Combining LSI and Vector Space to Improve Retrieval Performance

April Kontostathis
Department of Mathematics and Computer Science
Ursinus College
Collegeville, PA 19426
akontostathis@ursinus.edu

Abstract
We describe an approach to information retrieval using a technique which combines two existing well-known information retrieval techniques: traditional vector space retrieval and Latent Semantic Indexing (LSI). The technique described in this paper assigns a query to document similarity that is a weighted average of the similarity scores obtained by LSI and traditional vector space retrieval. Our approach improves retrieval performance by 12%, on average, over vector space retrieval on the seven collections studied. Four of the collections were chosen because their retrieval performance is higher when vector space retrieval is used instead of LSI, and our method improves retrieval performance by an average of 14% for these collections. Our work shows that LSI can be used to significantly improve retrieval performance on collections that previously did not benefit from LSI. We have also shown that a small, fixed LSI dimensionality reduction parameter (k=10) can be used to capture the ‘latent semantic’ information.

1 Introduction and Background
Many algorithms for searching textual collections have been developed. This paper focuses on two well-known systems: traditional vector space retrieval and Latent Semantic Indexing (LSI).

In traditional vector space retrieval, documents and queries are represented as vectors in \( t \)-dimensional space, where \( t \) is the number of indexed terms in the collection. Generally the document vectors are formed when the index for the collection is generated (these vectors form a matrix that is often referred to as the term by document or term-document matrix), and the query vectors are formed when a search of the collection is performed. In order to determine the relevance of a document to the query, the similarity between the query vector and each document vector is computed. Cosine similarity is often used to compare the vectors (van Rijsbergen, 1979). The cosine similarity metric provides an ordering of the documents, with higher scores assigned to documents that are considered more relevant to the query.

LSI has been applied to a wide variety of learning tasks, such as search and retrieval (Deerwester et al., 1990; Dumais, 1992), classification (Zelikovitz and Hirsh, 2001), and filtering (Dumais, 1992; Dumais, 1994). LSI is a dimensionality reduction approach for modeling documents.

LSI is based on a mathematical technique called Singular Value Decomposition (SVD). The SVD process decomposes a term by document matrix, \( A \), into three matrices: a term by dimension matrix, \( T \), a singular value matrix, \( S \), and a document by dimension matrix, \( D \). The number of dimensions is \( r \), the rank of \( A \). The original matrix can be obtained, through matrix multiplication of \( TSA^T \).

In an LSI system, the \( T \), \( S \) and \( D \) matrices are truncated to \( k \) dimensions. Dimensionality reduction is thought to reduce ‘noise’ in the term–doc
matrix, resulting in a richer word relationship structure that many researchers claim reveals latent semantics present in the collection (Deerwester et al., 1990; Dumais, 1992). Queries are represented in the reduced space by \( T_k^T q \), where \( T_k^T \) is the transpose of the term by dimension matrix, after truncation to \( k \) dimensions. Queries are compared to the reduced document vectors, scaled by the singular values \( (S_kD_k) \), by computing the cosine similarity. This process provides a similarity score for each document for a given query.

Choosing an optimal dimensionality reduction parameter \( k \) for each collection remains elusive. Traditionally, the optimal \( k \) has been chosen by running a set of queries with known relevant document sets for multiple values of \( k \). The \( k \) that results in the best retrieval performance is chosen as the optimal \( k \) for each collection. Optimal \( k \) values are typically in the range of 100-300 dimensions (Dumais, 1992;Letsche and Berry, 1997). Unfortunately, a \( k \) chosen using this technique is optimized to the queries in the training set.

Previous work has shown that LSI improves retrieval performance for some collections, but not every collection (Kontostathis et al., 2005; Jessup and Martin, 2001). Our technique combines LSI and traditional vector space retrieval, using a weighted average of the scores produced by these algorithms, to obtain the final document-query similarity score.

In Section 2 we describe our motivation and experimental design. In Section 3 we present our results, which show that our technique provides significant improvements in average precision over traditional vector space retrieval for each of the seven collections we studied. Furthermore, we show that a small, constant dimensionality reduction parameter, along with a constant value for the weighting parameter, can obtain consistently superior results across these same seven collections.

2 Approach

In this section we discuss the motivation for our approach, describe the evaluation plan and describe the methodology in detail.
few vectors, in combination with traditional vector retrieval, to get the best of both worlds?

