

Meta Latent Semantic Analysis

Marin Simina, Costin Barbu

Electrical Engineering and Computer Science Department
Tulane University
New Orleans, LA, 70118, USA
{simina, barbu}@eecs.tulane.edu

Abstract

Meta Latent Semantic Analysis (MLSI) is a novel approach to automated document analysis and indexing which relies on symbolic ontologies to further enhance the traditional probabilistic Latent Semantic Analysis (LSA) of documents. While LSA is able to discover clusters of related terms and documents in a given collection of documents, the proposed MLSI is able to meta-cluster such clusters by taking into account existing symbolic ontologies relevant for the analyzed collections of documents. Such an approach can be successfully used to improve the performance of fast LSI by random projection.

Introduction

A significant problem in information retrieval is developing intelligent user interfaces, which have to recognize the difference between the actual request of an user and what he intended to request. Intelligent retrieval has to address the potential ambivalence, impreciseness and vagueness of user requests. Typically, information is retrieved by literally matching terms in a collection of documents with those present in a user's query. However, such an approach is inaccurate since some words may have many possible meanings (polysemy) and since the same concept may be expressed using several words (synonymy).

Latent Semantic Analysis (LSA) (Deerwester et al. 1990) is an approach to intelligent automatic retrieval which attempts to address the above mentioned problems by using statistically derived indices to represent and retrieve collections of documents and component words in the same latent semantic space. LSA assumes that for any given collection of documents there is an implicit latent structure of word usage, hidden by the variability of words used to express such a structure. LSA, as described in Deerwester et al (1990) and Berry et al. (1995), uses a truncated *Singular Value Decomposition* (SVD) to estimate the structure of word usage across a given collection of documents. LSA then projects documents, words and queries into the same low-dimensional latent space. This facilitates retrieval based on the latent structure

of word usage since documents, which share frequently co-occurring words, will have a similar representation in the latent space even if they do not share common terms. Compared with other traditional methods used in information retrieval, retrieval based on LSA proved to be promising in many applications.

However, existing LSA methods rely purely on probabilistic knowledge and do not take advantage of any prior knowledge about a particular collection of documents. If the collection of documents is extremely large, some prior knowledge will be probably reflected in the LSA indexing, but performing LSA on a large collection of documents is computationally very expensive. If the collection of documents is small LSA will be fast, but a significant part of prior knowledge about the documents may not be reflected in the LSA indexing. Since one way to speed up LSA is to reduce the initial collection of documents by sampling documents and relevant terms (Papadimitriou et al. 1998), the role of previous knowledge in performing an accurate LSA becomes crucial.

The primary goal of this paper is to propose a novel approach to improve the accuracy of LSA by taking advantage of previous symbolic knowledge relevant for a collection of documents. We call our approach *Meta Latent Semantic Analysis* (MLSA), since its role is to refine the purely probabilistic LSA indexing and clustering by using supplementary knowledge about a set of documents. An example of such a supplementary knowledge is WordNet, a lexical ontology system that has been used with relative success to disambiguate word senses in traditional information retrieval (Voorhees 1993). We can imagine that most of the knowledge about related words present in WordNet can be recovered by performing LSA on an infinite collection of documents. While such an approach is not feasible, it suggests the role played by symbolic knowledge for enhancing LSA indexing. If we can incorporate relevant symbolic knowledge into LSA without adding a huge number of documents or words, we can benefit from a similar accuracy as that obtained from performing LSA on an infinite collection of documents. An issue that needs to be addressed when relying on two different sources of knowledge (probabilistic from LSA and symbolic from ontology) is to fine tune their contributions for increasing the accuracy while analyzing a given collection of documents.

The rest of the paper is organized as follows. Section 2 reviews the traditional LSA. Section 3 describes MLSA. Section 4 describes a fine-tuning algorithm for balancing the contribution of probabilistic and symbolic knowledge in MLSA. Section 5 presents the empirical evaluation of MLSA and compares it with traditional LSA. Section 6 presents the conclusions of this work.

