CS6370(NLP) Project: LSA Performance

Manish Nayak

Indian Institute of Technology, Madras
{ce22b069}@smail.iitm.ac.in

Abstract. **Keywords:** VSM · LSA · CRN · Query auto-completion

1 Introduction

Information Retrieval is a critical task with diverse applications across various search engines. It encompasses a range of problems where the objective is to retrieve pertinent documents from a database based on a given query. This field focuses on identifying the most relevant set of documents, often denoted as "k," from a dataset. An iconic example of Information Retrieval is exemplified by Google Search, where numerous web pages are indexed by web crawlers. When presented with a query, the search engine endeavors to fetch the most relevant collection of web pages or documents and arranges them based on relevance. Despite the prominence of web search engines, Information Retrieval extends to a broad spectrum of applications beyond web searches.

2 Problem Statement

We are driven by the goal of performing effective and efficient information retrieval on the Cranfield Dataset, which is a small corpus consisting of 1400 scientific abstracts and 225 queries. Associated with each query is a list of relevant documents with their corresponding relevance score. The relevance score denotes whether a document answers the query, whether it is just an example or if it only refers to a few things relevant to the document.

Our task is to build an IR system with auto-complete mechanism, which takes queries as input,(if the requested query is in the query corpus, it finds a few relevant queries already present in the query corpus else it directly compares itself to the documents) and reports the topmost k relevant documents as output if available.

3 Background

The baseline model (which uses the vector space formalism) assumes orthogonality between all words. This is not a valid assumption. On top of that, the vectors are very large and sparse because of the size of the vocabulary, as a result of which, it takes a considerable amount of time to compute the cosine similarity.

The other shortcomings of this model include:

- Failure to capture polysemy and synonymy.
- Addition of a new term requires us to recalculate all the vectors.
- The sequential ordering in which the terms appear in a document is lost in the vector space representation.
- It has limited conceptual understanding. It just represents documents as a linear combination of words and is purely a lexical matching approach.

4 Approach

We have to build upon the baseline model, which includes a few pre-processing steps namely, Sentence Segmentation, Word Tokenisation, Stopword removal, lemmatization; and evaluate the model performance based on the most frequently used metrics in the field.

We have tried to implement the LSA approach, CRN-based PMI approach; along with the feature of auto-completion of queries, based on a very similar tf-idf approach.

Ranking: Similar preprocessing steps are to be applied on the queries and the query vectors are obtained. Similarity between the documents and the query are established by cosine similarity. Then we rank based on output similarity scores. A large value of cosine means vectors are in a similar direction, small values of cosine mean vectors are not in a similar direction.

Evaluation: The ranked documents are evaluated based on different metrics like MeanPrecision@k, MeanRecall@k, MAP@k, nDCG@k, Mean F-Score@k.

4.1 Latent Semantic Analysis

Similar terms occur in similar documents, similar documents have similar terms, to break this circularity, we use the latent variables called concepts. So, we map the documents as well as terms to concept space (analogical to projection on eigen vectors) to find similarity between the documents using higher order associations(synonymy).This also helps us in reducing the dimension of vector representation for all documents by using LSA for better representation in terms of space(storage of huge vectors).LSA solves the problem of orthogonality of dimensions to an extent i.e. mitigates the problem of identifying synonymy (merge the dimensions associated with terms that have similar meanings), by capturing the higher order associations between the words. This can be understood in the mathematical sense, by representing the vectors in a more compact space (based on SVD - Singular Value Decomposition, which helps to find the latent semantic structure of words spread across the document).