2.2 Datasets and Evaluation.

Retrieval quality for an information retrieval system can be expressed in a variety of ways. In the current work, we primarily use precision and recall to express the quality of an information retrieval system. Precision is defined as the number of relevant documents returned divided by the total number of documents returned. Recall is the number of relevant documents returned divided by the total number of relevant documents.

In Section 3, we apply these metrics by computing precision at a given recall level. To calculate average precision we continue to retrieve documents for a given until a given percentage of correct documents has been retrieved (for example, 30%), and then compute the precision using the formula shown above. The average of the precision values for each query is computed to determine the average precision for a given collection.

Precision and recall require the existence of collections that contain a group of documents, a set of standard queries and a truth set for each query (a list of truly relevant documents). We used seven such collections during the course of our study. The collections we used are summarized in Table 1. These collections were downloaded from a variety of sources. MED, CISI, CRAN, NPL, and CACM were downloaded from the SMART web site at Cornell University. LISA was obtained from the Information Retrieval Group web site at the University of Glasgow. The OHSUMED collection was downloaded from the Text Retrieval Conference (TREC) web site at the National Institute of Standards and Technology. Not all of the documents in the OHSUMED collection have been judged for relevance for each query. In our experiments, we calculated precision and recall by assuming that all unjudged documents are not relevant.

Table 1 also shows the optimal \( k \) for LSI processing for each collection. The optimal \( k \) was identified by measuring the average precision for each collection as we varied \( k \) from 5 to 200 (in increments of 5) for the smaller collections (MED, CISI, CRAN, CACM), and from 25 to 500 (in increments of 25) for the larger collections (NPL, LISA, OHSUMED).

Term weighting schemes are commonly applied to the entries in both the query and the document vectors. In our studies we employed a log entropy weighting scheme. This approach defines a global term weight for each indexed term, \( i \), using the formula:

\[
g_i = 1 + (\sum_j p_{ij} \log_2(p_{ij})) / \log_2 n
\]

Where \( n \) is the number of documents in the collection, the summation traverses all documents in the collection, and \( p_{ij} \) is the probability of the term \( i \) appearing in the document \( j \) and is defined as:

\[
p_{ij} = f_{ij} / (\sum_j f_{ij})
\]

Where \( f_{ij} \) is the number of times term \( i \) appears in document \( j \). The local weight of a term in a document or query is then defined as:

\[
l_i = \log_2(1 + f_{ij})g_i
\]

The purpose of the log entropy weighting scheme is to reduce the relative importance of high frequency terms while giving words that distinguish the documents in a collection a higher weight. In our experiments we also normalized each query and document vector for all collections except OHSUMED. A significant portion of the OHSUMED collection contains title data only (as opposed to titles and abstracts), and normalization results in these documents receiving higher weight when they contain a term in common with the query. This problem was previously noted in (Hersh et al., 1994).
### Table 1: Summary of Collections Tested

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Description</th>
<th>Docs</th>
<th>Terms</th>
<th>Queries</th>
<th>Optimal LSI k value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MED</td>
<td>Medical abstracts</td>
<td>1033</td>
<td>5831</td>
<td>30</td>
<td>75</td>
</tr>
<tr>
<td>CISI</td>
<td>Information science abstracts</td>
<td>1450</td>
<td>5143</td>
<td>76</td>
<td>145</td>
</tr>
<tr>
<td>CACM</td>
<td>Communications of the ACM abstracts</td>
<td>3204</td>
<td>4863</td>
<td>52</td>
<td>200</td>
</tr>
<tr>
<td>CRAN</td>
<td>Cranfield collection</td>
<td>1400</td>
<td>3932</td>
<td>225</td>
<td>185</td>
</tr>
<tr>
<td>LISA</td>
<td>Library and Information Science Abstracts</td>
<td>6004</td>
<td>18429</td>
<td>35</td>
<td>500</td>
</tr>
<tr>
<td>NPL</td>
<td>Larger collection of very short documents</td>
<td>11429</td>
<td>6988</td>
<td>93</td>
<td>500</td>
</tr>
<tr>
<td>OHSUMED</td>
<td>Clinically-oriented MEDLINE subset</td>
<td>348566</td>
<td>170347</td>
<td>106</td>
<td>500</td>
</tr>
</tbody>
</table>

#### 2.3 Experimental Design.

The processing tasks are outlined below:

- The term-document matrix is created for each collection. The terms were words, special characters were removed, a standard English stop list was applied, and stemming was not used. A log entropy weighting scheme was applied.

