Investigating LSA and GVSM as Information Retrieval techniques

Advait Walsangikar - AE17B101 and Avinash Bagali - AE17B110

Indian Institute of Technology, Madras

Abstract. In this report, we investigate the effectiveness Latent Semantic Analysis (LSA) and Generalized Vector Space Model (GVSM) methods for modeling word-relatedness in Information Retrieval (IR) systems and also present a "hybrid" method incorporating the two approaches. We then compare the performance of these and the basic Vector Space Model (VSM) using mean F1-score, nDCG@k and MAP@k as the evaluation metrics.

Keywords: Information Retrieval · Latent Semantic Analysis · Generalized Vector Space Model · TF-IDF

1 Information Retrieval - Introduction

Information retrieval systems are extremely important when relevant information sources need to be identified out of a large ground set of resources.
Text-documents are written in natural language and thus quantifying their relevance is quite challenging. The most basic approach to this is the bag-of-words vector space model where context information for each term in the vocabulary is ignored and the terms are assumed to be orthogonal to each other. This approach thus has problems when queries involve synonyms/hyponyms/antonyms etc. which are general features of natural languages and thus motivates one to look at alternate methods that also incorporate word-relatedness.

Latent Semantic Analysis (LSA) and Generalized Vector Space Model (GVSM) are such methods to incorporate word-relatedness. While LSA can be used for intrinsic knowledge, GVSM can be used for extrinsic knowledge using external databases. We thus decided to observe and compare the performance of these methods individually and propose a hybrid method that could potentially be a better way to incorporate these different methods together.

2 Background

2.1 LSA - Latent Semantic Analysis

Latent Semantic Analysis is an intrinsic method used to establish word similarity. LSA uses SVD (Singular Value Decomposition) to transform the Term-Document matrix into a "Concept" space. The advantage of doing this is that we can express both documents as well as terms in this concept space and can easily establish a concept similarity using a measure like Cosine Similarity.

Suppose X is the term-document matrix with element (i, j) containing the tf-idf value corresponding to term i in document j. Then SVD of X gives us:

$$X = U\Sigma V^T$$

If we take k (less than no. of terms) entries with the highest value in the diagonal matrix Σ, we can reconstruct X upto a degree such that:

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

Now, the document can be effectively represented in a k-dimensional "latent concept" space and in a way, models relationship between terms. A document d and a new query q can be represented in the new space by :

$$d^{'} = \Sigma_k^{-1} U_k^T d$$

$$q^{'} = \Sigma_k^{-1} U_k^T q$$

The similarity between the two can then be established by cosine similarity measure:

$$sim(d, q) = \frac{d.q}{||d||.||q||}$$

2.2 GVSM - Generalized Vector Space Model [1]

In the Simple Vector Space Model, the term vectors were considered to be orthogonal to each other, whereas in reality there often exists a relationship between the terms. The Generalized Vector Space Model builds upon the Plain vector space model by taking word similarity into account.

This is done by converting a n-dimensional vector containing n elements each representing a term into a vector containing $\binom{n}{2}$ elements where each element is a pair-wise semantic relatedness score. In our model, we measure semantic relatedness between two terms t_i and t_j using WordNet.

$$t_i t_j = SR((ti, tj), (si, sj), O)$$

where, SR is the semantic relatedness measure between terms t_i and t_j that belong to synsets s_i and s_j respectively, in Thesaurus O and we get similarity betweeen document and query as

$$cos(d_k, q) = \frac{\sum_{i=1}^{n}\sum_{j=i}^{n} d_k(t_i, t_j)q(t_i, t_j)}{\sqrt{\sum_{i=1}^{n}\sum_{j=i}^{n} d_k(t_i, t_j)^2}\sqrt{\sum_{i=1}^{n}\sum_{j=i}^{n} q(t_i, t_j)^2}}$$

where, $d(t_i, t_j) = (tf\text{-}idf(t_i) + tf\text{-}idf(t_j)).SR(t_i, t_j)$

The performance of the IR system will also depend on the SR measure we choose.

