Bibliographic Latent Semantic Analysis

Xiangming Mu, Jun Zhang, Wen Hu, Tian Zhao, Mark Fredette, and Guangwu Xu

University of Wisconsin-Milwaukee
3210 N Maryland AV, Milwaukee, WI 53211
{mux,junzhang,whu,tzhao,mfredette,gxu4uwm}@uwm.edu

Abstract

In some cases, when users conduct a search, they not only want the retrieved documents to be relevant to the query, but also the documents to be relevant to each other. To address such expectation, we propose a new retrieval method-- Bibliographic Latent Semantic Analysis (BLSA) -- to automatically capture and build inter-links between two documents sharing the same bibliographic information such as authorship. BLSA expands the Latent Semantic Analysis (LSA)'s term-document matrix by including bibliographic information before the Singular Value Decomposition (SVD) processing. The primary advantage of applying BLSA in information retrieval is that, under certain circumstances, BLSA enhances the correlations between documents sharing the same bibliographic information. As a result, these documents will appear closer on the retrieval return list. Two case studies are given to illustrate the implementation of BLSA in real applications and to compare its performances with that of LSA and Vector Space Model (VSM).

Keyword: LSA, inter-links, BLSA, information retrieval

I. Introduction

Bibliographic information is critical for users searching for a book, journal paper, or an electronic document. Using the advanced search functions provided by Open Public Access Catalog (OPAC) systems or database search systems such as EBSCOHost, ProQuest, or Gale, users are able to narrow down the scope of their search and create a more relevant results list. For example, if a user provides a name in the author field and combines it with the subject search, only documents meeting the subject criteria and authorized by that author will be retrieved.

The challenge, however, is that general users are often presented with a simple search interface, like Google. This interface ignores advanced search features and explicit Boolean searches that may aide users in finding relevant information along multiple bibliographic fields. The trend of using simple, one-box search interfaces seems to only be growing, which makes it imperative for systems to develop ways that encourage user-system interaction so users can find other relevant information.

In practice, bibliographic information linking is widely utilized by commercial websites such as Amazon.com or Netflix.com for recommendations by suggesting related books or DVDs. The assumption here is that documents sharing the same bibliographic information are more likely to be relevant to each other than those that do not. The dilemma is that users may not engage in multiple interactions with the search results to discover material related by bibliographic information [3].

Since the catalog-based search systems have their own advantages and for some tasks the systems are irreplaceable [15], we need to find an alternative approach to supplement users' search

with bibliographic information. In this paper we present a new indexing method, Bibliographic Latent Semantic Analysis (BLSA), which automatically detects and captures the document inter-links created by sharing the same bibliographic information. It then incorporates them directly into the search results. BLSA differs from commercial recommendation features because the latent integration requires no extra clicks or multiple item selections.

After a brief introduction of related research, we will explain the details of our Bibliographic Latent Semantic Analysis (BLSA) indexing scheme. Then, we will present two case studies to illustrate how BLSA is applied in an information retrieval application.

II. Related Works

Information seeking, according to Marchionini, is a fundamental human behavior involving the interactions between users and information systems [9]. A successful search needs the "collaboration" between an IR system and users.

Guided by user-centered IR models, many studies have investigated how to develop appropriate user interfaces to encourage users' engagement in user-system interactions. After studying transaction logs obtained from four different web-based IR systems, Wolfram found that, on average, only about two queries were entered for each search session [14]. Belkin et al. found that the layout and size of the search box had an impact on the length of users' query. A large input box with multiple rows or a longer search field was correlated to a lengthy user query, and in turn, would lead to a search result list that clusters linked documents together [1]. However, user behavior studies suggested that general users are reluctant to engage in multiple-round interactions with search results [14].

Another research direction is to work on the system side that does not require users' extra input. One important research is the application of mixture models in IR—viewing documents as mixtures of a set of semantic topics. Latent Semantic Analysis (LSA) is a pioneer work in this area. By approximately reflecting the term-document matrix to a reduced-dimension space via Singular Value Decomposition (SVD), the LSA indexing creates the topics of a document [5]. After removing lower eigenvalue dimensions, the reset salient latent semantic topics are utilized for document relevance ranking. The main problems for LSA, however, are its computation complexity and lack of scalability, particularly when data collection is large.

Hofmann proposed a probabilistic approach to attempt to solve the problems and named it Probability LSA, or pLSA --- the probability of a word in the indexed vocabulary for a specific document can be represented by a mixture of probabilities distributed across a set of latent aspects [8]. These aspects reflect the latent semantic topics of the document and can be estimated by the Expectation-Maximization (EM) algorithm. Hofmann provided studies on the test collections of MED, CRAN, CACM, and CISI to demonstrate the advantage of pLSA over LSA in terms of precision. With the pLSA, major obstacles include the growing number of parameters with increasing corpus size and the unclear probability definition for documents outside of the training set [2].

Latent Dirichlet Allocation, or LDA, was introduced by Blei et al. in an attempt to solve the problems of pLSA. LDA is a three-level hierarchical Bayesian model that represents each word in a collection as a finite mixture over a set of latent topics, which in turn are represented as a finite mixture over a set of topic probabilities. In LDA, variational EM algorithm and approximate inference techniques are used for parameter estimation. Experimental studies on document modeling, text classification, and collaborative filtering examples were given by the authors and their results were compared with both the unigrams and the pLSA model [2].