- The SVD is performed on the term-document matrix. We used the Parallel General Text Processor (PGTP) to perform the SVD (Berry and Martin, 2004).

- Each query in the set of standard queries was run first using traditional vector space retrieval and again using LSI with a fixed value for dimensionality reduction, $k'$. The result of this process was two vectors, $a$ and $b$, respectively. These vectors have length $n$ where $n$ is the number of documents in the collection. The value in position $d$ of each vector measured the similarity of the query to document $d$.

- A single vector, $r$, was created by combining $a$ and $b$. A weighting factor, $w$, was used to produce a weighted average. In other words, $r = (1 - w)a + wb$. Values in this vector were sorted in descending order to determine the ranking of documents returned for each query.

- The average precision of the system was calculated as described in Section 2.2.

- The average precision was compared to the average precision obtained using only traditional vector space retrieval, and to the average precision obtained using only LSI at the optimal $k$ value for each collection.

This process was repeated using a variety of values for the two parameters: the dimensionality reduction parameter, $k'$, and the weighting parameter, $w$, which was used to compute the final similarity score.

#### 3 Experimental Results

The results of our experiments when $k' = 10$ and the $w = .2$ appear in Table 2. These values provide consistently good results across all collections. On average, the combining method improved retrieval performance an average of 12% over traditional vector space retrieval. All collections showed significant improvements, ranging from 8% to 19%. We note that LSI contributes only 20% of the weight to the final similarity score for the results shown in Table 2.

Four of the collections were included in our studies because it has been noted previously that traditional vector space retrieval is preferred to LSI for these collections (CACM, LISA, NPL, OHSUMED) (Kontostathis et al., 2005). The proposed method provides an average improvement over traditional vector space retrieval of 14% for these four collections, when $k' = 10$ and $w = .2$. The improvement over LSI is much higher (see Table 2).

Although $k' = 10$ and $w = .2$ provide good performance across all the collections, it is possible to tune the system for each collection. For example, $k' = 35$ and $w = .5$ provides an average precision of .72 for
Table 2: Experimental Results at $k' = 10$, with LSI representing 20% of the final score

<table>
<thead>
<tr>
<th>Collection</th>
<th>LSI Avg. Prec. (at Optimal $k$)</th>
<th>Vector Avg. Prec.</th>
<th>Experimental Avg. Prec.</th>
<th>% Impr. over LSI</th>
<th>% Impr. over Vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>MED</td>
<td>.722</td>
<td>.540</td>
<td>.583</td>
<td>-19%</td>
<td>8%</td>
</tr>
<tr>
<td>CISI</td>
<td>.222</td>
<td>.208</td>
<td>.230</td>
<td>4%</td>
<td>11%</td>
</tr>
<tr>
<td>CACM</td>
<td>.182</td>
<td>.206</td>
<td>.229</td>
<td>26%</td>
<td>11%</td>
</tr>
<tr>
<td>CRAN</td>
<td>.450</td>
<td>.398</td>
<td>.436</td>
<td>-3%</td>
<td>10%</td>
</tr>
<tr>
<td>LISA</td>
<td>.277</td>
<td>.302</td>
<td>.344</td>
<td>24%</td>
<td>14%</td>
</tr>
<tr>
<td>NPL</td>
<td>.142</td>
<td>.158</td>
<td>.188</td>
<td>32%</td>
<td>19%</td>
</tr>
<tr>
<td>OSHUMED</td>
<td>.022</td>
<td>.150</td>
<td>.170</td>
<td>673%</td>
<td>13%</td>
</tr>
</tbody>
</table>

MED, making this technique perform as well as LSI at the optimal setting ($k = 75$), with fewer dimensions. The optimal settings for each collection are shown in Table 3. To obtain this data, we varied the $w$ from .1 to .5 (in increments of .1), and continued to increase the value of $k'$ in increments of 5, up to $k' = 100$ for the larger collections and $k' = 200$ for the smaller collections. The optimal $k'$ for LSI was also tested if it fell outside these parameters (i.e. $k' = 500$ was tested for LISA, NPL and OHSUMED). In all cases the average precision at the optimal $k$ for LSI was worse than the average precision at the optimal $k'$ for the experimental technique (shown in Table 3) for all weighting factors .1 thru .5. Furthermore, the optimal $k'$ for the experimental technique was always smaller than the optimal $k$ for LSI.

