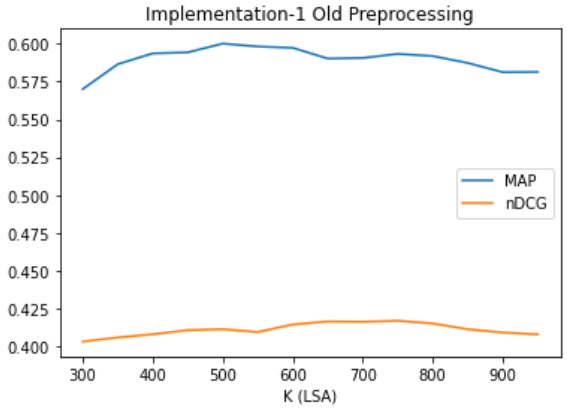
In the second implementation, query representation in concept space is calculated by taking the centroid of the representations of all the terms of the query in concept space.

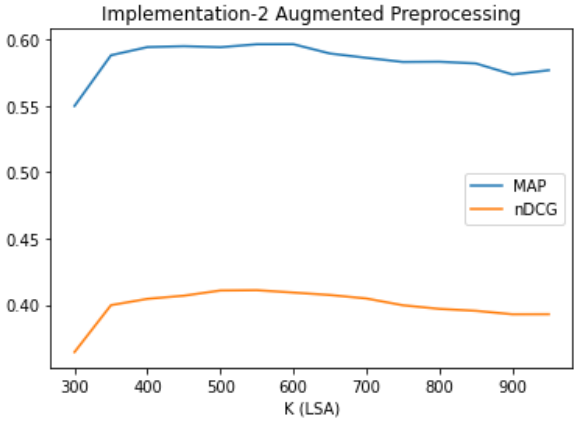
To determine the best possible preprocessing and implementation of LSA and to determine the optimal K value, we run experiments for 6 cases
Case 1: LSA Implementation-1 Augmented Tokenization
Case 2: LSA Implementation-1 Augmented Tokenization With Phrases
Case 3: LSA Implementation-1 Old Preprocessing
Case 4: LSA Implementation-2 Augmented Tokenization
Case 5: LSA Implementation-2 Augmented Tokenization With Phrases
Case 6: LSA Implementation-2 Old Preprocessing
K values are experimented with in the range of 300 to 1000 in steps of 50

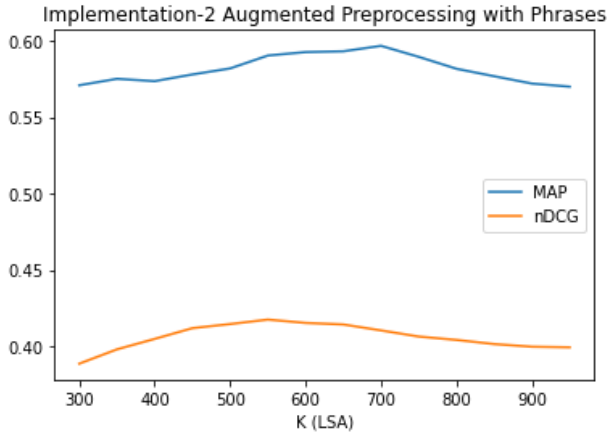
To determine the optimal value of K, we plot the values of MAP@10 and nDCG@10 for all 6 cases.

