Incorporating Ad Hoc Phrases in LSI Queries

Roger Bradford
Agilex Technologies Inc, 5155 Parkstone Drive, Chantilly, VA, U.S.A.

Keywords: Latent Semantic Indexing, LSI, Phrase-based Retrieval, Phrase Indexing.

Abstract: Latent semantic indexing (LSI) is a well-established technique for information retrieval and data mining. The technique has been incorporated into a wide variety of practical applications. In these applications, LSI provides a number of valuable capabilities for information search, categorization, clustering, and discovery. However, there are some limitations that are encountered in using the technique. One such limitation is that the classical implementation of LSI does not provide a flexible mechanism for dealing with phrases. In both information retrieval and data mining applications, phrases can have significant value in specifying user information needs. In the classical implementation of LSI, the only way that a phrase can be used in a query is if that phrase has been identified a priori and treated as a unit during the process of creating the LSI index. This requirement has greatly hindered the use of phrases in LSI applications. This paper presents a method for dealing with phrases in LSI-based information systems on an ad hoc basis – at query time, without requiring any prior knowledge of the phrases of interest. The approach is fast enough to be used during real-time query execution. This new capability can enhance use of LSI in both information retrieval and knowledge discovery applications.

1 INTRODUCTION

In 1988, researchers from Bellcore introduced the technique of latent semantic indexing (LSI) for text retrieval (Furnas et al, 1988). The technique relies on the notion of distributional semantics – specifically, that the meaning of a term in text is directly correlated with the contexts in which it appears. LSI accepts as input a collection of documents and produces as output a high-dimensional vector space. All of the documents in the collection are represented by vectors in this vector space. Similarly, all of the terms that comprise those documents are represented by vectors in this vector space (except for very frequently occurring terms that typically are treated as stopwords).

LSI employs the technique of singular value decomposition (SVD) to carry out a large-scale dimensionality reduction (as described in the following section). This dimensionality reduction has two key effects:

- Terms that are semantically related are assigned representation vectors that lie close together in the LSI vector space.
- Documents that have similar conceptual content are assigned representation vectors that lie close together in the space.

These characteristics of LSI spaces form the basis for a wide range of applications, ranging from automated essay scoring to literature-based discovery (Dumais, 2004). Typically, in these applications, little or no use is made of phrases. This is because, in the classical formulation of LSI, in order for a phrase to be used as such in a query, that phrase must be identified and treated as a unit in the initial stage of creating the LSI representation space. However, experience has shown that it is quite difficult to predetermine collections of phrases that will enhance performance in applications. This paper presents a technique for using phrases in LSI queries on an ad hoc basis – at query time, without requiring any changes in existing LSI spaces.

Although LSI primarily has been used with text, it is a completely general technique and can be applied to any collection of items composed of features. LSI has, for example, been used with great success in categorizing, clustering, and retrieving audio, image, and video data. Although this paper focuses on textual phrases, the approach described

Bradford R..
Incorporating Ad Hoc Phrases in LSI Queries.
DOI: 10.5220/0005073300610070
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval* (KDIR-2014), pages 61-70
ISBN: 978-989-758-048-2

here is equally applicable to non-textual linked features in other types of media.

2 LSI PROCESSING

The LSI technique applied to a collection of documents consists of the following primary steps (Dumais et al 1988):

1. A matrix A is formed, wherein each row corresponds to a term that appears in the documents, and each column corresponds to a document. Each element $a_{m,n}$ in the matrix corresponds to the number of times that the term *m* occurs in document *n*.
2. Local and global term weighting is applied to the entries in the term-document matrix. This weighting may be applied in order to achieve multiple objectives, including compensating for differing lengths of documents and improving the ability to distinguish among documents. Some very common words such as *and*, *the*, etc. typically are deleted entirely (i.e., treated as stopwords).
3. Singular value decomposition (SVD) is used to reduce this matrix to a product of three matrices:

$$A = U \Sigma V^T \quad (1)$$

   Let A be composed of t rows corresponding to terms and d columns corresponding to documents. U is then a t*t orthogonal matrix having the left singular vectors of A as columns. V is a d*d orthogonal matrix having the right singular vectors of A as columns. Σ is a t*d diagonal matrix whose elements are the singular values of A (the non-negative square roots of the eigenvalues of AA^T).
4. Dimensionality is reduced by deleting all but the *k* largest values of Σ, together with the corresponding columns in U and V, yielding an approximation of A:

$$A_k = U_k \Sigma_k V_k^T \quad (2)$$

   which is the best rank-k approximation to A in a least-squares sense.
5. This truncation process provides the basis for generating a *k*-dimensional vector space. Both terms and documents are represented by *k*-dimensional vectors in this vector space.
6. New documents (e.g., queries) and new terms are represented in the space by a process known as folding-in (Furnas et al, 1988). To add a new document, for example, that document is first subjected to the same pre-processing steps (e.g., stopword removal) as those applied to the original documents used in creating the space. The document then is assigned a representation vector that is the weighted average of the representation vectors for the terms of which it is composed. A similar process is employed to fold in new terms (see section 5.2 for more detail).
7. The similarity of any two objects represented in the space is reflected by the proximity of their representation vectors, generally using a cosine measure. Results of queries are sorted by cosine: the higher the cosine, the more similar the returned object (term or document) is to the query.

Extensive experimentation has shown that proximity of objects in such a space is an effective surrogate for conceptual similarity in many applications (Bradford, 2009).