Targeting the LDA's computation complexity when converging the parameters, Griffiths and Steyvers provided a simple inference method based on the Markov chain Monte Carlo technique, or

Gibbs sampling. They also presented an example of finding scientific topics from abstracts of papers published in PNAS from 1991 to 2001 to demonstrate an application of using Gibbs sampling [7].

The key concept that lies behind LSA, pLSA, and LDA is to provide a global analysis of the data corpus to elicit major "topics". In other words, these models build semantic connections between documents via the statistical analysis based on the "bag of words" assumption [7]. Such a statistical analysis approach has been successfully applied in automatic classifications to help to build semantic clusters (for example, see [12]). New LDA based algorithm has been proposed to further reduce the computation cost and improve the performance [11]. Furthermore, studies indicated that this technique can be applied in an array of application domains such as medical image clustering [10].

However, the problem of how to take advantage of the valuable bibliographic information to strengthen the document semantic analysis, and thus enhance the information retrieval effectiveness, is still not fully addressed in these studies. In this paper, we propose a bibliographic LSA (BLSA) indexing method to attempt to bridge the gap.

III. Bibliographic Latent Semantic Analysis (BLSA)

The relation of retrieved documents may be important to users under certain circumstances--it is likely that a user could be looking for information that is not directly related to the query, but is similar in some way. For example, if a retrieved document is judged as relevant, other documents from the same author might also be of interest. The assumption here is that if a document shares the same authorship with a relevant document, its probability of relevance is higher than a randomly selected document from the collection. This assumption can also be expanded to other bibliographic information. For example, two documents sharing same subject keywords should be more relevant to each other than two randomly selected documents.

Based on these assumptions, we propose the Bibliographic Latent Semantic Analysis (BLSA) approach to detect the latent document bibliographic links and incorporate them directly into the document indexing. Different from the commercial recommendation applications, for BLSA, there are no extra clicks or second round selections needed.

We predict that our new indexing approach will be able to "cluster" together documents linked by sharing bibliographic information on the result list (proof is given in a later section). In other words, when a document is picked as relevant, other documents presented around it are likely to be of interest.

A. *Latent Semantic Analysis (LSA)*

Latent Semantic Analysis is an indexing and ranking algorithm that employs Singular Value Decomposition (SVD) to find salient latent semantic topics by removing trivial "noise" [5]. Let T_{nm} be a term frequency matrix where the element (n,m) describes the frequency of term <*n*> in document <*m*>. Using SVD, T_{nm} can be decomposed into eigenvector matrix U_n and V_m^T. Here $\sum_r$ is a diagonal matrix containing the eigenvalues of T_{nm}, and r is the rank of T_{nm}

$$T_{nm} = U_n \Sigma_r V_m^T \qquad (1)$$

If we just choose the top k eigenvalues, then T_{nm} can be approximately represented by a new matrix $\widehat{T}_{nm}$.

$$\widehat{T}_{nm} = \hat{U}_n \hat{\Sigma}_k \hat{V}_m^T \cong T_{nm} = U_n \Sigma_r V_m^T \qquad (2)$$

The value of k can be chosen based on the tradeoff between truncation error and size [6]. After the transition, the original *n* dimensional term space is approximately reflected into a k dimensional space. In the new space, each dimension represents a latent semantic topic and its weight is represented by the corresponding eigenvalue in $\sum_k$. The top k latent semantic topics are selected for indexing and relevance ranking. The rest of the (r-k) dimensions are filtered out as "noise".

B. *Bibliographic Latent Semantic Analysis (BLSA)*

BLSA uses the same algorithm as LSA, but with bibliographic vectors added. In other words, additional bibliographic vectors are integrated into the term frequency matrix before the Singular Value Decomposition (SVD) process. Let E_i be a vector ($e_{i1}, e_{i2}, e_{i3}, \ldots, e_{im}$) and let e_{ik} denotes the weight of a specific bibliographic character <*i*> in the <*k*> document. There are total *m* documents to be indexed. The value of e_{ik} can be defined as

$$e_{ik} = \begin{cases} w_i & contains\ i\ feature \\ 0 & not\ contains\ i\ feature \end{cases} \qquad (3)$$

where w_i will be decided based on the average document length. The larger the value of w_i, the more weight will be counted for this <*i*> bibliographic character in the indexing. For the term frequency matrix T_{nm}

$$T_{nm} \rightarrow \begin{bmatrix} t_{1,1} & \cdots & t_{1,m} \\ \vdots & \ddots & \vdots \\ t_{n,1} & \cdots & t_{n,m} \end{bmatrix} \qquad (4)$$

Suppose we have $\tau = (\tilde{n} - n)$ dimension bibliographic characters (e.g., author, keywords) to be integrated into the index scheme. A new extended ($\tilde{n}$, *m*) matrix $\tilde{T}_{\tilde{n}m}$ can be built

$$\tilde{T}_{\tilde{n}m} \rightarrow \begin{bmatrix} t_{1,1} \ldots t_{1,m} \\ \vdots \quad \ddots \quad \vdots \\ t_{n,1} \ldots t_{n,m} \\ e_{n+1,1} \ldots e_{n+1,m} \\ \vdots \quad \ddots \quad \vdots \\ e_{\tilde{n},1} \ldots \quad e_{\tilde{n},m} \end{bmatrix} \qquad (5)$$

Here the sub-matrix $E_{\tau,m}$ contains the bibliographic vectors. Now we can apply SVD to the new matrix $\tilde{T}_{\tilde{n}m}$ to obtain the new top *k* eigenvalues.

$$\hat{\tilde{T}}_{\tilde{n}m} = \hat{\tilde{U}}_{\tilde{n}} \hat{\Sigma}_k \hat{\tilde{V}}_m^{\ T} \cong \tilde{T}_{\tilde{n}m} = \tilde{U}_{\tilde{n}} \hat{\Sigma}_r \tilde{V}_m^{\ T} \qquad (6)$$

The new *k* dimensional index matrix $\hat{\tilde{T}}_{\tilde{n}m}$ integrates the bibliographic links of documents via SVD transition. Accordingly, document rankings per user query will be computed based on the new matrix.