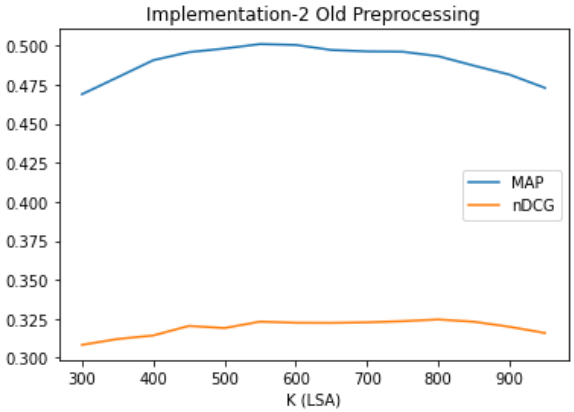


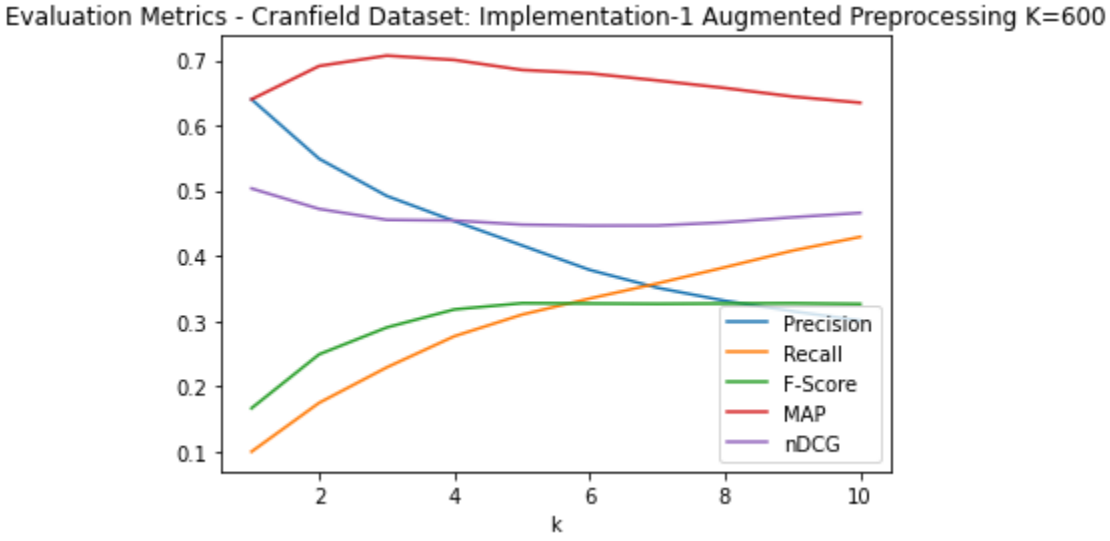



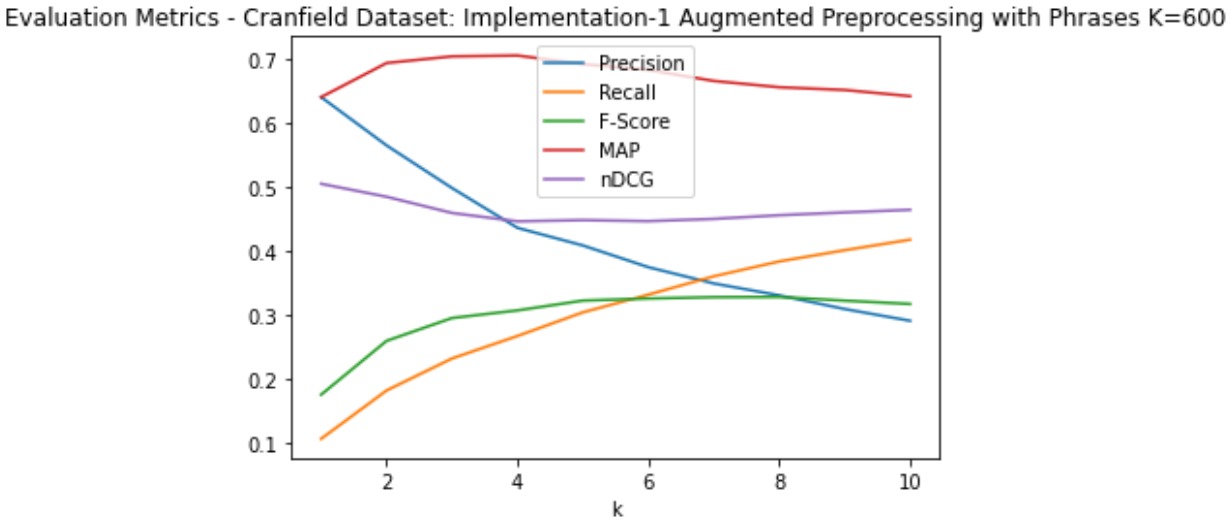



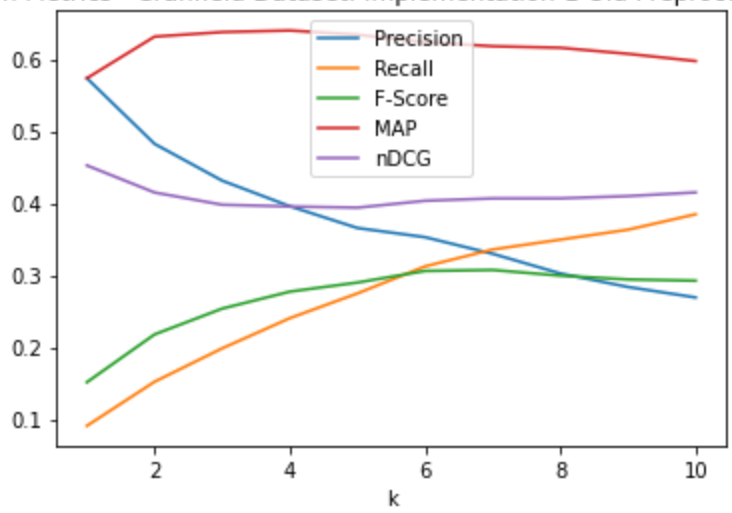

Using both MAP and nDCG curves, we can see that Implementation-1 is better than Implementation-2 since it has higher MAP and nDCG values for the same K. Further, we see that Augmentation with phrases performs better than both Augmentation and Old Preprocessing in terms of nDCG as well as MAP. Further, the variation 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, we see that K=600 is optimal in terms of maximum MAP@4 and at 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 6 cases for K=300 to 1000 in steps of 50. The plots for K=600 can be seen below.