3 Methodology

We explore three different models to build a information retrieval system on the Cranfield dataset

1. LSA
2. GVSM
3. Hybrid

Models

LSA We create the term document matrix from Cranfield dataset. The term document matrix's shape is $(N_{docs} * N_{terms})$ and each of it's entries consists of the tf-idf score for the doc-term pair. We perform singular value decomposition on this matrix and transform the documents and the queries (which can be seen as small documents)to the concept space. We then calculate the cosine similarity between the document and a given query and rank the documents in order of decreasing cosine similarity.

GVSM Given a vocabulary consisting of n unique terms, in GVSM we create a new document vector consisting of n*(n-1)/2 elements, where each element represents the pair wise Semantic relatedness of the n terms in the document. (We use the lch-similarity measure calculated on WordNet as a measure of the Semantic Relatedness.) We apply the same transformation to the queries and rank the documents based on the cosine similarity with the given query.

Hybrid LSA is an intrinsic method to incorporate word-relatedness whereas, GVSM is an extrinsic method. We combine these methods in a sequential manner. We first build a GVSM model using the Cranfield dataset and apply LSA to the resulting term-document matrix. The reasoning behind this is that we first incorporate external knowledge using GVSM method and get a richer representation for each document, then we apply LSA to convert it to a concept space to model intrinsic relatedness in the dataset itself. This way, we expect to get better document retrievals.

We would like to mention that GVSM and HYBRID have an $O(n^2)$ time and space complexity and thus the experiments were performed only on a small sized dataset.

Evaluation

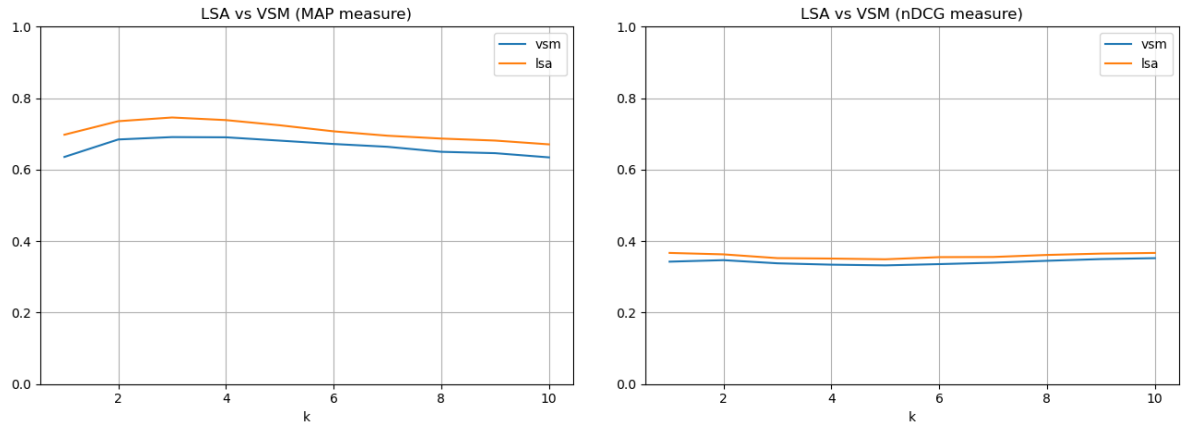
We will use the plain Vector space model as a baseline to compare the performance of each of the models and quantify their performance using the following metrics:

- nDCG @ k (Normalized Discounted Cumulative Gain)
- MAP @ k (Mean Average Precision)
- F1 score @ k

