A comprehensive study of LSA and P-LSA over VSM for the task of Information retrieval.

Chandra Churh Chatterjee(CS21M013), Anurag Mahendra Shukla(CS21M007),
K Revanth Manik Sai(ME18B050)
Department of CSE, IIT Madras.

Abstract:

The assignments preceding the project involved developing an extensive IR(Information retrieval system) using the classical Vector space model using TF-IDF weight measures. The IR system based on the TF-IDF values as the weight measures for the vector space model did not perform well. Moreover, the vector space model has some limitations in its assumptions. Thus the goal of this project is to tackle some of those limitations using the LSA and the PLSA models. Also the long term aim is to perform a comprehensive and comparative study of the techniques for the task of Information retrieval. Hence in this project we implement LSA (algebraic model), PLSA (probabilistic model) to overcome some of the cons of the Simple Vector space model (VSM) and do a comparison based analysis to understand the advantages and disadvantages of all the three aforementioned.
Keywords: IR - Information Retrieval, LSA - Latent Semantic Analysis, P-LSA - Probabilistic latent semantic analysis and VSM - Vector Space Model.

Introduction:

In the previous assignments, we implemented a IR system using the simple vector space model using the TF-IDF weight measures for the term document matrix. The Cranfield dataset was used for the same. The documents in the dataset were segmented into sentences, following tokenization, stop word removal. The inflection forms of the words were reduced to the root form forms using the snowball stemmer and then the term document matrix was generated using the term frequency (no of times the word occurs in the document) and the inverse document frequency (the number of documents in which the term actually occurs). Same preprocessing was performed to the queries and the query term matrix was created. Cosine similarity was used as the similarity measure as it was calculated between each query and every

document in the corpus, thus allowing us to rank the documents from highest to lowest similarity and then retrieve the top k documents for a particular query.

However, in this project, we improve upon the previously developed IR system by incorporating LSI(Latent Semantic Indexing) into the vector space model. The idea of LSI or LSA is to find out the latent or hidden representations or the concepts or topics that may aid in finding the semantic relatedness between the terms of the vocabulary, which was a major limitation of the vector space model. The LSA also allows us to get the best k reduced rank approximation of the term document matrix, thus allowing us to represent the documents as well as the terms in the reduced k-rank concept space or the dimensions. The LSA takes into account the semantic similarity and hence performs well for synonyms. It also weighs the latent semantic concepts in identifying the similar terms more than just the frequency based similarity.

The Project also involves a deep conception of how a probabilistic model may also improve upon the performance of the simple vector space model for the task of Information retrieval. Thus it involved implementing a Probabilistic latent semantic model for the IR system. The basic idea behind the PLSA is that it models the co-occurrence of the word and the documents as a mixture of conditionally independent multinomial distributions. The important assumption is that each document consists of some topics and each topic can be modeled as a collection of words.

Thus in the following sections we discuss the Methodologies in detail, experimentation, conclusions and inferences and finally some references.

Methodologies:

LSA (Latent Semantic Analysis)

The latent semantic analysis model that has been implemented for the task of the information retrieval is based on some basic assumptions which are as follows:

- The distributional assumption states that words with similar meanings appear together.
- There is some underlying or latent structure in word usage that is partially obscured by variability in word usage.

The key idea is nothing but dual mode factor analysis. The term document matrix already exists from the TF-IDF vector space model, where each cell weighted by the TF*IDF values. Finding the latent structure requires us to perform Singular Value decomposition on the term document matrix, giving us the resultant matrices.

$$A = \mu \Sigma v^T,$$

Where, X is the original term document matrix, μ is the term concept matrix, Σ is the diagonal matrix consisting of the singular values that are given by $\sqrt{eigenvalues(AA^T)}$ and the matrix ν^T refers to the concept document matrix.

Thus we see that as a result of the Singular values decomposition of the original term document matrix we are uncovering the latent dimensions which should aid in understanding the semantic relatedness between the term and the documents. The matrix U is the column orthogonal matrix and the matrix V^T is the row orthogonal matrix.