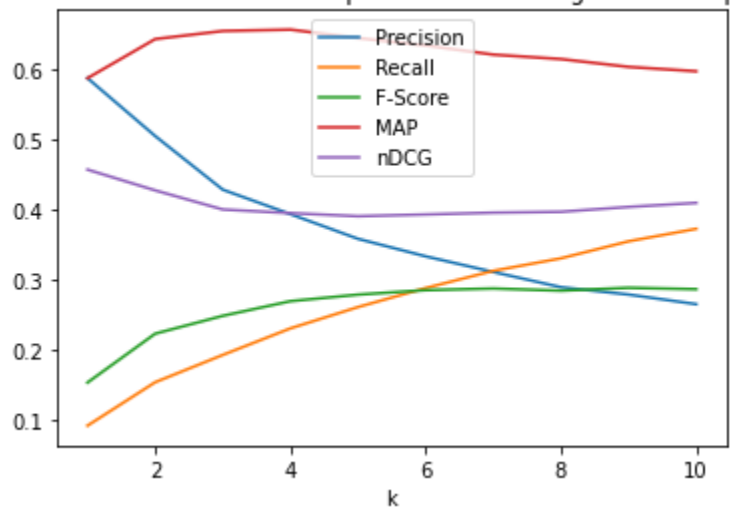


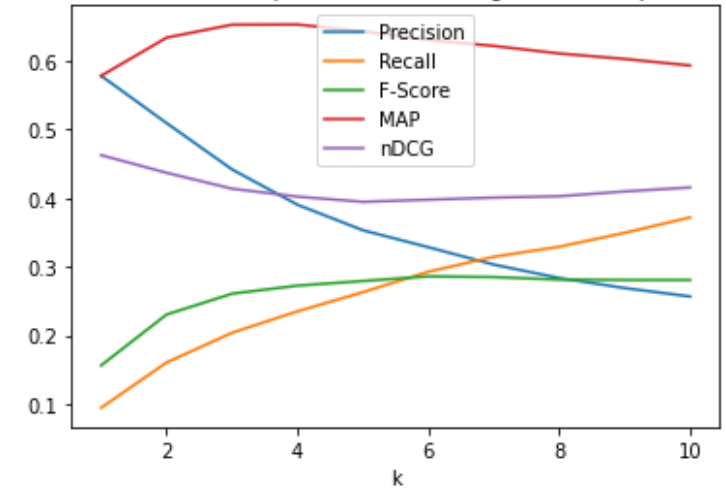
Evaluation Metrics - Cranfield Dataset: Implementation-1 Old Preprocessing K=600

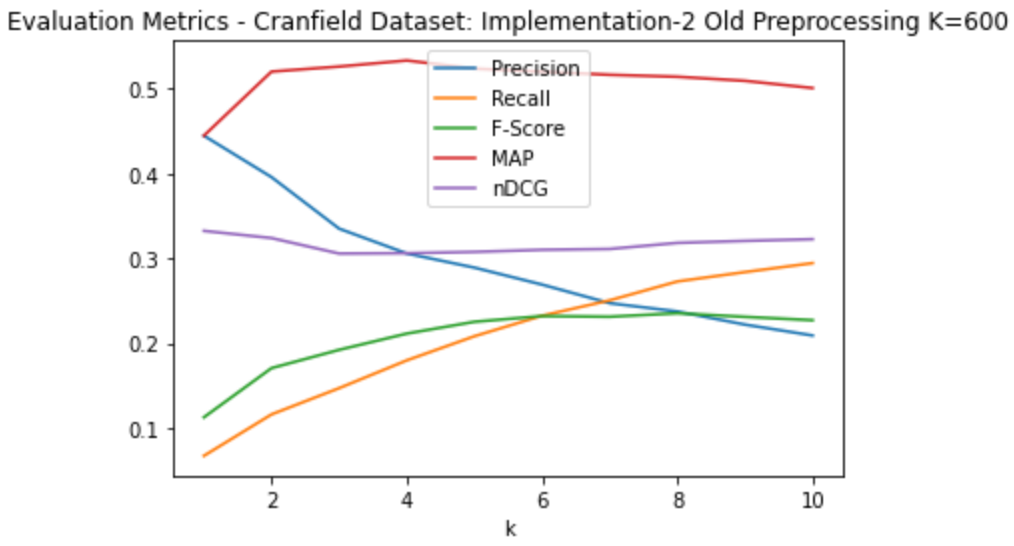



Evaluation Metrics - Cranfield Dataset: Implementation-2 Augmented Preprocessing K=600