Latent Semantic Analysis Review

LSA assumes that each document is represented as an unordered collection of terms (words). LSA over a set of documents requires the construction of an $m \times n$ matrix A of terms by documents. The elements of this matrix represent the occurrences of each term in a particular document:

$$A = [a_{ij}] \quad (1)$$

where a_{ij} represents the number of occurrences of term i in document j . Local and global weightings can be applied (Dumais 1991) to account for the importance of terms within or among documents:

$$a_{ij} = L(i, j) \times G(i) \quad (2)$$

where $L(i, j)$ represents the local weighting for term i in document j and $G(i)$ represents the global weighting for term i . The SVD of matrix A of rank r is given by:

$$A = U \Sigma V^T \quad (3)$$

where $U^T U = V^T V = I_n$ and $\Sigma = \text{diag}(\sigma_1, K, \sigma_n)$, $\sigma_i > 0$ for $1 \leq i \leq r$, $\sigma_j = 0$ for $j \leq r+1$.

If we define:

$$A_k = U_k \Sigma_k V_k^T \quad (4)$$

where $k < r$ and $\Sigma_k = \text{diag}(\sigma_1, K, \sigma_k)$, then it can be proved (Berry et al.1995) that A_k is the best rank- k approximation to A . For the purpose of LSA, U represents term vectors, Σ represents singular values, V represents document vectors, k represents the number of factors and r represents rank of A .

The derived matrix A_k captures the most important correlations among terms and documents, but removes most of the noise and variability in word usage, present in A , that plagues word-based retrieval methods. Since the number of dimensions, k , is much smaller than the number of unique terms, m , minor differences in terminology will be ignored.

In LSA queries must be represented in the same k -dimensional space as documents and terms and the query vector is located at a weighted sum of its constituent term vectors. This process is called “folding-in” queries in the latent semantic space where terms and document documents are represented:

$$\hat{q} = q^T U_k \Sigma_k^{-1} \quad (5)$$

Given a query q , the query vector \hat{q} can be compared to all existing document vectors, based on their similarity, or closeness, to the query. Similarity is usually measured by the cosine between the query vector and the document vectors (Deerwester et al.1990).

Different approaches have been proposed to improve the accuracy of LSA retrieval. Dumais (1992) shows how retrieval accuracy can be improved by using differential term weighting and relevance feedback. Zelikovitz and Hirsch (2001) integrate background knowledge, in the form of supplementary documents, to the initial set of documents used to build the matrix A . However, none of these existing approaches used ontological knowledge, widely used in building intelligent systems, to improve the performance of traditional LSA. In what follows, we describe such a system.

Meta Latent Semantic Analysis

The main method used to incorporate previous knowledge in standard LSA is to add local and global weightings to matrix A to fine-tune the importance of terms within or among documents (Dumais 1991). The issue is how to compute these local and global weightings. We propose a novel method to incorporate previous knowledge about terms by considering generalizations of the terms present in the term/document matrix A . Consider a set of documents that contain information about cars and motorcycles, but where the terms cars and motorcycles do not occur frequently in the same documents. Traditional LSA will place the terms cars and motorcycle close in the latent semantic space only if the documents containing the terms cars and motorcycle contain frequently a common set of other terms. A user, interested in cars as transportation vehicles, may or may not retrieve documents about motorcycles in such a case, especially if transportation is not an indexing term. The result depends entirely on the available set of documents and not on the intent of the user. Moreover, it is not clear in such a case how to automatically compute local and global weights to address similar situations. However, if we find a way to incorporate the knowledge that both cars and motorcycles are vehicles, in all the documents that contain the term car or motorcycle, then the words car, motorcycle and vehicle will be represented closer in the latent semantic space, regardless of how many other irrelevant terms co-occur in these documents. This is the main idea behind MLSA.