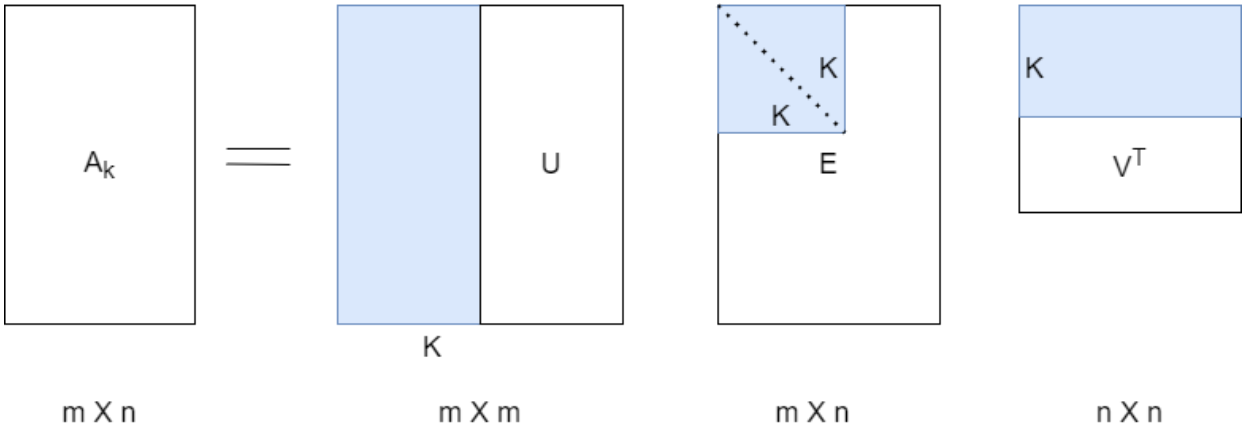


Fig: SVD for LSI

In the above image, the matrix A_k is the reduced rank approximation of the original matrix A.

Now after we get the matrices U, Σ and V^T, we can thus transform the query vectors into the concept space as well. The matrix U is known as the hanger matrix that is column orthogonal, Σ is the stretcher matrix and the matrix V^T is the Aligner matrix, which is row orthogonal. The U matrix contains the eigenvectors of the matrix AA^T and the matrix V^T contains the eigenvectors of the matrix A^TA.

Now the query vectors after being pre-processed in the same way as in the vector space model will be represented in the concept space by the following operation.

$$\widehat{q} = q\,\mu_k\,\Sigma_k^{-1}$$

Now after transforming the query vectors, in the same concept space, we can thus calculate the cosine similarity for each query vector and all the document vectors and thus get the top k similarity based ranks for every query vector.

Important property: One of the most important properties of the SVD is that it gives the best k rank approximation of the original matrix in the least squares sense measured by the frobenius norm of the discrepancy matrix $(M - M_k)$ given by:

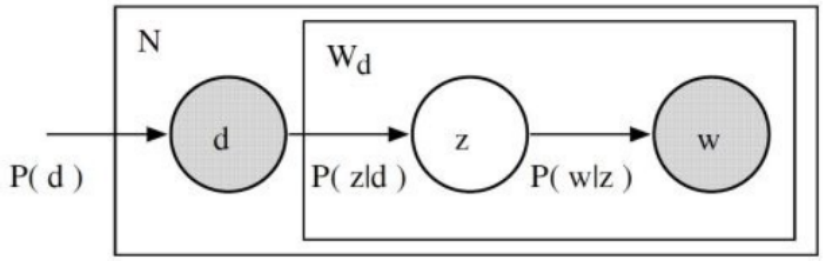
$$||F||_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m} X_{ij}^2}$$

Thus the theoretical comprehensive study of the LSA for IR is concluded here and in the future sections we will look into the experimentation and the inference and the conclusions derived from implementing LSA.

P-LSA (probabilistic Latent Semantic Analysis)

As the name suggests, the P-LSA is actually a probabilistic model. The basic idea is to learn a probabilistic model with latent concepts that can generate the data that we observe in our document term matrix. PLSA adds a probabilistic spin to the assumptions of the LSA.
The different probabilities used for modeling are as follows:

- P(z|d) - probability that a concept z is present in the document d.
- P(w|z) - Probability with which a word w is drawn from the concept z



The joint probability of a word w and a document d thus can be modeled as follows:

$$P(d|w) = P(d) \sum_{z} P(z|d)P(w|z)$$

P(d), P(z|d) and P(w|z) turn out to be the parameters of the model and thus can be learned using the EM (Expectation maximization) algorithm and the probability P(d)
Can be directly modeled from the corpus itself.
There exists an interesting connection of LSA with PLSA which can be understood as follows.