3.1 Discussion

The primary advantage of the combining method is that there is little variability as the parameters change. Figures 4 and 5 give examples of how the average precision changes when dimensionality reduction parameter and the weighting factor are varied independently. In Figure 4, where $w$ has the constant value .2, the average precision varies by only 0.0032 as $k'$ varies from 5 to 300. We see a bit more fluctuation in Figure 5 where we have set $k'$ to the constant value 10 and vary $w$ between .1 and .5. The average precision varies by .0195 over the range of values; however, varying from .1 to .3 results in only a .0046 variation in average precision (from max to min).

A comparison of average precision in Tables 2 and 3 show that use of the optimal settings does not greatly affect the average precision on six of the seven collections. CRAN, CISI, LISA, NPL, and OHSUMED derive a small benefit from tuning the parameters, ranging from .003 and .017. CACM obtains no improvement from tuning (when precision is reported to 3 significant digits). The MED collection is the exception, it greatly benefits from tuning as the average precision moves from .583 to .772, but MED is an unusual collection that was derived by combining the results from a series of keyword searches, and is known to greatly benefit from the use of dimensionality reduction because the truncated SVD can easily re-segment the document space (Deerwester et al., 1990).

One common problem with information retrieval systems in general, and LSI in particular, is that optimal parameter settings vary widely when different corpora are tested. Figures 1-3 show the great degree of fluctuation in the performance of the LSI system that results when the parameter $k$ is changed. Another problem is that the use of gold standard queries...
Table 3: Optimal parameter settings and resultant average precision

<table>
<thead>
<tr>
<th>Collection</th>
<th>$k'$</th>
<th>Weighting Factor</th>
<th>Average Precision</th>
<th>% Impr. over LSI</th>
<th>% Impr. over Vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>MED</td>
<td>35</td>
<td>.5</td>
<td>.722</td>
<td>0%</td>
<td>34%</td>
</tr>
<tr>
<td>CISI</td>
<td>75</td>
<td>.2</td>
<td>.243</td>
<td>9%</td>
<td>17%</td>
</tr>
<tr>
<td>CACM</td>
<td>10</td>
<td>.2</td>
<td>.229</td>
<td>26%</td>
<td>11%</td>
</tr>
<tr>
<td>CRAN</td>
<td>125</td>
<td>.5</td>
<td>.453</td>
<td>1%</td>
<td>14%</td>
</tr>
<tr>
<td>LISA</td>
<td>80</td>
<td>.3</td>
<td>.360</td>
<td>30%</td>
<td>19%</td>
</tr>
<tr>
<td>NPL</td>
<td>15</td>
<td>.2</td>
<td>.191</td>
<td>35%</td>
<td>21%</td>
</tr>
<tr>
<td>OSHUMED</td>
<td>95</td>
<td>.3</td>
<td>.174</td>
<td>69%</td>
<td>16%</td>
</tr>
</tbody>
</table>

Figure 5: The affect of varying weighting factor for OHSUMED, when $k'$ remains stable

and truth sets invariably results in systems which are tuned to a particular set of queries. We contend that, for general search and retrieval applications, a hope of capturing a small improvement in average precision is not worth the disadvantage of tuning the system to a particular set of queries. Therefore, we are comfortable choosing parameters that work well for all systems. The choice of $k' = 10$ and $w = .2$ will have a significant positive impact on the average precision for systems using traditional vector space retrieval.

### 3.2 Cost

The computational complexity of SVD approximation algorithms is dependent on the number of factors required ($k$), the sparseness of the original term-document matrix, and the number of documents in the corpus (Bingham and Mannila, 2001). Many approximation algorithms for computing only the $k$ largest singular values and their associated vector exist (Berry, 1992; Berry et al., 1993). Furthermore, the data in Figure 5 of (Yang, 1995) indicates that the complexity of the Lanczos algorithms is close to linear when $k < r/2$ (for the three collections tested). Thus computation of only 10 singular values and their association vectors will have significantly reduced cost when compared to the usual 100-300 dimensions required for LSI.