MLSA considers all the known generalization terms (GTerms) for a set of indexing terms relevant for a given set of documents. Such a knowledge is available usually in the form of ontologies (e.g., WordNet), and can be conceptually represented as a tree, where any internal node can be viewed as a generalization term of its children. MLSA then adds all the GTerms of a set of indexing terms as virtual terms in the set of indexing terms (see Fig. 1).

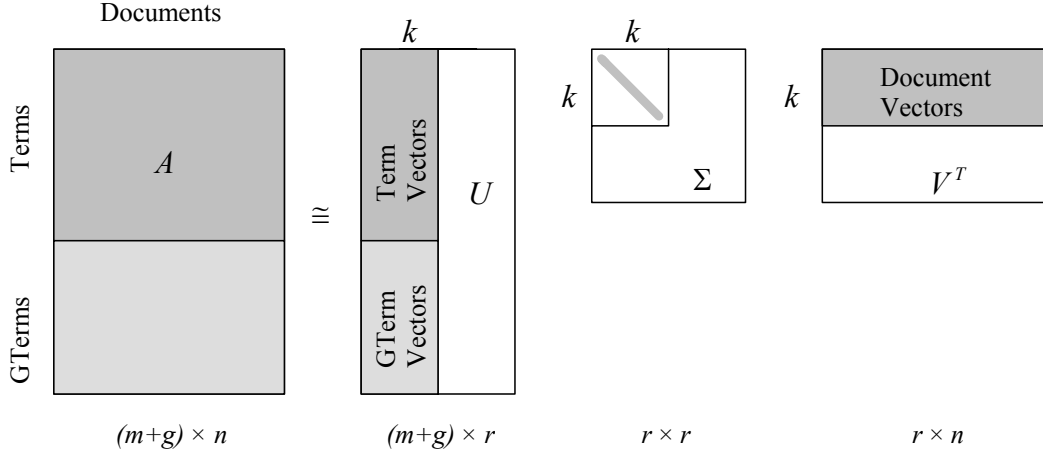


Fig.1. Representation of $A \cong U_k \Sigma_k V_k^T$ for MLSA.

The process may be continued by adding higher order GTerms (GTerms of virtual terms). For each occurrence of a term T_i , having a known Gterm T_l , in a document D_j , the value of the frequency a_{ij} is incremented by one. As a result, all the GTerms become virtual terms in the set of documents. They will modify the latent semantic space by bringing closer all the terms that have the same GTerm. In turn, the answers to a given query will reflect the knowledge about GTerms. The resulting MLSA algorithm is presented in Fig. 2.

function MLSA (Documents, Terms, Ontology)
returns Latent Semantic Space

1. GTerms = all the known generalizations of Terms from Ontology
2. Create a matrix A of (Terms + GTerms) by Documents
 - 2.1. Fill the submatrix Terms by Documents as in standard LSA
 - 2.2. **for each** GTerm T_l **in** GTerms
for each D_j **in** Documents
for each Term T_i **in** Terms
if T_l is a GTerm for T_i **then** increment a_{ij}
3. $[U_k, \Sigma_k \text{ and } V_k] = \text{SVD}(A)$; $A_k = U_k \Sigma_k V_k^T$
4. **return** $[U_k, \Sigma_k \text{ and } V_k]$

Fig.2. The MLSA algorithm

The retrieval algorithm for answering a given query is presented in Fig. 3. It identifies all the GTerms corresponding to a given query and then it performs standard LSI retrieval on a query containing both the initial

query and its corresponding GTerms. It returns all the s nearest neighbors within a predefined similarity angle α (see Fig. 4).

function retrieval (query, $[U_k, \Sigma_k \text{ and } V_k]$, s)
returns closest s nearest neighbors D_j

1. vterms = all indexing GTerms of query Terms
2. q = frequency vector of all Terms and all vterms
3. $\hat{q} = q^T U_k \Sigma_k^{-1}$; fold in query vector in matrix A
4. **return** (s nearest neighbors to \hat{q}) ; cosine similarity

Fig. 3. The MLSA retrieval algorithm

A Fine-Tuning Algorithm

The MLSA algorithm attempts to improve the performance of standard LSA without applying any local or general term weights (see Equation (2)) beside the introduction of virtual terms (GTerms). However, further tuning of term weights a_{ij} may be necessary especially when the distance metric computed by LSA among terms (and documents) is different from the implicit distance metric of the ontology and from the implicit distance in the user's intentions.