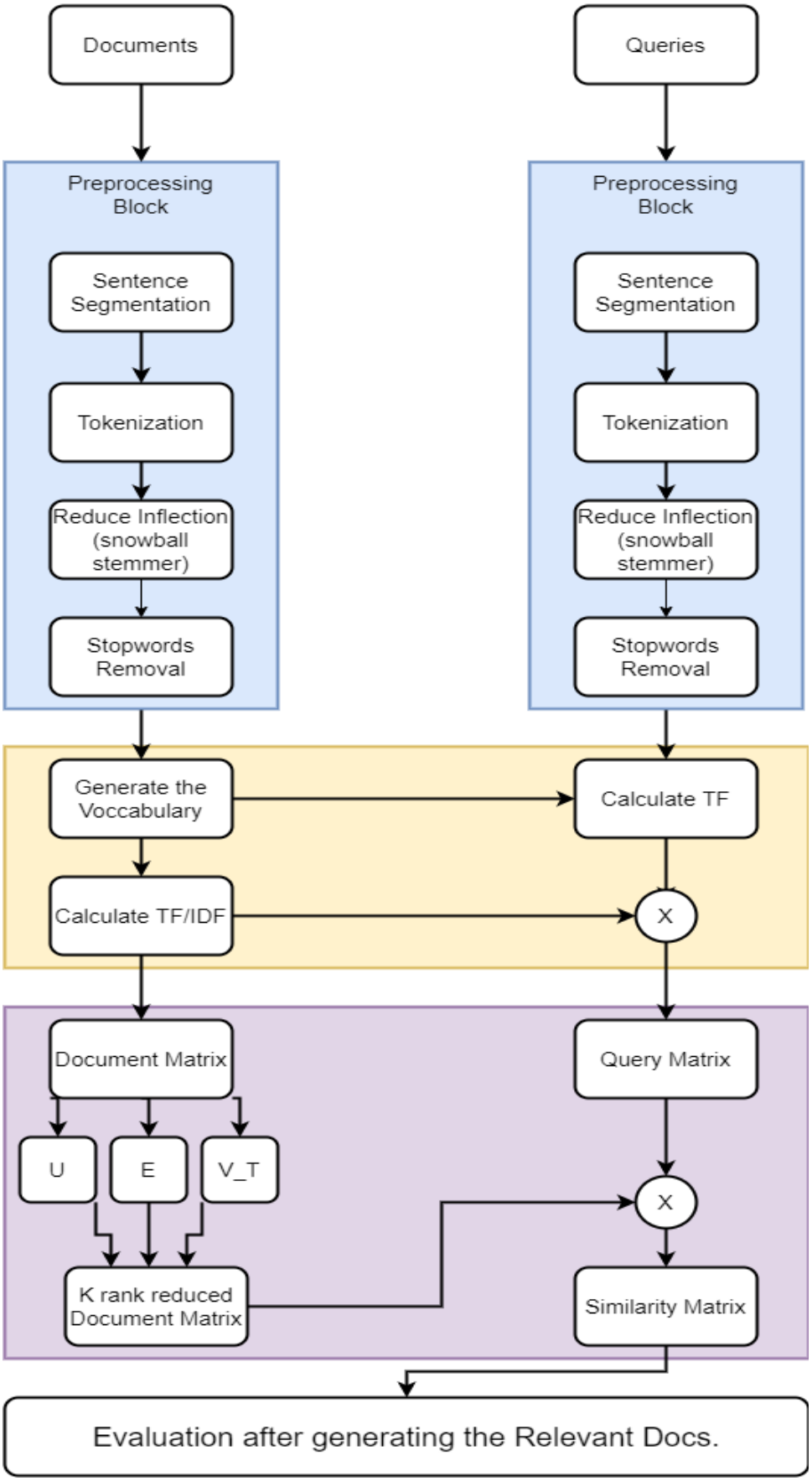
$$P(D,W) = \sum_{Z} P(Z)P(D|Z)P(W|Z)$$

$$A \approx U_t S_t V_t^T$$

In our experimentation, we implemented PLSA using normalized non negative matrix factorisation and not using Expectation maximization which we will go through in the Experimentation section of this report.
Thus the theoretical explanation of the PLSA concludes here and in the further sections we will see the implementation of the PLSA.