4 Experiments

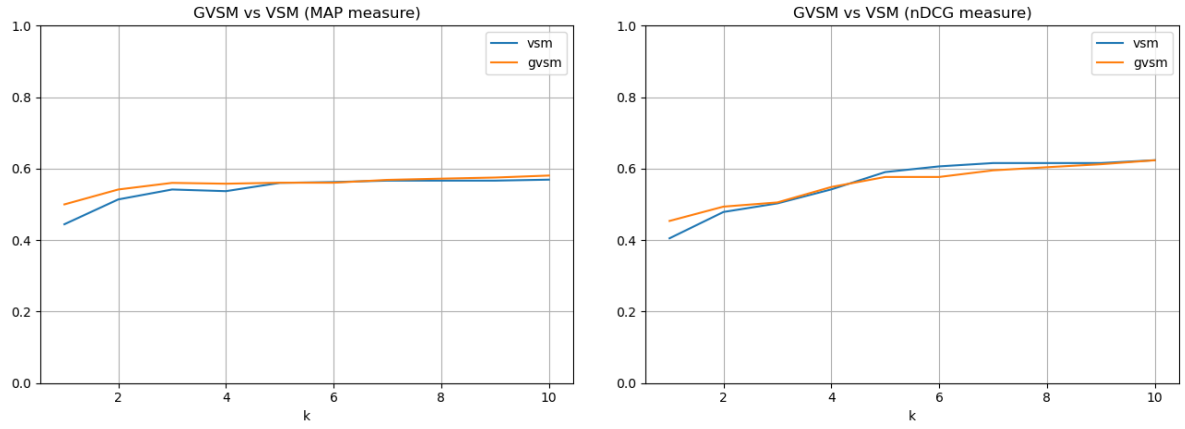
4.1 Experiment I

We obtain NDCG and MAP scores for each GVSM and LSA and compare it with baseline VSM. For LSA vs VSM, we use the full Cranfield dataset containing 1400 documents and 225 queries whereas for GVSM vs VSM, we used a subset of the Cranfield dataset containing 25 documents and 36 queries.



(a) MAP (b) nDCG

Fig. 1: LSA vs VSM



(a) MAP (b) nDCG

Fig. 2: GVSM vs VSM

4.2 Experiment II

We obtain F1, NDCG and MAP scores for GVSM, LSA and HYBRID on a subset of the Cranfield dataset containing 25 documents and 36 queries and compare them together to determine the best performing IR technique.

5 Results

– From the results of Experiment I (Fig. 1, 2), we observe that LSA and GVSM perform better than VSM.

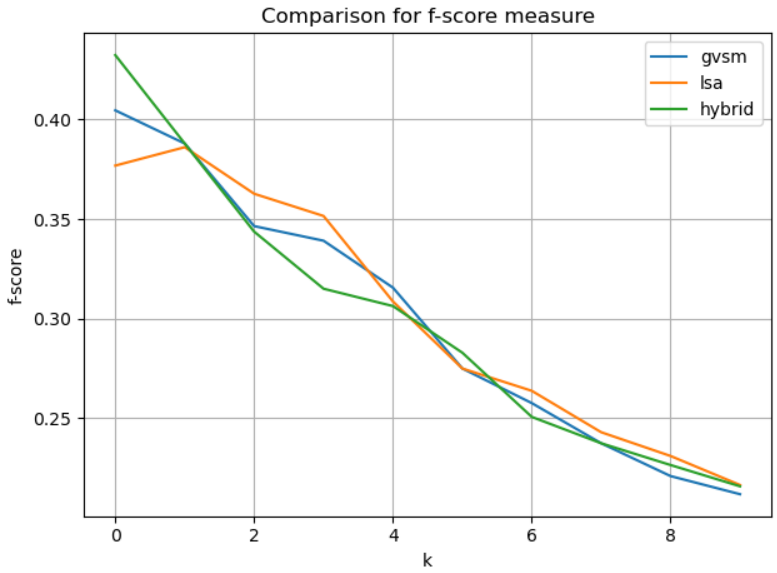


Fig. 3: Comparing GVSM, LSA and Hybrid F-score@k for k = 1:10 on Cranfield Subset