But how to quantify user's intentions? We propose the following approach intended to identify both false positives and false negative. Given a query, we will ask the retrieval algorithm to retrieve a larger number of documents than normal (within angle β , not within the predefined angle α , see Fig. 4). Then the user will be asked to label the retrieved documents. The user labels then the documents within angle β relevant to the query

q . For misclassified retrievals, common GTerms between the query and retrieved documents are first identified. Next, the weight of such GTerms is increased for false negatives and decreased for false positives. If we recompute the latent semantic space with updated weights for A , this will bring closer to the query the false negatives and it will move farther the false positives. As a consequence, the number of false positives and false negatives will decrease. The process of weight updating can be repeated iteratively to improve the retrieval capabilities of MLSI. However, this process is expensive since it involves recomputing the latent semantic space every time.

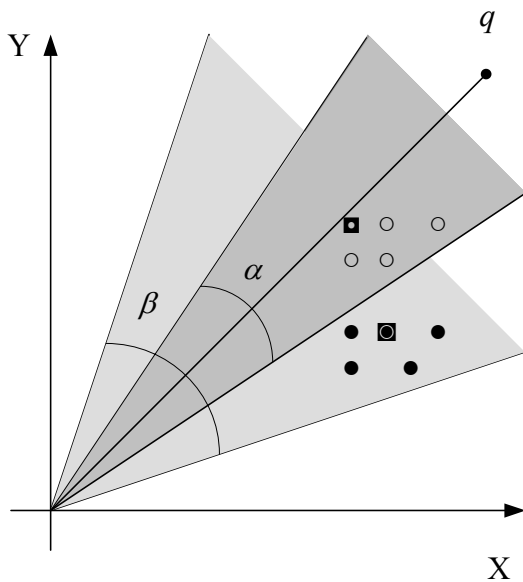


Fig.4. Relevance feedback of D_j given q , where α is the predefined similarity angle.

Legend: \circ = positive, \bullet = negative,
 \blacksquare = false positive, \blacksquare = false negative

Evaluation

In this section we are evaluating the performance of the proposed algorithms on a sample small database of article titles, labeled D1 thru D14 as shown in Table 1. Terms occurring in more than one title are underlined.

The 39 x 14 term-document matrix can be easily built based on the text from Table 1. The elements of this matrix are the term occurrence frequencies of the terms in each of the documents. The rank-2 approximation A_2 is obtained by computing the truncated SVD (with $k=2$) and a two-dimensional plot of the terms and documents is illustrated in Figure 5. For better visualization we shall represent the

same plot in Figure 6, but keeping the documents label and only the terms from the last 3 documents D12, D13, D14. Then we add to the term-documents matrix the generalization terms extracted from the WordNet ontology and their corresponding weights according to the proposed MLSA algorithm and the truncated SVD (with $k=2$) is again computed. The plot shown in Figure 7 clearly illustrate that conceptual related documents are migrating one to another.

D1: Distribution and abundance of antelopes
D2: Manage the deer <u>habitat</u>
D3: Giraffe: <u>habitat</u> , life cycles, <u>feeding</u>
D4: Salamander: evaporative <u>water loss</u> characteristics
D5: Frogs' metabolic depression due to <u>water loss</u> in their <u>habitat</u> .
D6: <u>Feeding captive</u> ostrich
D7: <u>Farming of captive</u> emus
D8: New methods for kiwi <u>farming</u>
D9: Hamsters: the <u>social behavior</u> of these <u>mammals</u>
D10: Squirrels <u>habitat</u>
D11: Beaver: a rare <u>mammal</u>
D12: Koala: <u>habitat</u> , <u>feeding</u> and <u>social behavior</u>
D13: Kangaroo distinguishing features from other <u>mammals</u>
D14: Wombat: another <u>mammal</u> on the endangered species list.