The complete Proposed methodology is displayed as follows in the next page.

Flow of the complete Proposed Methodology



Experimentation:

LSA (Latent Semantic Analysis) -

The LSA has been implemented in the same code that was developed while developing the vector space model for the IR system. The term document matrix that we obtained there is the same one that has been used for performing SVD to get the latent/ hidden concepts.

We perform the SVD using truncatedSVD function provided by the Sci-kit learn library, where we experiment with various values for the number of components or the number of latent concepts as we say. We transform the query vectors into the k rank reduced concept space using the dual mode analysis. After the transformation, the cosine similarity is calculated and the as in the vector space model, we rank the documents based on similarity for each query vector and finally the model is evaluated using the Cranfield dataset using evaluation measures such as Precision, Recall, Mean Average Precision, nDCG and F_score.

PLSA (Probabilistic Latent Semantic Analysis) -

The probabilistic latent semantic analysis is implemented using non-negative matrix factorization. Relation between PLSA and NMF and Implications, suggests an important property that any local max likelihood solution of PLSA is the solution of NMF(Non-negative matrix factorisation) with KL-divergence.
As a result of matrix factorisation thus we observe that the original term document matrix after normalization can be factored into the following matrices.

$$A \approx \hat{A} = W \cdot H$$

Where, W is (N X K) and H is (K X M), thus fulfilling the dimensional reduction task. In terms of the the conditional probabilities, the matrices W, H represent the conditional probabilities P(z|d) and P(w|z). Thus these matrices are then used to transform the query vectors. Finally the similarity of the query vectors is calculated using cosine similarity between each query and every document vector followed by ranking the documents for each query and then evaluating the model using the same measures such as the Precision, Recall, Mean Average Precision, nDCG and F_score.

Finally a comparative study of the performances is performed to establish some light on the advantages and disadvantages of the 2 proposed methods over the simple vector

space model. The number of components were varied and hyper parameter tuning was performed using a manual iterative approach. nDCG (normalized discounted cumulative gain) was chosen to be the most important evaluation measure for the comparative analysis.

The entire code was developed using Python 3.9 and some important libraries such as Sci-kit learn, nltk, numpy, pandas and matplotlib. The models were developed using CPU and no GPU was required for the same.

Observations and Inferences:

Metric/Measure based observations

We developed the LSA and the PLSA model as mentioned in the experimentation and the Methodology. The LSA and PLSA were tuned for the number of components or the latent dimensions. The LSA was performed for different values of the number of latent dimensions starting from 100 to 1100 with 100 as steps in between. The results of such an experiment are as follows.

	Number of components for LSA										
Metrics		100	200	300	400	500	600	700	800	900	1000	1100
	F-Score	0.28	0.31	0.31	0.321	0.324	0.325	0.324	0.325	0.326	0.325	0.325
	NDCG	0.37	0.44	0.45	0.46	0.473	0.475	0.476	0.476	0.477	0.476	0.477

From the above table we can see that using 900 components for the LSA we achieve the best NDCG and F-Score value. Thus after tuning the value of the number of components, we stick with the number of components to be 900.

Now we perform a comparative analysis of the methods and see the performance of all three methods (VSM, LSA and PLSA). The methods are evaluated using the precision, recall, MAP(mean average precision), F_score (harmonic mean of the presion and the recall) and the NDCG (normalized discounted cumulative gain). In the below diagrams we see the corresponding metrics for all the three methods (Vector space model), (Latent semantic analysis) and the (probabilistic Latent Semantic Analysis).