In table 1, we summarized the differences of BLSA, LSA, PLSA, and LDA algorithms in terms of indexing matrix, computing cost, meaning of k, processing method, and scalability.

Table 1: A comparison of LSA, pLSA, LDA, and BLSA

	LSA	pLSA	LDA	BLSA
Indexing Matrix	Term Frequency	Term Frequency	Term Frequency	Augmented Term Frequency
Computing Cost	Expensive	Moderate Expensive	Moderate Expensive	Expensive
Meaning of k	Not meaningful	Meaningful	Meaningful	Not Meaningful
Processing method	SVD	EM	EM	SVD
Scalability	Weak	Moderate	Strong	Weak

IV. Correlation Analysis

Traditional retrieval performance evaluation tools such as precision, recall, average precision and Mean Average Precision (MAP) provide ways of assessing the effectiveness of an IR approach. But these tools do not specify how and to what extent the search results are related to each other. As compared to LSA, we believe that the BLSA indexing scheme provides stronger connections among documents when they share bibliographic information as links. Such connections within BLSA will turn a query's return list more cohesive. Instead, we provide a mathematic proof via the theoretical correlation analysis.

We first consider how augmenting term-frequency vectors with additional information changes their correlation.

Let **x** and **y** be two term frequency column vectors, representing two arbitrary documents in a corpus. Then, their correlation coefficient (cosine of the angle between them) is

$$c(\mathbf{x},\mathbf{y}) = \frac{\mathbf{x}\cdot\mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|}$$

where the dot "." represents the vector inner product and "|| ||" represents the norm (length). For the sake of simplicity, consider a very simple way to augment term-frequency vectors: using weighted numbers representing author information. This way, **x** is augmented to

$$\mathbf{x}_a = [\mathbf{x}', xa_1, ..., xa_k]'$$

Here ' represents transpose, k is the total number of different authors, and for each i, $xa_i = w > 0$ if the i author is an author of **x**, and 0 otherwise. The vector **y** can be augmented in a similar way.

Now assume the sets of authors in **x** and **y** do not intersect. Then, the correlation coefficient between the augmented **x** and **y** is

$$c(\mathbf{x}_a,\mathbf{y}_a) = \frac{\mathbf{x}_a\cdot\mathbf{y}_a}{\|\mathbf{x}_a\|\|\mathbf{y}_a\|} = \frac{\mathbf{x}\cdot\mathbf{y}}{\|\mathbf{x}_a\|\|\mathbf{y}_a\|} < \frac{\mathbf{x}\cdot\mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|} \quad (7)$$

That is, after augmentation, the correlation between documents that do not have authors in common declines.

Now we consider the case when **x** and **y** do have some authors in common. First, consider the case that they have only one author in common (actually, this is the most difficult case to prove our results). Furthermore, assume that on average, a document has k authors, where k is a relatively small integer. After author augmentation, the document vector becomes **x**' and **y**' and the square of their correlation coefficient (cosine) becomes

$$\begin{aligned} c(\mathbf{x}',\mathbf{y}')^2 &= \frac{(\mathbf{x}'\cdot\mathbf{y}')^2}{\|\mathbf{x}'\|^2\|\mathbf{y}'\|^2} = \frac{(\mathbf{x}\cdot\mathbf{y}+w^2)^2}{(\|\mathbf{x}\|^2+kw^2)(\|\mathbf{y}\|^2+kw^2)} \\ &= \frac{(\mathbf{x}\cdot\mathbf{y})^2}{\|\mathbf{x}\|^2\|\mathbf{y}\|^2}\cdot\frac{(1+w^2/(\mathbf{x}\cdot\mathbf{y}))^2}{(1+kw^2/\|\mathbf{x}\|^2)(1+kw^2/\|\mathbf{y}\|^2)} \\ &= c(\mathbf{x},\mathbf{y})^2\cdot\frac{1+2w^2/(\mathbf{x}\cdot\mathbf{y})+w^4(\mathbf{x}\cdot\mathbf{y})^2}{1+kw^2(1/\|\mathbf{x}\|^2+1/\|\mathbf{y}\|^2)+k^2w^4/(\|\mathbf{x}\|^2\|\mathbf{y}\|^2)} \end{aligned}$$

Now, we want to find a condition under which the second term is greater than 1. Specifically, in order to have

$$\frac{1+2w^2/(\mathbf{x}\cdot\mathbf{y})+w^4/(\mathbf{x}\cdot\mathbf{y})^2}{1+kw^2(1/\|\mathbf{x}\|^2+1/\|\mathbf{y}\|^2)+k^2w^4/(\|\mathbf{x}\|^2\|\mathbf{y}\|^2)}\geq 1$$

we need to have

$$1+2w^2/(\mathbf{x}\cdot\mathbf{y})+w^4/(\mathbf{x}\cdot\mathbf{y})^2\geq 1+kw^2(1/\|\mathbf{x}\|^2+1/\|\mathbf{y}\|^2)+k^2w^4/(\|\mathbf{x}\|^2\|\mathbf{y}\|^2) \quad (8)$$

If we have

$$2w^2/(\mathbf{x}\cdot\mathbf{y})\geq kw^2(1/\|\mathbf{x}\|^2+1/\|\mathbf{y}\|^2) \quad (9)$$

that is

$$k(\mathbf{x}\cdot\mathbf{y})\leq \frac{2\|\mathbf{x}\|^2\|\mathbf{y}\|^2}{\|\mathbf{x}\|^2+\|\mathbf{y}\|^2} \quad (10)$$

then in the inequality of (8), the second term on the left-hand side will be equal to or greater than the second term on the right-hand side.