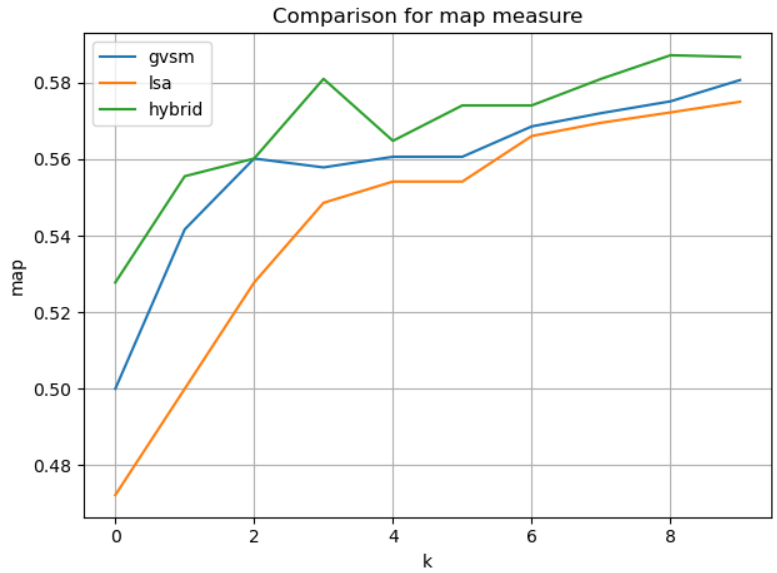


Fig. 4: Comparing GVSM, LSA and Hybrid MAP@k for k = 1:10 on Cranfield Subset

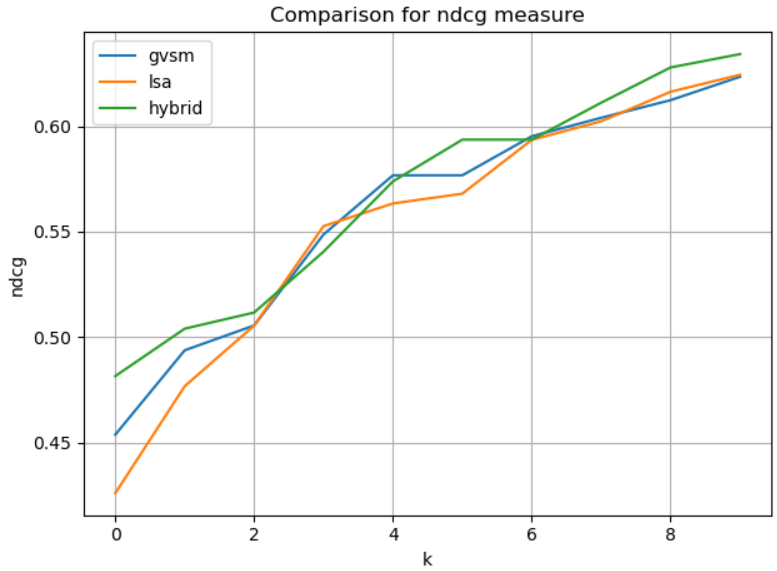


Fig. 5: Comparing GVSM, LSA and Hybrid nDCG@k for k = 1:10 on Cranfield Subset

– If we look at the plots of Experiment II:
  • for the f-score metric (Fig. 3), the performance of all 3 IR methods are comparable and there is no clear winner
  • for MAP metric (Fig. 4), HYBRID > GVSM > LSA.
  • for nDCG metric (Fig. 5), HYBRID > GVSM > LSA.

6 Conclusion

– We observe that Hybrid model performs better than the GVSM and LSA individually, but since the experiment could be performed only on a very small dataset, we cannot say anything about it's general behavior at this moment.
– The computation time to calculate the augmented vector space in the case of GVSM and Hybrid is a big disadvantage. Due to which, in practice LSA might be a better option.

.

References

1. Tsatsaronis, George Panagiotopoulou, Vicky. (2009). A Generalized Vector Space Model for Text Retrieval Based on Semantic Relatedness.. 70-78. 10.3115/1609179.1609188.