Evaluation Metrics - Cranfield Dataset: Implementation-2 Augmented Preprocessing with Phrases K=600






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 below, we see the following:

- nDCG@10 for Augmented Tokenization with Phrases is 0.002 less than that of Augmented Tokenization
- However, 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.

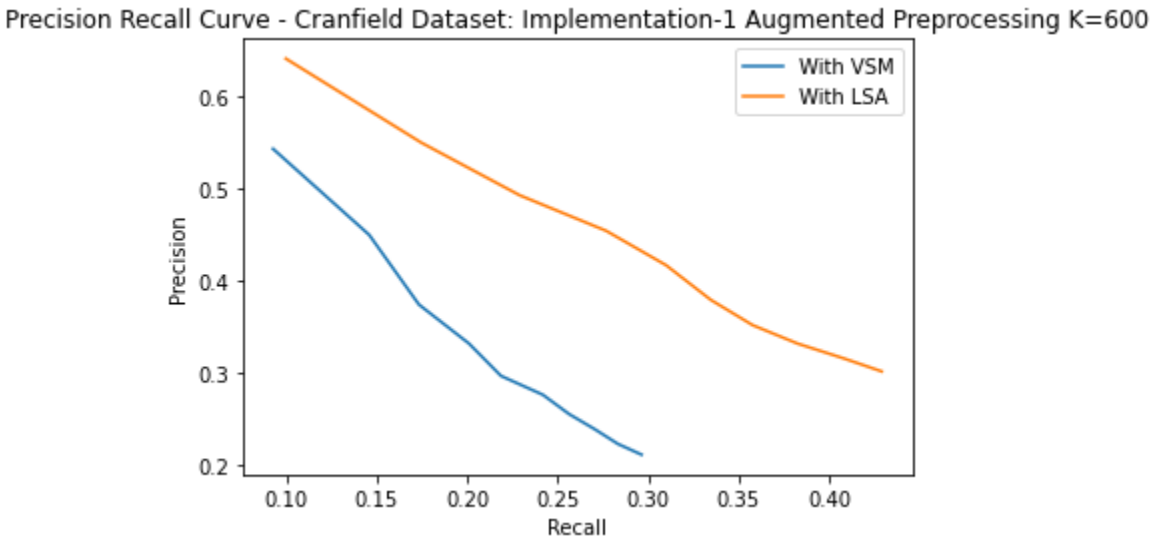
| | Precision@k | Recall@k | Fscore@k | nDCG@k | MAP@k |
|---|---|---|---|---|---|
| **1** | 0.640000 | 0.105724 | 0.174696 | 0.504444 | 0.640000 |
| **2** | 0.564444 | 0.181253 | 0.258860 | 0.484135 | 0.693333 |
| **3** | 0.497778 | 0.231316 | 0.294519 | 0.458853 | 0.703704 |
| **4** | 0.435556 | 0.266460 | 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: Values of all metrics for k = 1 to 10 for K = 600 Augmented Tokenization with Phrases

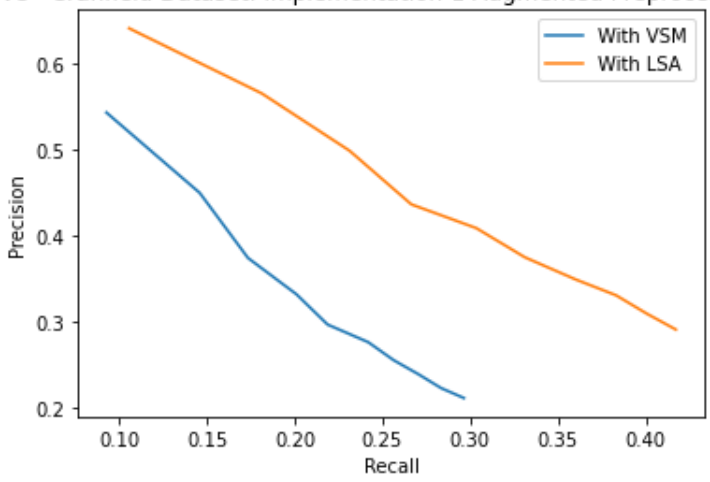
| | Precision@k | Recall@k | Fscore@k | nDCG@k | MAP@k |
|---|---|---|---|---|---|
| **1** | 0.640000 | 0.099860 | 0.166439 | 0.503333 | 0.640000 |
| **2** | 0.548889 | 0.174737 | 0.249068 | 0.471980 | 0.691111 |
| **3** | 0.491852 | 0.229089 | 0.290265 | 0.455249 | 0.707037 |
| **4** | 0.453333 | 0.276858 | 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: Values of all metrics for k = 1 to 10 for K = 600 Augmented Tokenization