Similarly, to have the third term on the left hand side of inequality (8) to be greater or equal to the third term on the right hand side, we need to have

$$w^4/(\mathbf{x}\cdot\mathbf{y})^2\geq k^2w^4/(\|\mathbf{x}\|^2\|\mathbf{y}\|^2) \quad (11)$$

that is

$$k(\mathbf{x}\cdot\mathbf{y})\leq\|\mathbf{x}\|\cdot\|\mathbf{y}\| \quad (12)$$

However, this is already implied by the inequality (10). To see this, notice that from (10), we have

$$k(\mathbf{x}\cdot\mathbf{y})\leq \frac{2\|\mathbf{x}\|^2\|\mathbf{y}\|^2}{\|\mathbf{x}\|^2+\|\mathbf{y}\|^2}=\|\mathbf{x}\|\cdot\|\mathbf{y}\|\cdot\frac{2\|\mathbf{x}\|\|\mathbf{y}\|}{\|\mathbf{x}\|^2+\|\mathbf{y}\|^2}\leq\|\mathbf{x}\|\cdot\|\mathbf{y}\| \quad (13)$$

where we used a well-known inequality $2ab\leq a^2+b^2$. Based on the analysis above, if condition (10) is satisfied, we will have

$$[c(\mathbf{x}',\mathbf{y}')]^2\geq[c(\mathbf{x},\mathbf{y})]^2$$

Since both x and y are term frequency vectors and are non-negative, this implies

$$c(\mathbf{x}',\mathbf{y}')\geq c(\mathbf{x},\mathbf{y})$$

This suggests that under appropriate conditions, the correlation between documents that share the same authors can increase after augmentation. Furthermore, the condition of (10) is not very restrictive. For example, suppose a typical document has 10^3 words. Then right hand side of (10) is on the order of 10^3 and the inequality is equivalent to

$$k\leq\frac{10^3}{\mathbf{x}\cdot\mathbf{y}} \quad (14)$$

If the documents have 1/4 words in common (quite large), i.e., $\mathbf{x}\cdot\mathbf{y}=10^3/4$, we have

$$k\leq 4 \quad (15)$$

That is, the number of shared authors is no more than 4. From this simple example, we can also see that the lower the correlation between the two documents is, the more likely that the augmentation will increase their correlation.

Next we will present two case studies to demonstrate how BLSA is implemented in a real case retrieval system. The goal for these experiments is not to draw any significant conclusions

about the BLSA's merits but to demonstrate an application of the BLSA in real-world IR cases. The methodology, however, can be expanded to a large test collection.

V. Case Studies

Case One: Faculty publication dataset

We built a small data set by collecting 272 citations of faculty publications from a well-known university's library and information science school. Each citation includes the title of the publication and the author name. There are in total 22 authors. The topics cover a wide range of areas in library and information science (see Table 2 for top ten indexed terms). We implement the BLSA algorithm in MATLAB, which contains a built-in SVD package. We used the Apache Lucene (http://lucene.apache.org/) as our test search engine and replaced the default TF-IDF ranking algorithm with the BLSA. The built-in stoplist and stemming method are also applied.

Table 2: Top 10 most frequent terms after stemming

Terms	*Frequency*
inform	55
librari	48
copyright	30
learn	29
educ	25
ethic	25
digit	23
legal	18
develop	17
distanc	17

Mean Average Precision (MAP)

In this case we used mean average cut-off Mean Average Precision (MAP). Considering the small size of the collection, only top 20 return-hits were used. That is about the average documents read by users based on several transaction log analysis studies [14] and is referred to as MAP@20. In the following section of the paper, for convenience, we still use the term MAP but it is actually the top cutoff MAP not the general MAP.

$$MAP \quad @ \quad 20 \quad = \quad \frac{1}{mn} \sum_{i=1}^{m} \sum_{r=1}^{n} (p(r) \times rel \ (r))$$

Where

m is number of runs conducted
n is the number retrieved
p(r) is the precision for top r document
rel(r) is the relevance judgment (binary value: 1 for relevant and 0 for not relevant)

Queries

Four queries were chosen based on informal interviews with faculty members from the studied library school: two single-word ("video", "archive"), one two-word ("information organization"), and one three-word ("copyright and legal issues") queries. The term "and" here is a stop-word and would not be indexed. For simplicity, all weighting factors in formula (5) are given the value of one.

First, we needed to choose the k value. We tested three truncation *k* values for best LSA performance. Considering the size of index term *n=273,* we decided to use 40, 80, and 160 for testing. Figure 1 depicts the Mean Average Precision for 20 cut-off documents using the three k

values for LSA. We note that there is no significant difference among the three different k values except k=40 is a little bit better at the beginning. So we chose k=40 to compute the MAP@20 scores for both LSA and BLSA. We also tested different k values in the BLSA and confirmed that k=40 is the optimal value.