SVD decomposes the word-document matrix as : $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V^T}$, where $\mathbf{U}$ is matrix whose columns are the eigenvectors of $\mathbf{MM^T}$, $\mathbf{\Sigma}$ is a diagonal matrix whose entries consist of the singular values(square roots of eigenvalues) of $\mathbf{MM^T}$, arranged in descending order and $\mathbf{V^T}$ is a matrix whose columns are eigenvectors of $\mathbf{M^TM}$, this can then be used to get the k-rank approximation $\mathbf{M_k} = \mathbf{U_k}\mathbf{\Sigma_k}\mathbf{V_k^T}$, by selecting the k-largest diagonal values. It can also be

thought of as transformation of vector space, where V_T keeps the length of vectors the same, Σ has the effect of stretching or compressing all coordinate points along the values of its singular values, and finally U rotates our feature space.

Intuitively, to rotate as well as stretch the documents into the concept space, we do the following transformation:

$$d' = \Sigma_k \cdot V_k^T \tag{1}$$

Similarly, the query document has to be transformed into the concept vector space by using the exact transformation as done to the document vector.

$$\hat{q} = qU_k\Sigma_k^{-1} \tag{2}$$

$$q' = \hat{q} \cdot \Sigma_k \tag{3}$$

4.2 Case Retrieval Net (CRN) based PMI (Pointwise Mutual Information)

The most fundamental item in the context of CRNs are so-called Information Entities (IEs). These may represent any basic knowledge item, such as a particular attribute-value-pair. A case then consists of a set of such IEs, and the case base is a net with nodes for the IEs observed in the domain and additional nodes denoting the particular cases. IE nodes may be connected by similarity arcs, and a case node is reachable from its constituting IE nodes via relevance arcs.

The concept of Information Entities (IEs) and their relationships within a Case Retrieval Network (CRN) can be related to Pointwise Mutual Information (PMI) in the context of capturing semantic relationships and associations between these entities.

The Pointwise Mutual Information approach involves calculating the PMI score for pair of words, based on their co-occurrence frequencies in the corpus.

$$PMI(w_1, w_2) = \log_2(\frac{P(w_1, w_2)}{P(w_1) \times P(w_2)})$$

where $P(w_1, w_2)$ is the probability of co-occurrence of w_1 and w_2, $P(w_1)$ is the probabiltiy of w_1 occurring, and $P(w_2)$ the probabiltiy of w_2 occurring individually.

The PMI score reflects how much the co-occurrence of two words is greater than what would be expected if they were independent. A positive PMI score indicates that the words co-occur more often than expected by chance, while a negative PMI score indicates that they co-occur less often than expected.

In NLP tasks such as information retrieval, text classification, or sentiment analysis, PMI can be used to identify significant word associations, extract meaningful features, or improve the performance of machine learning models by capturing semantic relationships between words.

5 Experimentation

The purpose of hypothesis testing in this study is to rigorously evaluate the statistical significance of observed differences in the performance metrics between the Vector Space Model (VSM) and Latent Semantic Analysis (LSA) techniques for information retrieval. The process would be carried out with :

5.1 Latent Semantic Analysis

5.1.1 Dataset Used

We used Cranfield Dataset containing 225 queries and 1400 documents to implement LSA. LSA assumes there exists a hidden semantic structure in the data that is partially obscured by the randomness of word choice with respect to retrieval. Singular value decomposition is a statistical technique used to identify the latent structure and remove the arbitrary noise in the data by retaining the best k-concepts and also amplifying the magnitude of those k-concepts as per there importance using the singular values.

Procedure:

1. Construct a term-document matrix using TF-IDF.
2. Apply SVD on the term-document matrix to get the k-rank approximation of the term-document matrix, where k is a hyperparameter which we will tune using Cumulative Explained Variance Ratio(Rule-of-Thumb) set to 80%. k is nothing but the number of concepts, and eigen vectors in SVD transformations imply the directional vectors in the concept space.
3. Transform the query and documents into the concept space. Cosine similarity is used to find the similarity between the query and the documents. Ranking is done based on the similarity score.

5.1.2 LSA Hyperparameter Tuning

For choosing the best k value, Cumulative Explained Variance Ratio was used to obtain the top k singular values, which capture nearly 80% of the energy (total variance) . We used 80% to prevent any chances of overfitting due to high cumulative explained variance value.For Cranfield Dataset, k was found to be 440.