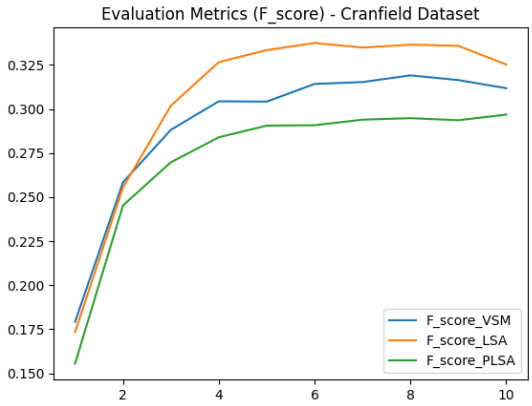


Fig - F_score metrics

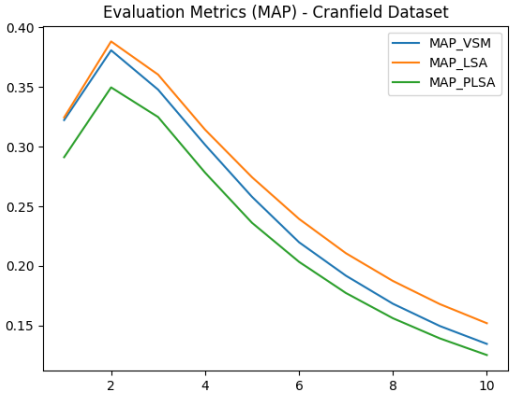


Fig - Mean average Precision

From the above figures, it is clear that the LSA gives better F-score results than a simple vector space model and, surprisingly, PLSA performs the poorest. From the graph it is clear that for every rank **K,** LSA is performing at least as good as the vector space model. The LSA results that we have generated are for 900 latent representations or the latent concepts that are the same for the PLSA as well.

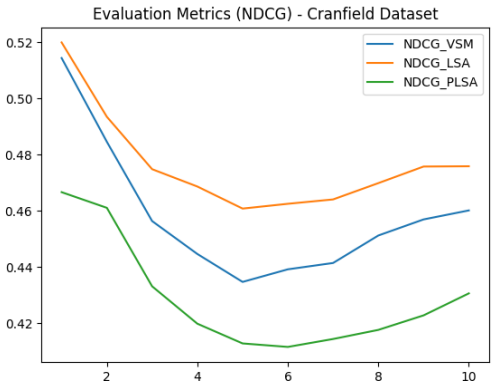


Fig - NDCG metric

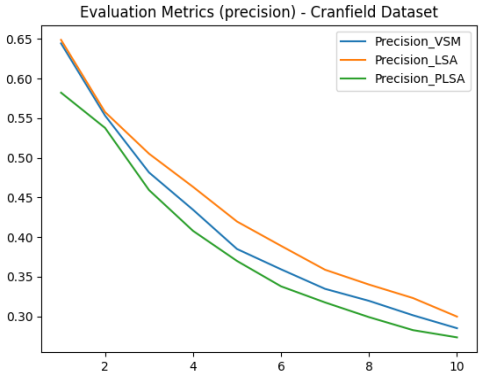
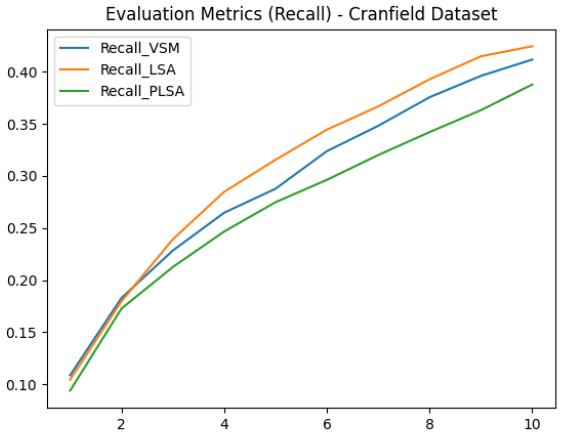


Fig - Precision metric

The NDCG value also is the best from LSA compared to that of the vector space model. Thus it can be observed that the LSA performs better than the simple vector space model even though the performance improvement is only marginal.

The below figure is that of the Recall metric, as it should be, the recall is increasing with the value of **K**, as it is not a good metric for Information retrieval systems.



The results of the above observations are listed in the table below. The below table actually gives the values of F-score and the NDCG (considered as the best metrics for IR evaluation). From the table we can actually confirm the values and see that LSA with the latent dimensions of 900 performs better than the vector space model.