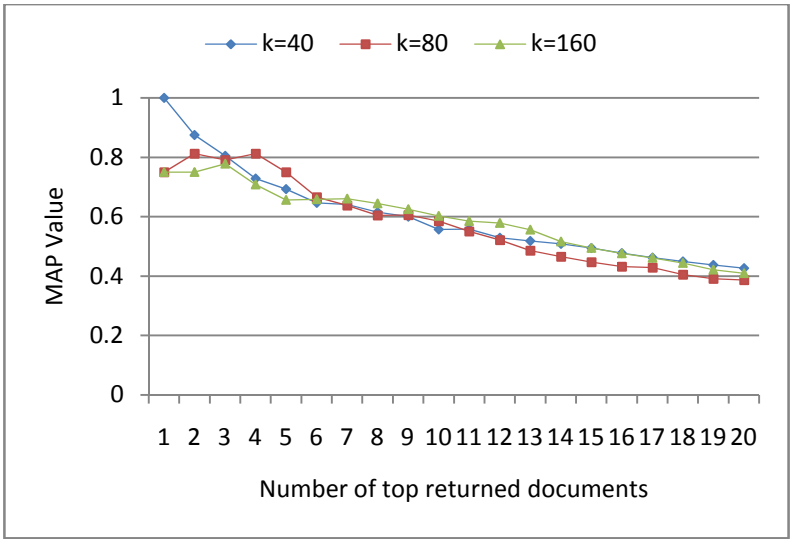


Figure 1: Mean Average Precisions for Top 20 Cut-off (MAP20) with three different k values

Figure 2 shows the MAP@20 curves for BLSA, LSA, and the benchmark Vector Space Model (VSM). We note that BLSA is consistently better than the LSA and VSM. A randomization test (permutation test) with 1000 random runs indicates that BLSA is better than LSA and the results are statistically significant (α=0.015<0.05).

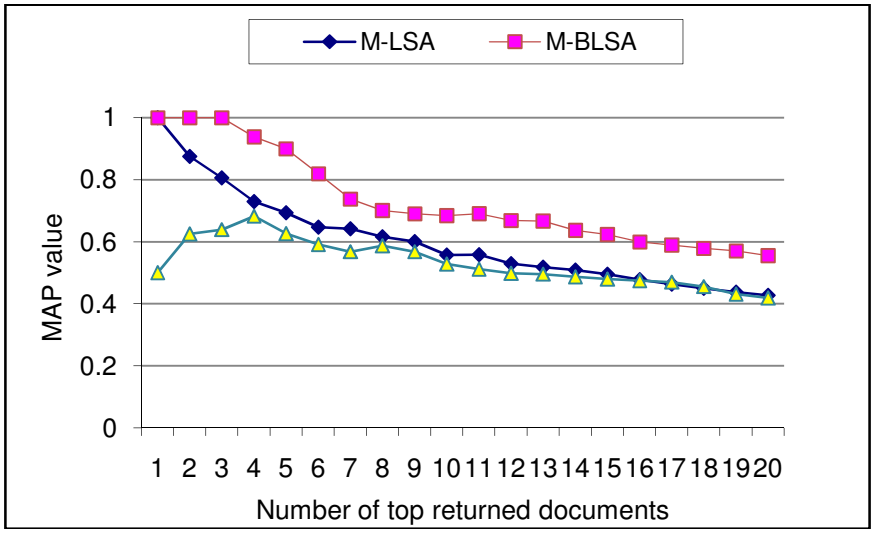


Figure 2: Mean Average Precision for top 20 cut-off (MAP@20) of three algorithms: LSA, BLSA, and VSM (the line with triangles)

Case Two: Rockwell Automation KnowledgeBase

Question-and-answer systems (typically in the form of a FAQ, forum, or knowledge database) are used to help a user or customer to get quick answers to questions not generally addressed in available documentation such as help files or manuals. The Rockwell Automation KnowledgeBase (KB) is one such question-and-answer system (see Figure 3). The KB contains answers to tens of

thousands of customer-related issues about Rockwell Automation's products. Rockwell provides access to the KB so that customers can quickly find answers to their questions without having to call technical support. Since there are too many entries to simply browse for the desired answer, a customer's ability to effectively search the KB is essential to the system's success.

Data

For this case study, we randomly selected a subset of the KB database consisting of approximately one-third of the total documents available in the system. We did not use the entire KB database because of the limitation of LSA processing power; we implemented our BLSA using standard MATLAB SVD package and our experiment computer does not have sufficient memory to process the entire KB database term-document matrix. In the future we can solve this problem by replacing the current SVD software with a new one developed from an improved SVD algorithm. Another solution is to build the BLSA model on top of the pLSA rather than on the LSA [8].