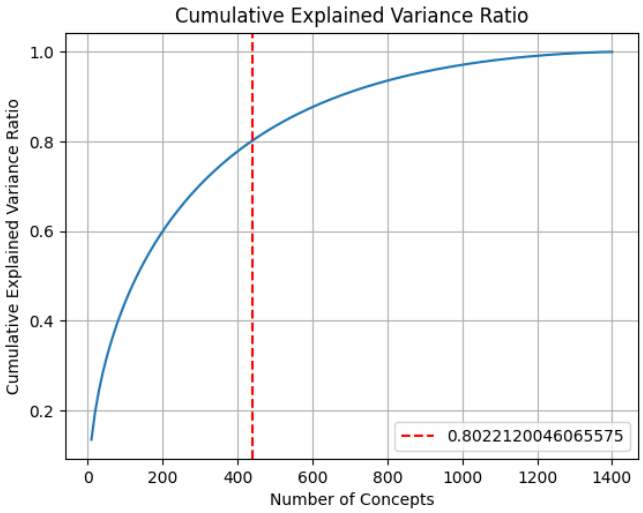


Fig. 1: Cumulative Explained Variance Ratio vs k

5.1.3 VSM Vs LSA Results

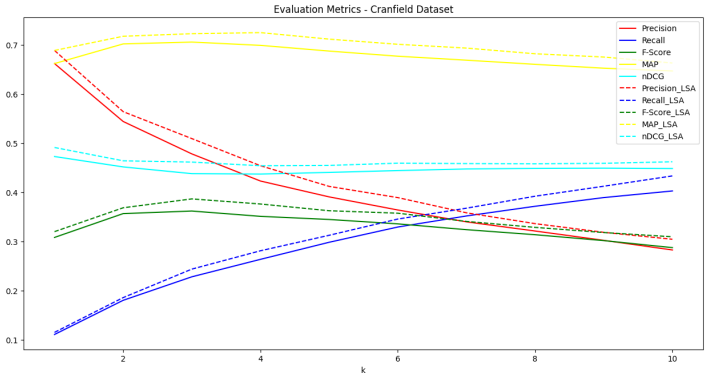


Fig. 2: LSA vs VSM

	LSA with 440 latent dimensions					Basemodel				
	Precision	Recall	f-score	MAP	n-DGC	Precision	Recall	f-score	MAP	n-DGC
1	0.6889	0.1156	0.3204	0.6889	0.4916	0.6622	0.1113	0.3086	0.6622	0.4732
2	0.5644	0.1863	0.3689	0.7178	0.4645	0.5444	0.1808	0.3571	0.7022	0.4520
3	0.5096	0.2442	0.3870	0.7230	0.4619	0.4785	0.2285	0.3623	0.7059	0.4384
4	0.4544	0.2815	0.3766	0.7252	0.4544	0.4233	0.2641	0.3515	0.6993	0.4375
5	0.4124	0.3130	0.3629	0.7119	0.4551	0.3911	0.2988	0.3452	0.6876	0.4409
6	0.3896	0.3460	0.3579	0.7014	0.4597	0.3644	0.3298	0.3362	0.6772	0.4446
7	0.3587	0.3681	0.3412	0.6937	0.4588	0.3403	0.3523	0.3244	0.6691	0.4479
8	0.3367	0.3925	0.3291	0.6822	0.4584	0.3217	0.3720	0.3142	0.6607	0.4491
9	0.3190	0.4128	0.3186	0.6755	0.4594	0.3027	0.3897	0.3022	0.6528	0.4495
10	0.3049	0.4338	0.3100	0.6634	0.4625	0.2831	0.4031	0.2879	0.6471	0.4487

5.2 CRN

5.2.1 Dataset Used

We used Cranfield Dataset containing 225 queries and 1400 documents to implement CRN. CRN is a method to compute similarity propagation by initiating activation through the similarity arches and retrieve documents based on the transformed query and doc vector using term-term similarity matrix. PMI is used to compute the distributional similarity between the terms assuming that words are similar if they co-occur in more than one documents.