	Metrics					
Methodologies		Precision	Recall	MAP	F-Score	NDCG
	VSM	0.28	0.41	0.314	0.311	0.460
	LSA	0.30	0.42	0.147	0.326	0.477
	PLSA	0.272	0.393	0.119	0.298	0.434

The LSA using the pennTreeBank tokenizer performed better than using the naive method and also we observed that lemmatization gave poorer results as compared to the snowball stemmer.

Hence we stuck with the snowball stemmer and the pennTreeBank tokenizer in our proposed methodology.

Retrieved documents based observations

Some of the queries and the retrieved docs form them for both the VSM and LSA model are represented below.

1. "query number": 1 - "query": "what similarity laws must be obeyed when constructing aeroelastic models of heated high speed aircraft ."
   For the above query
   - Ground Truth (most relevant document) - Doc Id - 486
   - In the VSM retrieved document list Doc Id occurs at position - 10
   - In the LSA retrieved document list Doc Id occurs at position - 4
2. "query number": 4 - "query": "can a criterion be developed to show empirically the validity of flow solutions for chemically reacting gas mixtures based on the simplifying assumption of instantaneous local chemical equilibrium ."
   For the above query
   - Ground truth (most relevant document) - Doc Id - 488
   - In the VSM retrieved document list Doc Id occurs at position - 2
   - In the LSA retrieved document list Doc Id occurs at position - 1
3. "query number": 145 - "query": "what are the best experimental data and classical small deflection theory analyses available for pressurized cylinders in bending ."
   For the above query
   - Ground truth (most relevant document) - Doc Id - 955
   - In the VSM retrieved document list Doc Id occurs at position - Does not exist in the top 15 retrieved docs.
   - In the LSA retrieved document list Doc Id occurs at position - 11

Thus from the above observations we see that the LSA is better at retrieving the relevant documents.

Observation backing the fact that LSA relies more on the semantic relatedness than counts/frequencies as in VSM and hence it gives better performance than the VSM.
For Example:

For the query - ("query number": 50 - "query": "what are the effects of small amounts of gas rarefaction on the characteristics of the boundary layers on slender bodies of revolution .").
According to the VSM model the second most retrieved document turns out to be DocId - 192 - which seems to be completely based on the counts/ frequencies of the words, whereas in the LSA DocId-192 is the 4th most relevant document. But at the same time the ground truth 2nd relevant document is the DocId - 326, which in the VSM

is the 9th relevant document and in the LSA it is the 3rd most relevant document. Thus we see that in LSA the DocId -326 is more relevant than the DocId-192, which is originally true because VSM only relies on the counts/frequencies whereas LSA depends more on the semantic relatedness. **Hence one of our hypotheses mentioned in the proposal stands justified based on the inductive analysis on the aforementioned observations.**

Synonymy inferencing using LSA.

DocId - 168	Words		
Methods		Distort (occurring in the document)	Deform (not occurring in the document)
	VSM	1.85	0
	LSA	56.4	13.9

From the above table it is evident that LSA using semantic relatedness improves the weight of the synonyms of the words occurring in the document. Whereas VSM is only frequency based, hence LSA can handle synonymy but VSM cannot handle synonymy. The average weight of the words for the DocId - 168 mentioned above is 5.5, thus we see that LSA is increasing the weight of the synonyms of distort above the average weight. **Thus our second hypothesis, that LSA handles synonyms better than VSM stands also justified.**

Conclusions:

Thus we see that **LSA with the assumption that "words with similar meanings appear together"** performs better than **VSM with the assumption "words occurring in a document are statistically independent", On** the task of **Information Retrieval, On** the evaluation measures {**MAP, F-Score, NDCG**}, **On** the **Cranfield** dataset.

Similarly PLSA performs poorly compared to both LSA and VSM, **On** the task of **IR, On** the evaluation measures {**MAP, F-Score, NDCG**}, **On** the **Cranfield** dataset. The reason for that could probably be due to overfitting on the dataset, and also general convention do LDA instead of only doing PLSA.

References:

1. Course materials provided by our course Instructor.
2. Relation between PLSA and NMF - https://dl.acm.org/doi/abs/10.1145/1076034.1076148
3. Vector Space Model - https://en.wikipedia.org/wiki/Vector_space_model
4. LSA - https://en.wikipedia.org/wiki/Latent_semantic_analysis
5. Evaluation measures for IR - https://en.wikipedia.org/wiki/Evaluation_measures_(information_retrieval)