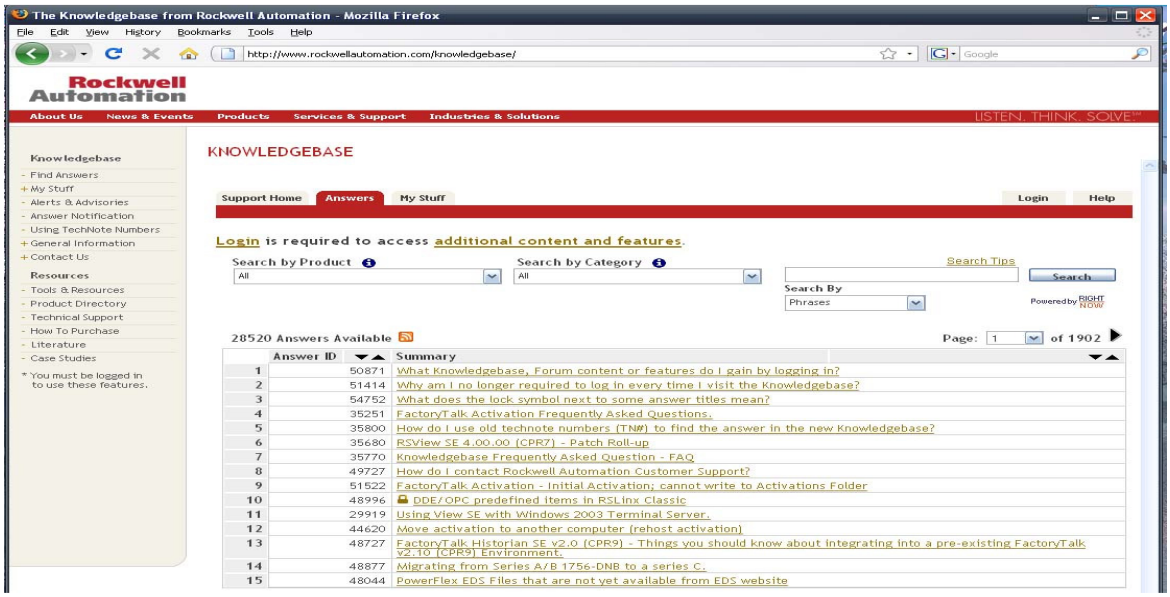


Figure 3: A screen shot of Rockwell Automation's KowledgeBase System Interface

As a result, the data collection contains 4617 documents. Each document has a description summary, a group of keywords, product category or classification, and the text of the answer. The average length of summary is 16 words while the answer part can be a couple words to several hundred words. There are 78 top level product categories and 480 second level product classifications. After indexing, the total number of terms is 2298 for summary and 4874 for answer text.

Queries

Based on the results from the statistical query transaction log analysis of the Rockwell KB system, we obtained the 50 most used queries. Three of these queries (Table 3) were used for testing in this study, including a one-word query: "1794-ADN"; a two-word query: "unrecognized devices", and a multiple-word query: "unable to save tag database". The controlled group is the LSA indexed summary. The BLSA model is constructed with "keywords" as the bibliographic links between documents. One expert from Rockwell with domain knowledge made all the relevance judgments based on objective criteria (see Table 3).

Table 3: Testing queries and relevant judgment

Query	*Relevant Judgment*
1794-ADN	If an entry had the module name (1794-ADN) in the summary, question, or answer fields, then it was relevant. If it did not, then it was not relevant
unrecognized devices	If an entry was related to the error "unrecognized device", which happens when a device is missing its EDS (Electronic Data Sheet), then it was relevant. Otherwise, it was not relevant.
unable to save tag database	If an entry had to do with the error "unable to save to tag database", it was relevant. Otherwise, it was not relevant

Figure 4-5 are Average Precision for top 50 document cut for BLSA and LSA (MAP@50) for one word and for multiple words. We note that BLSA gains a better Average Precision (AP) score; particularly for the one-word query in Figure 4 (the difference is statistically significant). But due to the small data collection and limited number of queries, we have no intention to draw any general conclusions that BLSA has a better AP than LSA.

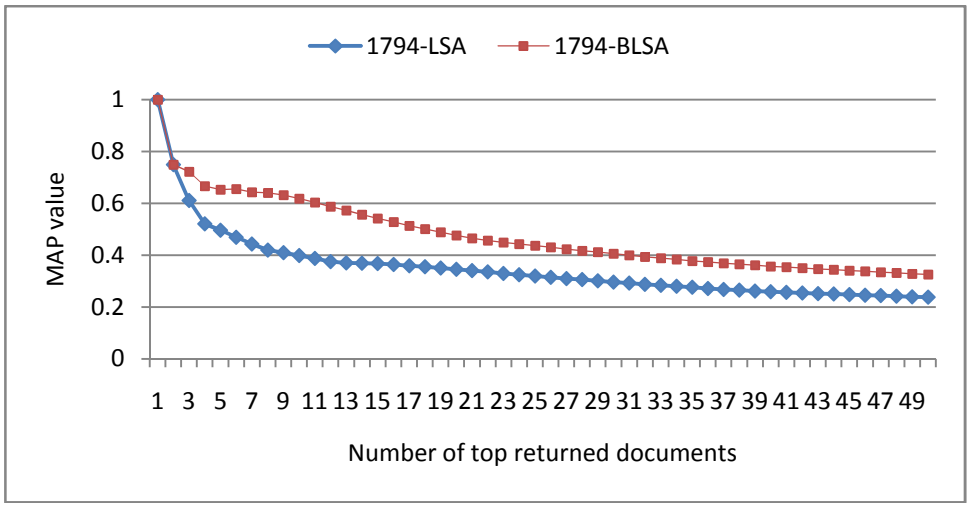


Figure 4: Average Precision for top 50 cut-off returns for LSA and BLSA using one word query