Procedure:

1. Construct a term-document matrix using TF-IDF.
2. Construct a term-term matrix using PMI.
3. Compute matrix multiplication on query and document (since we are changing the vector space of query, same we need to do for documents for proper retrieval).
4. Cosine similarity is used to find the similarity between the query and the documents. Ranking is done based on the similarity score.

5.2.2 CRN vs VSM Results

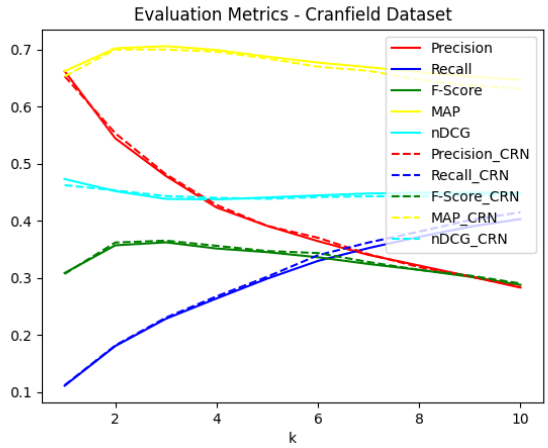
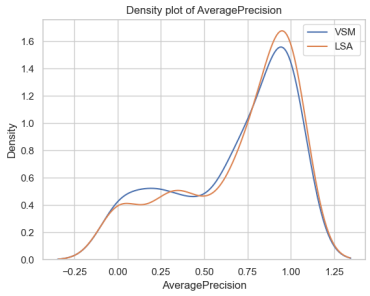


Fig. 3: CRN vs VSM @6

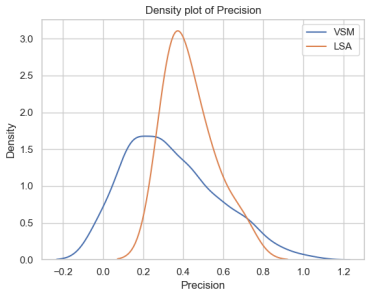
	CRN					Basemodel				
	Precision	Recall	f-score	MAP	n-DGC	Precision	Recall	f-score	MAP	n-DGC
1	0.6533	0.1120	0.3071	0.6533	0.4625	0.6622	0.1113	0.3086	0.6622	0.4732
2	0.5533	0.1818	0.3620	0.7000	0.4533	0.5444	0.1808	0.3571	0.7022	0.4520
3	0.4815	0.2305	0.3653	0.7000	0.4434	0.4785	0.2285	0.3623	0.7059	0.4384
4	0.4278	0.2678	0.3563	0.6963	0.4403	0.4233	0.2641	0.3515	0.6993	0.4375
5	0.3911	0.3021	0.3471	0.6843	0.4385	0.3911	0.2988	0.3452	0.6876	0.4409
6	0.3704	0.3397	0.3436	0.6699	0.4417	0.3644	0.3298	0.3362	0.6772	0.4446
7	0.3416	0.3620	0.3279	0.6627	0.4429	0.3403	0.3523	0.3244	0.6691	0.4479
8	0.3189	0.3811	0.3141	0.6471	0.4428	0.3217	0.3720	0.3142	0.6607	0.4491
9	0.3027	0.4013	0.3042	0.6376	0.4448	0.3027	0.3897	0.3022	0.6528	0.4495
10	0.2840	0.4150	0.2903	0.6312	0.4456	0.2831	0.4031	0.2879	0.6471	0.4487