Table 1 Sample database of article titles

Let's consider that the user is typing the query: *kangaroo* and *mammals*. The word "and" is omitted being filtered out by the stop words list and the query consists now only of the words "kangaroo mammals". The retrieval algorithm suggests as candidates the documents D9, D12, D13, D14 (as illustrated in Figure 7).

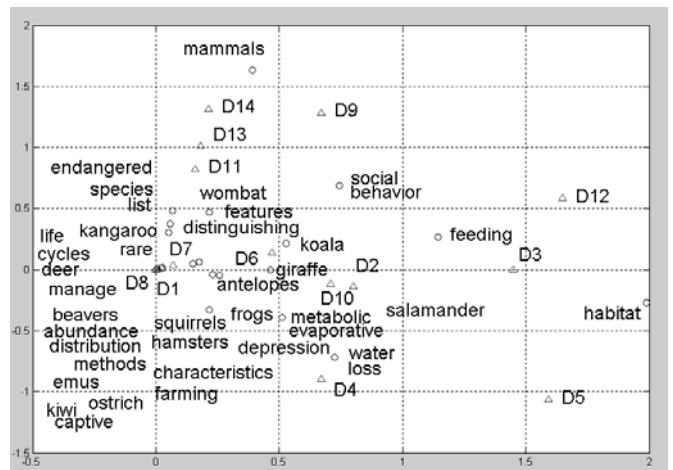


Figure 5 Two – dimensional plot of terms and documents for the 39 x 14 example

The documents D12, D13, D14 are related through the general term marsupial that is the generalization of the words: koala, kangaroo, wombat, and documents D9, D13, D14 are related by the term mammals. A much visible “migration” and “cluster” formation of the documents occur when the weights tuning algorithm is applied as shown in Figure 8. This time the retrieval algorithm indicates as candidates the documents D9, D10, D11, D12, D13, D14.

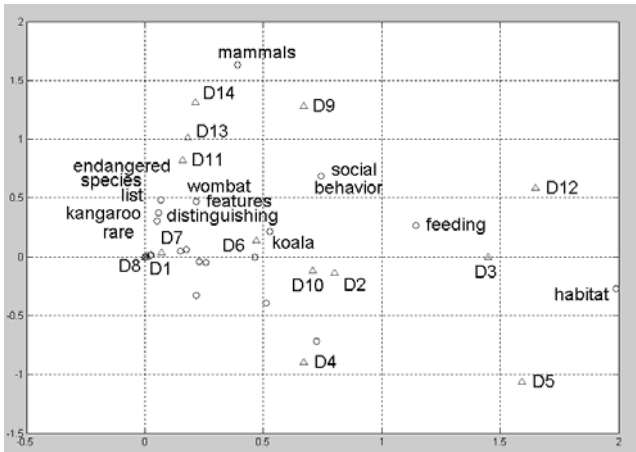


Figure 6 Two – dimensional plot of terms and documents for the 39 x 14 example, with labels of terms from documents D12, D13, D14, only

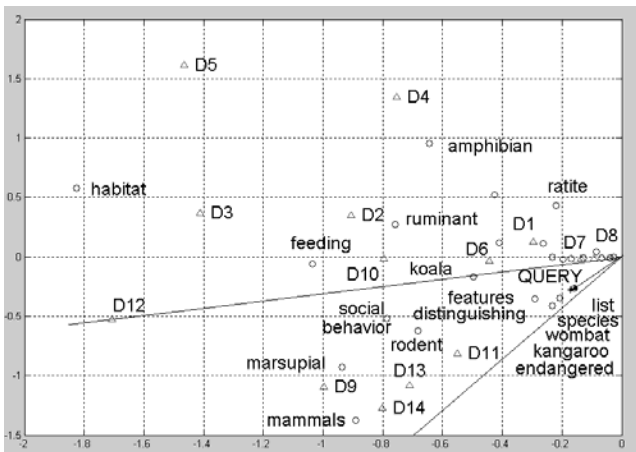


Figure 7 Two – dimensional plot of terms and documents for the 44 x 14 example, after the generalized terms have been added