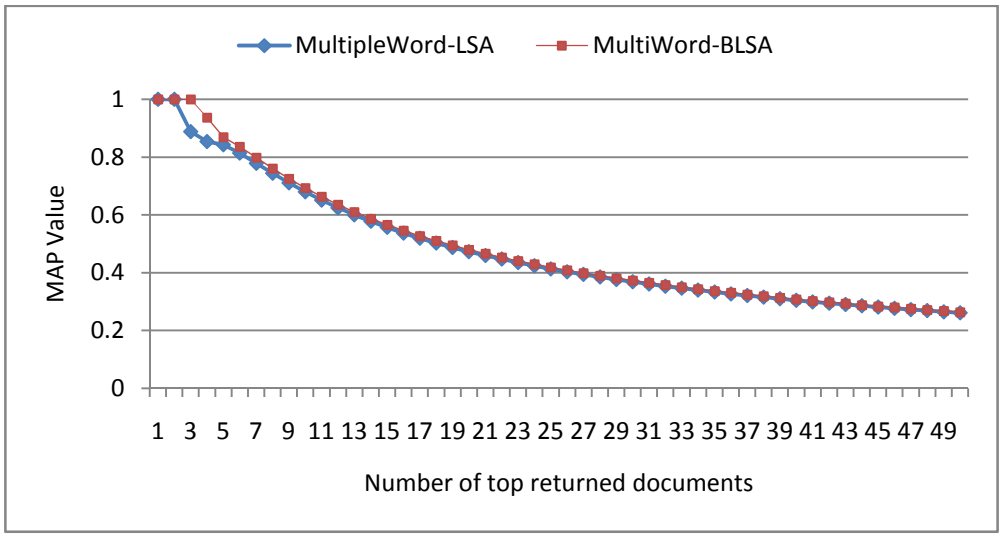


Figure 5: Average Precision for top 50 cut-off returns for LSA and BLSA using multiple-word query

Adjacency Matrix Analysis

One way to determine how related each of the results are to each other, is to employ graph spectral theory for adjacency matrix analysis [4]. In our case, from each list of 50 results returned for a given query for the Rockwell KB data, a 50 by 50 adjacency matrix A is constructed, where $A_{ij} = 1$ if *result-i* is related to *result-j*, otherwise $A_{ij} = 0$. The spectrum of the adjacency matrix is calculated, and the three largest eigenvalues are produced. The sum of the resulting eigenvalues is then used as a measure on how connected the results are to each other. For each of the three sample queries, this sum was calculated for BLSA, traditional LSA, and Vector Space Model (VSM) (see Table 4).

From Table 4, it appears that the LSA and BLSA techniques generally had larger eigenvalues than the Vector Space model. The sum of the three largest eigenvalues for LSA and BLSA are 100.31 and 106.57, respectively. These are much larger than that of the Vector Space Model (55.36). In addition, compared to LSA, the BLSA is 6.5% better. This confirms our conclusion drawn from prior section by correlation analysis, namely, BLSA returns more interrelated results than does the LSA. In other words, documents that share the same bibliographic information such as subject keywords will be more cohesively presented in the return list.

Table 4: Eigenvalues from the 50 by 50 return hit adjacency matrix for three different queries and their sum

	Eigen Value	Vector Space	LSA	BLSA
One-Word	First	8	42.34	41.22
Query	Second	0	5.58	5.53
	Third	0	5.54	2.44
Two-Word	First	8.8	7.42	6.24
Query	Second	3.64	3.17	3.15
	Third	3	2.48	2.22
Multi-Word	First	24.56	32.03	43.92
Query	Second	5.7	1.75	1.85
	Third	1.66	0	0
Sum		55.36	100.31	106.57

VI. Conclusions and Future Work

Recent research demonstrated that assistant tool supported interface is much more effective than a simple search interface like those typically found at commercial search engine sites [13]. The challenge, however, is that these assistant tools usually need additional trainings. In this paper we propose a new approach that "simulates" the advanced Boolean search by automatically adding the bibliographic matrix to the term matrix. One advantage for our approach is that it does not require the user to have knowledge about Boolean logic and special training to use. Further, our approach does not require multiple rounds of feedback from users. The key concept of our approach is to integrate the latent-semantic bibliographic information into the indexing scheme. The new scheme is built upon the Latent Semantic Analysis technology (LSA) and is referred to as Bibliographic Latent Semantic Analysis (BLSA). We chose to add bibliographic terms to LSA because it is the base of a series of algorithms such as pLSA and LDA that create reduced dimensions from the original term frequency matrix.

BLSA constructs a new augmentation semantic space by employing the bibliographic document links. A correlation analysis of the document matrix indicates that under appropriate conditions, the value of correlation increases for documents that are augmented with shared

bibliographic links. We also find that the lower the correlation between the two documents, the more likely that the augmentation will increase their correlation.

The primary limitation for this paper is that we only provide theoretical proofs for the merit of BLSA. Considering the small size of the test collection and limited number of testing queries, the case studies utilized to illustrate the implementation of BLSA in an IR application, and the significant difference between BLSA and LSA, we do not attempt to claim any statistically significant conclusions.

As the key benefit of the BLSA is to provide a more coherent return list rather than improving the precision, in some cases the traditional precision/recall evaluation approach is not the best way to assess the performance of BLSA. For example, suppose we have two documents D1 and D2 sharing the same author. D1 and D2 are ranked by a non-BLSA algorithm as r1 and r2 (r1 <r2) in terms of relevance to a query Q. After applying the BLSA algorithm, D1 and D2 will be re-ranked as r3 and r4 respectively. The re-ranking will make D1 and D2 to be closer on the ranking list, so we will get r1<r3<r4<r2. That means document D2 improves its ranking from r2 to r4 because it connects to a top ranking document D1. But document D1 decreases its ranking from r1 to r3 because it connects to a lower ranking document D2. The achievement of BLSA is that D1 and D2 get closer on the ranking list (which makes the query returns more coherent), not an improved Average Precision score.