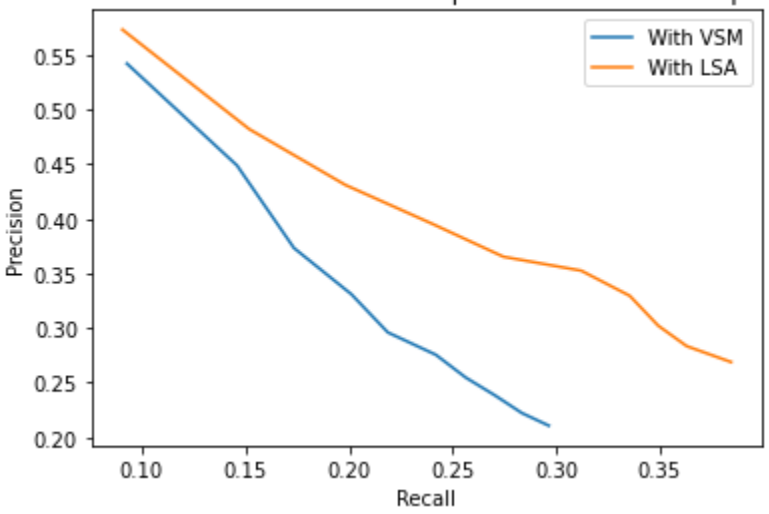
Precision Recall Curves for K = 600 for each of the 6 cases:



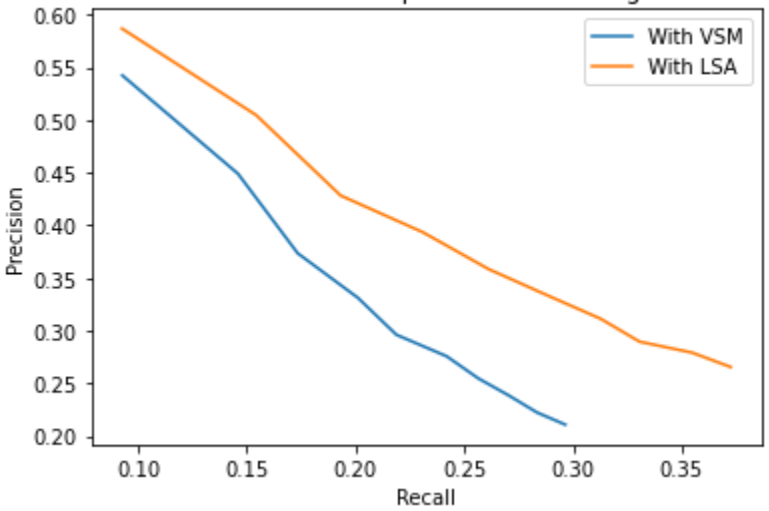
Precision Recall Curve - Cranfield Dataset: Implementation-1 Augmented Preprocessing with Phrases K=600



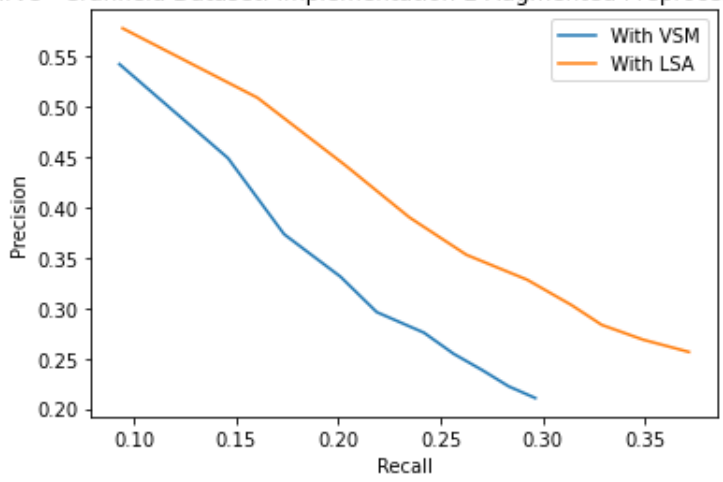

Precision Recall Curve - Cranfield Dataset: Implementation-1 Old Preprocessing K=600