6 Observations

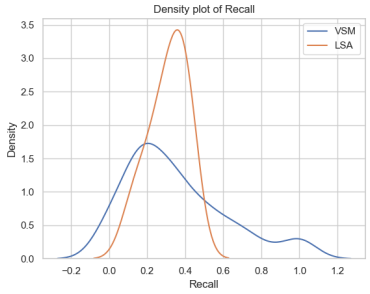
6.1 LSA



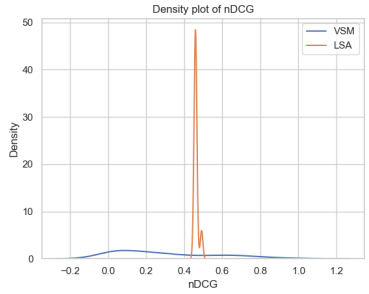
(a) LSA vs. Avg Precision @6



(b) VSM vs. LSA Precision @6



(c) LSA vs. VSM Recall @6



(d) VSM vs. LSA NDCG @6

Fig. 4: Comparison of LSA and VSM

From the distributional graphs, we observe that LSA shifts the distribution slightly towards right and has become somewhat smoother. This implies that Document Representation is improved using Latent Semantic Analysis.Here are a few Examples where the LSA performed better than baseline model

- query id : 109
- query 174 : 'panels subjected to aerodynamic heating .'
- relevant documents : [860, 861, 606, 980, 12, 766].
- documents retrieved by VSM: [1008 , 859, 658 ,864 , 856, 857, 627 , 391 ,766 , 858]
- documents retrieved by LSA :[1008, 859, 658, 856, 857, 391, 627, 766, 858, 948]

Doc 766 : "experimental investigation at mach number of 3. 0 of effects of ***thermal*** stress and buckling on flutter characteristics of flat single-bay ***panels***

of length-width ratio 0. 96 . flat, single-bay, skin stiffener ***panels*** with length-width ratios of 0.96 were tested at a mach number of 3.0, at dynamic pressures ranging from 1,500 to stagnation ***temperatures*** from 300 f to effects of ***thermal*** stress and buckling on the flutter of such ***panels*** . the ***panels*** supporting structure allowed partial ***thermal*** expansion of the skins in both the longitudinal and lateral directions . panel skin material and skin thickness were varied . a boundary faired through the experimental flutter points consisted of a flat-panel portion, a buckled-panel portion, and a transition point, at the intersection of the two boundaries, where a panel is most susceptible to flutter . the flutter region consisted of two fairly distinct sections, a large-amplitude flutter region and a small-amplitude flutter region . the results show that an increase in panel skin ***temperature*** flutter . the flutter trend for buckled ***panels*** is reversed . use of a modified ***temperature*** parameter, which approximately accounts for the effects of differential pressure and variations in panel skin material and skin thickness, reduced the scatter in the data which resulted when these effects were neglected . the results are compared with an exact theory for clamped ***panels*** for the condition of zero midplane stress . in addition, a two-mode /transtability/ solution for clamped ***panels*** is compared with the experimentally determined transition point ."

- Explanation : LSA performed better because it captured higher order associations between the words.

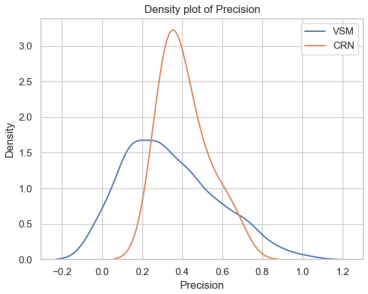
For example, in doc 766, the word ***thermal*** and ***temperature*** are related to heating in the query. As a result, it identified the relevant document, But in VSM, it can't retrieve document based on higher order association

6.2 CRN

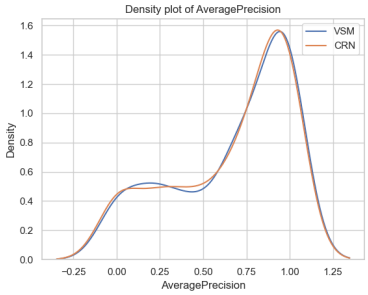
From the distributional graphs, we observe that CRN shifts the distribution slightly towards right and has become somewhat smoother. This implies that Document Representation is improved using Latent Semantic Analysis.Here are a few Examples where the CRN performed better than baseline model