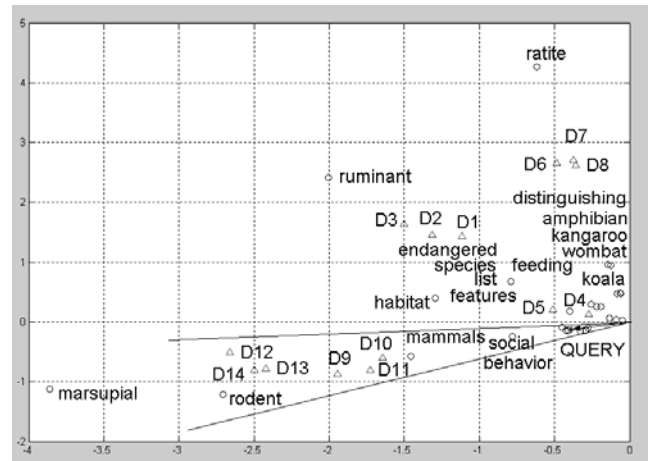


Figure 8 Two – dimensional plot of terms and documents for the 44 x 14 example, after the weights of generalized terms have been tuned/updated

One can observe that the documents D9, D10, D11 are correlated through the general term rodent which is the generalization of the terms: hamster, squirrel and beaver. On the other hand the documents D9, D11, D13 and D14 have a common word (mammal), documents D9 and D12 have two terms in common (social behavior), documents D10 and D12 have one word in common (habitat).

Conclusions

The MLSA algorithm presented in this paper represents a new approach to incorporate symbolic ontologies in the standard LSA. The introduction of virtual GTerms for a set of Terms, increases their covariance and consequently changes the latent semantic space for the updated matrix A updated. In the updated latent space, the set of terms corresponding to a GTerm will be placed closer, according to a cosine-based distance. As a consequence, documents containing different Terms, but the same GTerms, will be placed closer in the updated latent semantic space. Given a query, this will facilitate LSA retrieval of “similar” documents, where similarity depends both on probabilistic knowledge, as used by traditional LSI, and on symbolic ontological knowledge.

References

Bartell, B.T., Cottrell, G.W. and Belew, R.K. 1995. “Representing Documents Using an Explicit Model of Their Similarities,” *J. Amer. Soc. Info. Sci.*, 46, 251-271.
 Berry, M.W., Dumais, S.T., and Letsche, T.A. 1995. “Computational methods for intelligent information

access,” in *Proc. of the 1995 ACM/IEEE conference on Supercomputing (CDROM)*, San Diego.

Brookstein, A. 1986. “Performance of Self-taught Documents: exploiting co-relevance structure in a document collection,” in *Proc. of the ACM SIGIR*, pp. 244-248.

Deerwester, S. C., Dumais, S. T., Landauer, T.K., Furnas, G.W. and Harshman, R.A. 1990. “Indexing by latent semantic analysis,” in *Journal of the American Society of Information Science*, vol. 41, pp. 391-407.

Dumais, S.T. 1991. “Improving the retrieval of information from external sources,” in *Behavior Research Methods, Instruments, & Computers*, vol. 23, pp. 229-236.

Papadimitriou, C.H. Raghavan, P., Tamaki H. and Vempala, S. 1998. “Latent semantic indexing: a probabilistic analysis,” in *Proc. of the 17th ACM Symposium on Principles of Database Systems*.

Voorhees, E.M. 1993. “Using WordNet to Disambiguate Word Senses for Text Retrieval,” in *Proceedings of the 16th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, 1993*, 171 -180, Pittsburgh, PA.

Zelikovitz, S. and Hirsh, H. 2001. “Using LSI for text classification in the presence of background text, in *Proc. of CIKM-01, 10th ACM International Conference on Information and Knowledge Management*.