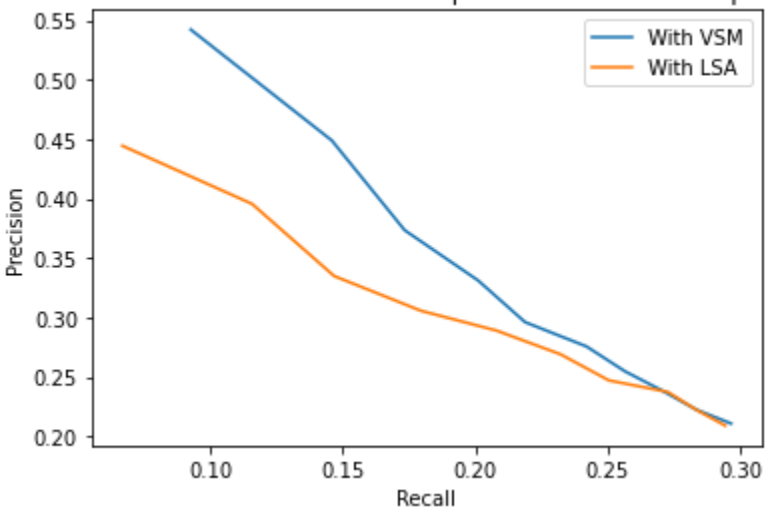

Precision Recall Curve - Cranfield Dataset: Implementation-2 Augmented Preprocessing K=600



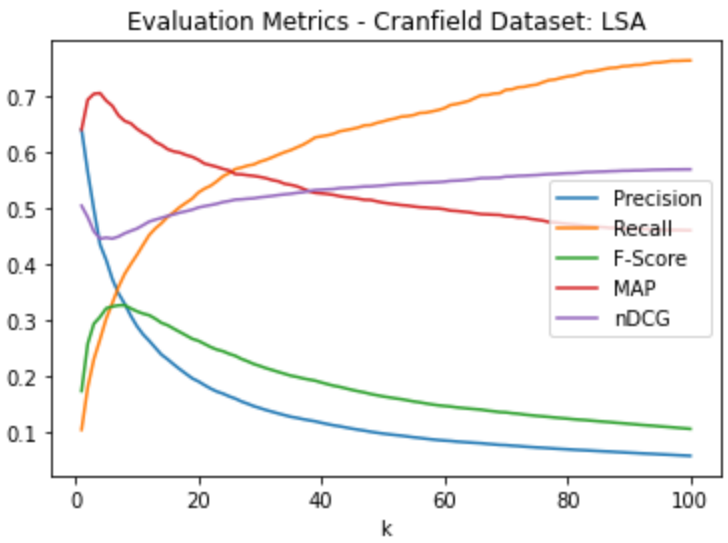
Precision Recall Curve - Cranfield Dataset: Implementation-2 Augmented Preprocessing with Phrases K=600




Precision Recall Curve - Cranfield Dataset: Implementation-2 Old Preprocessing K=600




In all the cases, we see that the area under precision-recall curves for LSA is higher than that of VSM which clearly indicates the superiority of LSA over VSM.




LSA Implementation-1 Augmented Tokenization with Phrases K=600 for the first 100 ranks

- ESA

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.

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. 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. 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. Following this, all documents and queries were transported 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. Now, 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: 500 Articles

Examples from the 500 articles used:

| | |
|---|---|
| Aerodynamics | Standard_conditions_for_temperature_and_pressure |
| Anemometer | Wing |

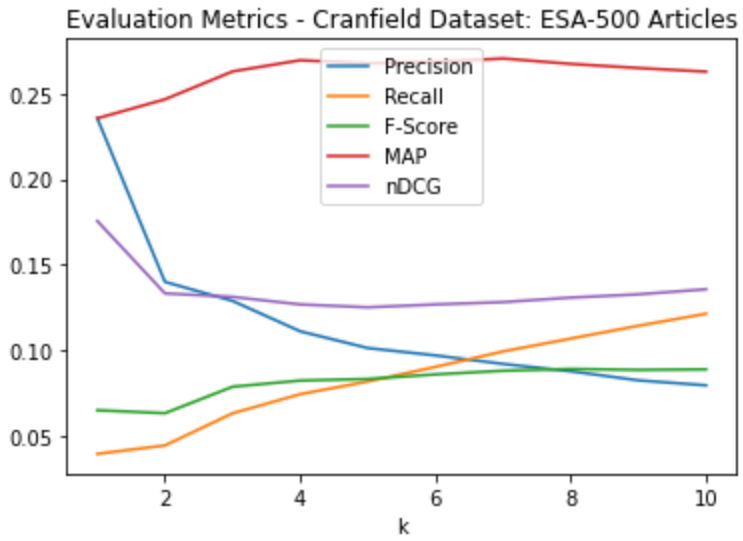
| | |
|---|---|
| Fluid_dynamics | Area_rule |
| Jet_engine | Thrust |
| Lift_(force) | Wind_tunnel |
| Laminar_flow | Windmill |
| Mach_number | Ames_Research_Center |