- query id : 109
- query 174 : 'panels subjected to aerodynamic heating .'
- relevant documents : [860, 861, 606, 980, 12, 766].
- documents retrieved by VSM: [1008 , 859, 658 ,864 , 856, 857, 627 , 391 ,766 , 858]
- documents retrieved by CRN :[1008 , 391 , 31 , 766 , 859 , 856 , 864, 66 , 51 , 627]

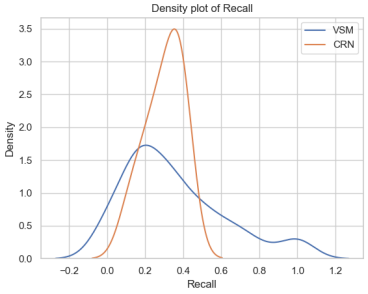
Doc 766 : "experimental investigation at mach number of 3. 0 of effects of ***thermal*** stress and buckling on flutter characteristics of flat single-bay ***panels*** of length-width ratio 0. 96 . flat, single-bay, skin stiffener ***panels*** with length-width ratios of 0.96 were tested at a mach number of 3.0, at dynamic pressures



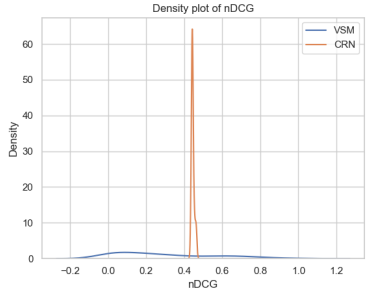
(a) CRN vs. Precision @6



(b) VSM vs. CRN Avg Precision @6



(c) CRN vs. VSM Recall @6



(d) VSM vs. CRN NDCG @6

Fig. 5: Comparison of CRN and VSM

ranging from 1,500 to stagnation ***temperatures*** from 300 f to effects of ***thermal*** stress and buckling on the flutter of such ***panels*** . the ***panels*** supporting structure allowed partial ***thermal*** expansion of the skins in both the longitudinal and lateral directions . panel skin material and skin thickness were varied . a boundary faired through the experimental flutter points consisted of a flat-panel portion, a buckled-panel portion, and a transition point, at the intersection of the two boundaries, where a panel is most susceptible to flutter . the flutter region consisted of two fairly distinct sections, a large-amplitude flutter region and a small-amplitude flutter region . the results show that an increase in panel skin ***temperature*** flutter . the flutter trend for buckled ***panels*** is reversed . use of a modified ***temperature*** parameter, which approximately accounts for the effects of differential pressure and variations in panel skin material and skin thickness, reduced the scatter in the data which resulted when these effects were neglected . the results are compared with an exact theory for clamped ***panels*** for the condition of zero midplane stress . in addition, a two-mode /transtability/ solution for clamped ***panels*** is compared with the experimentally determined transition point .”

– Explanation : CRN performed better because it captured similarity between the words.

For example, in doc 766, the word ***thermal*** and ***temperature*** are related to ***heating*** in the query. As a result, it identified the relevant document, But in VSM, it can't retrieve document based on similarity

References

1. Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41(6), 391–407.
2. *YouTube*. (n.d.). [Video]. Retrieved from `https://www.youtube.com/watch?v=c7e-D2tmRE0`
3. Hui, J. (n.d.). Machine Learning: Singular Value Decomposition (SVD) Principal Component Analysis (PCA). Retrieved from `https://jonathan-hui.medium.com/machine-learning-singular-value-decomposition-svd-principal-component-analysis-pca-1d45`
4. Bitext. (n.d.). What is the Difference Between Stemming and Lemmatization? Retrieved from `https://www.bitext.com/blog/what-is-the-difference-between-stemming-and-lemmatization/`