Furthermore, the combined technique will require minimal extra memory during query run time, when compared to traditional vector space retrieval. The term and document vectors must be stored in memory, but this requires only an additional 79MB of RAM for OHSUMED when $k' = 10$ (350,000 x 10 x 16 = 53MB for the document vectors and 170,000 x 10 x 16 = 26 MB for the term vectors, assuming double precision values are stored). The additional memory requirements for the other collections are much smaller, ranging from .8 MB for CRAN to 3.7 MB for LISA. Even at the optimal $k'$ of 95, OHSUMED would need only an additional 754 MB of RAM.

We chose $k' = 10$ because it was the smallest $k'$ which was optimal for any collection. A closer look at the data reveals that setting $k' = 5$ results in very little degradation in precision. When rounded to three decimals, OHSUMED show no difference in precision as $k'$ increases from 5 to 10. Average precision for LISA is actually slightly higher at $k' = 5$ (.188). NPL, CRAN, CISI, and CACM showed minor differences ranging from .001 to .015. The largest difference was found when the MED collection was tested. Average precision at $k' = 5$ is .559 and this increases to .583 when $k' = 10$; however, even at $k' = 5$, MED shows a 3.5% improvement over traditional vector space retrieval. Thus, if computational resources are severely limited, it is possi-
ble to set $k'$ to a value smaller than 10 and still obtain results superior to traditional vector space retrieval.

Increasing $k'$ to 15 results in no improvement for OHSUMED (rounding to 3 decimals), a slight degradation in performance for CISI and CACM (-.004 and -.005, resp.), slight improvements for CRAN (.003), LISA (.007), and NPL (.003), and a larger improvement for MED (.027).

4 Related Work

The idea of combining information retrieval techniques to improve retrieval performance is not a new one. Many techniques have been proposed. Croft provides an excellent summarization of approaches in (Croft, 2000) where he also provides a framework for understanding when and why combination approaches work.

In our review of the literature we find that only the optimal $k$ value has been used when LSI has been combined with other retrieval systems (Diamond and Liddy, 1998; Kenmogne, 2005; Dumais, 1996; Lochbaum and Streeter, 1989). Our studies are the first to indicate that the latent semantic information is capturing in a small number of vectors, and to leverage that knowledge in a way that reduces LSI overhead while improving retrieval performance.

5 Conclusions

We have presented an approach to information retrieval using a weighted average of the search results from traditional vector space retrieval and LSI. We have also identified parameter settings which work across collections. Our system significantly improves retrieval performance over traditional vector space retrieval.

Our work demonstrates that LSI can be used, very effectively, in combination with other retrieval systems. We have also shown that very few dimensions are required to capture the ‘latent semantic’ information provided by LSI. Thus, we provide both practical and theoretical contributions.

From a practical perspective, our approach provides substantial benefit without regard to the previous retrieval effectiveness of LSI on a given corpus. If a collection is known to have improved retrieval performance using LSI, we can meet or exceed LSI performance with many fewer dimensions. This will reduce preprocessing overhead (SVD costs) and query runtime costs (both CPU time and memory) over the optimal LSI system used alone. If a collection performs better with traditional vector space retrieval, our approach provides a clear retrieval performance advantage, with a minimal amount of additional processing and memory.

From a theoretical perspective, we have confirmed that LSI captures latent semantic information with only a few dimensions for seven collections that range in size from very small (1033 docs) to moderately large (over 300,000 docs).

6 Future Work

In the future we would like to apply this technique to other applications that have successfully used LSI or SVD, such as collaborative filtering (Hersh and Zhukov, 2004) or text annotation (Martin and Foltz, 2004).

Future work that involves an analysis of the values produced by LSI, like the studies described in (Kontostathis and Pottenger, 2006) and (Kenmogne, 2005), can focus on fewer singular values and their associated vectors.

Acknowledgements

We would like to thank Professor Brian D. Davison at Lehigh University for his invaluable comments on an early draft. This work was partially supported by the National Center for Supercomputing Applications under IRIO050005 and utilized the copper system.

References


Ella Bingham and Heikki Mannila. 2001. Random projection in dimensionality reduction: applications to
image and text data. In Knowledge Discovery and Data Mining, pages 245–250.