Two preliminary case studies, the faculty publication dataset and the Rockwell KB system, demonstrate the feasibility of implementing BLSA in a real IR application. The study results also indicated in some cases the BLSA achieved better MAP scores than the LSA and VSM. In addition to the theoretical proof of improved correlations between "linked" documents, further evaluation metrics and experimental evidences are needed to validate the merits of the BLSA over LSA and VSM. For example, we can also define a new index that is able to effectively capture the BLSA's "clustering" feature and we can use it to evaluate the degree of coherence of the query returns.

In the future we also plan to conduct usability studies with real human tasks to evaluate the BLSA performance.

References

[1] Belkin, N., Kelly, D. K., Kim, J., Lee, H., Muresan, G., Tang, M., et al. (2003). Query length in interactive information retrieval. In *Proceedings of the 26th annual international ACM SIGIR Conference on Research and Development in Information Retrieval*.

[2] Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3, 993-1022.

[3] Borgman, C. (1996). Why are online catalogs still hard to use? *Journal of the American Society for Information Science*, 47 (7), 493-503.

[4] Cvetković, D. M.; Doob, M.; and Sachs, H. Spectra of Graphs: *Theory and Applications*, 3rd rev. enl. ed. New York: Wiley, 1998

[5] Deerwester, S., Dumais, S., Furnas, G., Landauer, T., & harshman, R. (1990). Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41 (6), 391-407.

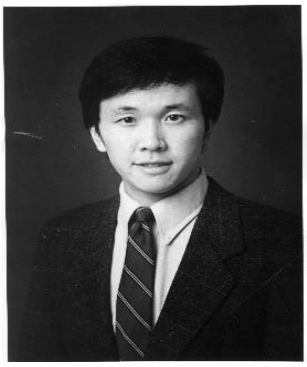
[6] Efron, M. (2005). Eigenvalue-based model selection during latent semantic indexing. *Journal of the American Society for Information Science*, 56 (9), 969-88.

[7] Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. In *Proceedings of the National Academy of Sciences*, 101 (1), 5228-35.

[8] Hofmann, T. (1999). Probabilistic Latent Semantic Indexing. In *Proceedings of the Twenty-Second Annual International SIGIR Conference on Research and Development in Information Retrieval*.

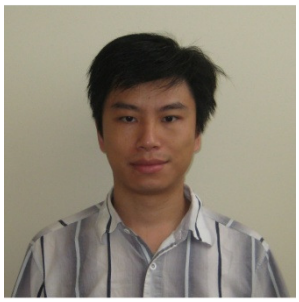
[9] Marchionini, G. (1995). *Information seeking in electronic environments*. Cambridge University Press.
[10] Neveol, A., Deserno, T. M., Darmoni, S.J., Guld, M.O., and Aronson, A.R. (2009). Natural language processing versus content-based image analysis for medical document retrieval. *Journal of the American Society for Information Science and Technology*, 60(1). 123-134.
[11] Newman, D., Asuncion,A., Smyth, P., & Welling, M. (2009). Distributed algorithms for topic models. *Journal of Machine Learning Research*, 10, 1801-1828.
[12] Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1), 1-47
[13] Topi, H. & Lucas, W. (2005). Searching the Web: operator assistance required. *Information Processing & Management*, 41(2), 383-403.
[14] Wolfram, D. (2007). Search characteristics in different types of Web-based IR environments: Are they the same? *Information Processing & Management*, 44 (3), 1279-1292.
[15] Yee, M. (2005). *Beyond the OPAC: future directions for web-based catalogues*. Australian Committee on Cataloguing. Perth, Western Australia.

Xiangming Mu received his Ph.D in information and library science from University of North Carolina at Chapel Hill in 2004. He is currently an assistant profession in the University of Wisconsin-Milwaukee. His research interests include biomedical and multimedia information retrieval and user interface designs.



Jun Zhang received his B.S. in computer engineering from Harbin Shipbuilding Engineering Institute, Harbin, China, in 1982 and was admitted to the graduate program of the Radio Electronic Department of Tsinghua University. After a brief stay at Tsinghua, he came to the U.S. for graduate study on a scholarship from the Li Foundation, Glen Cover, New York. He received his M.S. and Ph.D. both in electrical engineering from Rensselaer Polytechnic Institute in 1985 and 1988, respectively. He joined the faculty of the Department of Electrical Engineering and Computer Science, University of Wisconsin-Milwaukee, and currently is a professor. His research interests include image processing and signal processing. He has been an associate editor of IEEE Trans. Image Processing.

Wen Hu received the B.S. degree in computer science from the Harbin Engineering University, Heilongjiang, China in 1996 and the M.S. degree in Computer Science from the University of Wisconsin-Milwaukee in 2003. Currently, she is in the Ph.D. program in electrical engineering at the University of Wisconsin-Milwaukee. Her research interests are information retrieval, data mining and image processing



Tian Zhao is an associate professor at the Department of Computer Science of University of Wisconsin -- Milwaukee. His main research interests are in the areas of programming languages, type systems, and program analysis. He also studies the problems of software engineering, information retrieval, and geospatial information system.

Mark Fredette received his B.S. in computer engineering from the Milwaukee School of Engineering in 2005. He is currently working towards his M.S. in electrical engineering at the University of Wisconsin Milwaukee. Since 2003, he has been employed by Rockwell Automation where he currently holds the position of Senior Software Engineer. His research and professional interests include information retrieval, signal processing, artificial intelligence, belief propagation, and software engineering.

Guangwu Xu received his Ph.D. in mathematics from SUNY Buffalo. He is now with the Department of EE & CS, University of Wisconsin-Milwaukee. His research interests include cryptography and information security, compressed sensing, computational number theory, algorithms, and functional analysis.