Portion of the term article matrix used [500-articles]

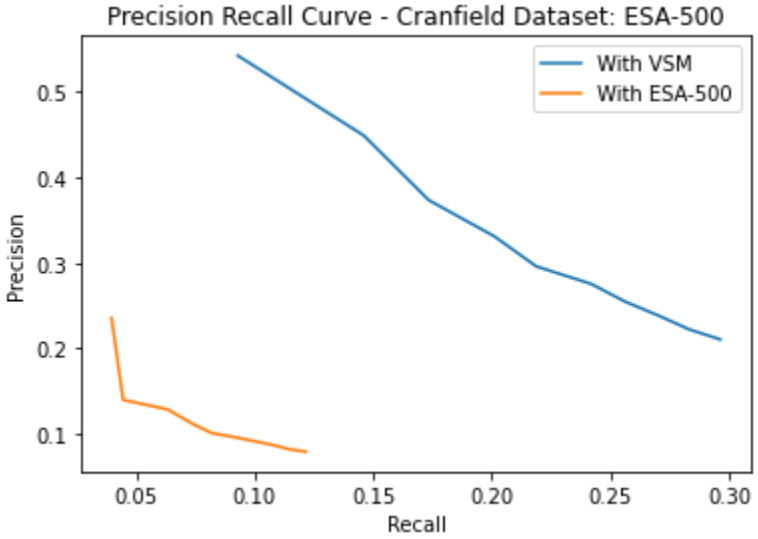
| | Aerodynamics | Anemometer | Fluid dynamics | Jet engine | Lift (force) | Laminar flow | Mach number |
|---|---|---|---|---|---|---|---|
| **aerodynamic** | 33.444298 | 1.320170 | 6.160792 | 2.200283 | 8.361075 | 0.880113 | 2.640339 |
| **greek** | 18.741394 | 15.617828 | 3.123566 | 6.247131 | 0.000000 | 0.000000 | 0.000000 |
| **ἀήρ** | 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 |

| | 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 of all metrics for k=1 to 10 for ESA with 500 Articles




Plots of all metrics for k=1 to 10 for ESA with 500 Articles




Precision Recall Curves for ESA with 500 Articles

Experiment-2: 3343 Articles

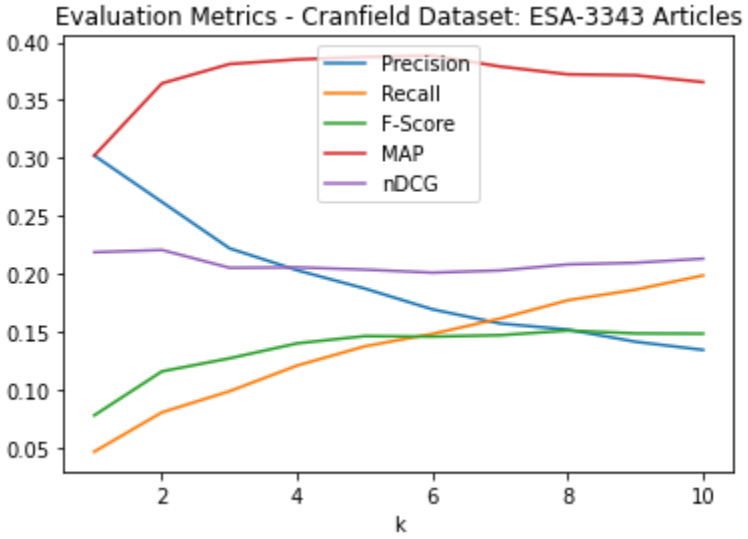
Examples from the articles used:

| | |
|---|---|
| Aerodynamics | Kite |
| Anemometer | Faster-than-light |
| BASE_jumping | Frederick_Abel |
| Extravehicular_activity | Firearm |
| Fluid_dynamics | Overview_of_gun_laws_by_nation |
| Fin | Gunpowder |

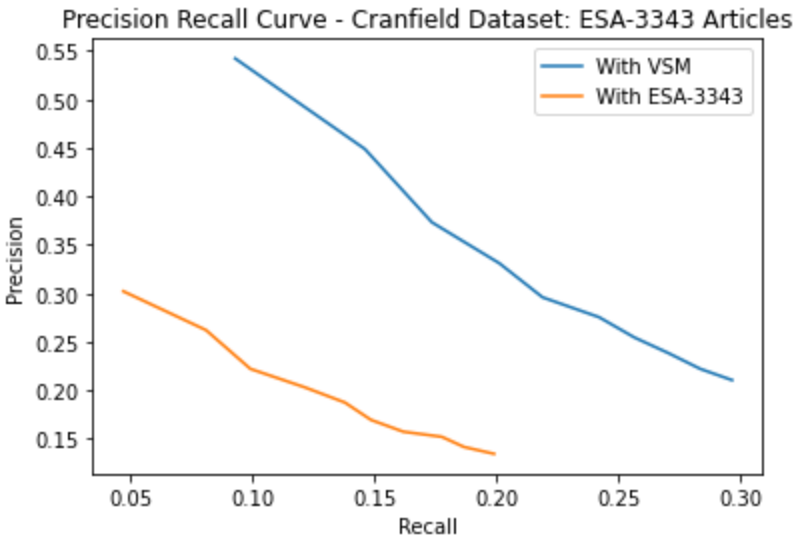
As we can see, the articles in this set, obtained from PetScan with the topic as Aerodynamics and depth = 3 are less relevant to the topic of Aerodynamics.

Portion of term article TF-IDF matrix used:

| | 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 |




Plots of all metrics for k=1 to 10 for ESA with 500 Articles




Precision Recall Curves 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 of all metrics for k=1 to 10 for ESA with 3343 Articles

MAP@10 with 3343 articles = 0.36 MAP@10 with 500 articles = 0.26
nDCG@10 with 3343 articles = 0.21 nDCG@10 with 500 articles = 0.14
From this we can see that even when the background knowledge added is less relevant to the central theme of the articles, the nDCG and MAP improve as compared to the case with less background knowledge.

## Results and Analysis

| | Precision@10 | Recall@10 | F-score@10 | nDCG@10 | MAP@10 |
|---|---|---|---|---|---|

| | | | | | |
|---|---|---|---|---|---|
| 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 |

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)

- LSA vs VSM

Now, 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:

1) Queries that LSA performed well on but VSM performed poorly
2) Queries that VSM performed well on but LSA performed poorly
3) 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 .***
LSA Predicted: [22, 21, 550, 571], VSM Predicted: [102 846  45], Ground Truth: [534, 21, 22, 550],

Doc 21: heat transfer in slip flow
Doc 22: slip-flow heat transfer to a flat plate
Doc 550: laminar heat transfer in tubes under slip-flow conditions
Doc 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. Upon careful observation of the retrieved documents by LSA, it's 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. Also, the fact that we used bigrams (heat_transfer) as terms in the term-document matrix is likely to have aided in the concept formation of heat transfer.

Ground Truth:
Doc 534: consideration of energy separation for laminar slip flow in a circular tube

Here we see that the above relevant document was not retrieved by the LSA in the top 10, the likely reason being the absence of the concept heat transfer or other concepts like slip flow with low concept strengths. (singular values after SVD decomposition)

However, in the case VSM model predicts documents such as Doc 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 ~ 1) with VSM and significantly poorer results (nDCG < 0.1 and AP < 0,1) with LSA were not found.

**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***

LSA Predicted: [433, 1062, 924], VSM Predicted: [ 365 1185  655], Ground Truth: [1290, 1338, 1339, 1340, 1341, 879]

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

The above documents would have been retrieved by LSA mainly due to the presence of concepts representing wing body +  tail + lift etc. It is also important to note that the document containing semantically similar terms of the query (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 pr unswept wings….* etc. Though these documents contain the term wing in them, the term 'flutter' may not be well incorporated in the concept containing the wing body + lift etc.

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

The above documents would have been retrieved by VSM due to the presence of keywords such as 'large' and  'mass' in the documents.

- ## ESA

**Case 1: Queries that LSA 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***

ESA Predicted: [617 615 483] Top-3 wikipedia articles- ['Drag (physics)',  'Ventilation (architecture)',  'External ballistics']
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.

Ground-truth: 613, 614, 615, 616, 617, 618, 548

**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***

ESA Predicted: [ 543 1360 1293]
Top-3 wikipedia articles- ['Speed of sound' , 'Sweep theory', 'Jet engine']

- 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, this 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.

VSM Predicted:[ 945 1063 988]

- 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,

Ground Truth: [985, 990, 945]

- 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 LSA and VSM performed poorly**

**Query-ID - 141 *what analytical solutions are available for stresses in edge-loaded shells of revolution***

ESA Predicted: [424 736 735] Top-3 wikipedia articles- ['Navier–Stokes equations' 'Wind power' 'Generation on the Wind']

- 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

VSM Predicted: [930 890 889]

- 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

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

## Conclusion and Future Directions:

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 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)

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.