3 PHRASES

People directly perceive phrases as having utility in representing the information content of documents. For example, Figure 1 shows the distribution of the number of tokens constituting indexing elements chosen by professional indexers for a set of technical journal articles in INSPEC (Hulth, 2004). Less than 14% of the indexing elements chosen consist of single tokens.

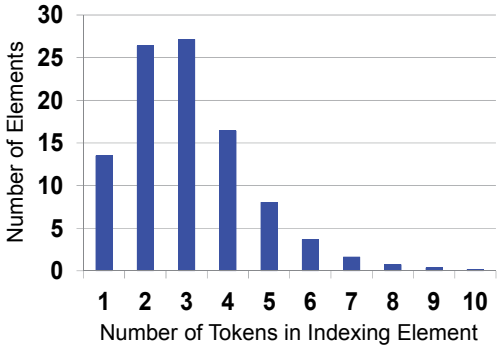

Figure 1: Human choice of phrases as indexing elements.

Phrase search is widely used in information retrieval. In (Manning, Raghavan, and Schütze, 2008) it is reported that as many as 10% of web queries are explicit phrase queries (i.e., entered within quotes) and many more are implicit phrase queries entered without double quotes. Use of phrases in queries is a standard capability in most contemporary text retrieval systems.

With the classical formulation of LSI, if, for example, *life cycle analysis* is used as a query, the system will retrieve the LSI representation vectors for each of these individual words and then form a query vector that is the weighted average of these three vectors. Since the words *life*, *cycle*, and *analysis* are used in many contexts other than that indicated by the phrase *life cycle analysis*, the query vector thus formed will constitute a poor approximation of an appropriate vector for representing the specific concept of *life cycle analysis.*

The problem is particularly acute for names of persons. For example, in many document collections, an LSI search for *George Cheney* (the internationally known speaker and writer on organizational communication) would yield very poor results, because the many references to *George Bush* and *Dick Cheney* would create representation vectors for *George* and *Cheney* that were closely associated with politics as opposed to the educational context of *George Cheney*. Although entity extraction can be, and often is, used in creating LSI spaces, even the best contemporary entity extractors miss or incorrectly render on the order of 20% of the names in typical real-world document collections. This can be a major limitation in discovery applications where names of people provide key context.

It has long been recognized that treating phrases as units could have beneficial effects in LSI applications. In one of the earliest papers on LSI, the inventors of the technique noted the potential value of use of phrases:
"We think we can greatly improve performance by ***incorporating short phrases****..."* (Dumais et al, 1988) (emphasis added).

4 RELATED WORK

There is a long history of investigation of the utility of phrases in information retrieval. Global approaches (i.e., broadly treating phrases in a document collection as indexing elements) have yielded generally disappointing results.

As early as 1975, Salton et al showed that use of statistically-derived phrases could improve retrieval performance for SMART, a then-current vector-space-based text retrieval system (Salton, Yang, and Yu, 1975). The authors obtained improvements in average precision of 17-39%, but were dealing with very small text collections (each less than 500 documents). In 1989, Fagan carried out similar experiments, obtaining mixed results, varying from -11% to +20%, with somewhat larger test collections (up to 12 thousand documents) (Fagan, 1989). In 1997, Mitra conducted similar tests with a collection of 211 thousand documents, obtaining precision improvements of 1% (Mitra et al, 1997).

There are two primary factors responsible for the observed decline over time of phrase impact in these studies – the larger sizes of the test collections employed and the fact that the comparison baseline (retrieval performance using individual terms) improved significantly over that 20-year time frame. In discussing results from the first five TREC conferences, Mitra et al noted that: "*In the past five years of TREC, overall retrieval effectiveness for SMART has more than doubled, but the added effectiveness due to statistical phrases has gone down from 7% to less than* 1%" (Mitra et al, 1997).

In 1999, Turpin and Moffat provided additional evidence that, as performance of vector-space-based information retrieval systems employing individual terms improved, the utility of adding phrases diminished (Turpin and Moffat, 1999).

Summarizing the first eight years of TREC testing, Harman noted: "*the investigation of the use of phrases in addition to single terms ... has long been a topic for research in the information retrieval community, with generally unsuccessful results. ... almost all* (TREC) *groups have experimented with phrases. In general, these experiments have been equally unsuccessful"* (Harmon, 2005).

In 2006, Metzler, Strohman, and Croft reported results of their work over three years on the TREC Terabyte tracks. During that period they performed a variety of tests using statistically-derived phrases. None of the techniques tried yielded improvement over a bag-of-words baseline (Metzler, Strohman, and Croft, 2006).

Multiple approaches have been tried for automatically identifying phrases of interest within a document collection. In general, these produce far more candidate phrases than is desirable. A number of approaches have been tried for pruning long lists of candidate phrases. Somewhat surprisingly, applying filtering criteria such as mutual information and entropy generally has not resulted in lists of phrases that yield significant retrieval enhancement.

There has been some success with selecting specific types of phrases. Ogawa et al employed two-term noun phrases in tests using TREC-8 queries applied to the WT2g document collection (247 thousand documents) and TREC-9 queries applied to the WT10g document collection (1.7 million documents). They achieved improvements

in average precision of 3 to 12 percent using phrases derived from the TREC queries. For phrases derived from query expansion, they achieved improvements for the smaller test set, but not for the larger one (Ogawa et al, 2000). Zhai et al achieved similar results for noun phrases with TREC-5 queries (Zhai et al, 1996). Kraiij and Pohlmann found modest improvements using proper names as indexing units (Kraiij and Pohlmann, 1998). Jiang et al had good success with classifier-thing bigrams (word pairs in which the first word effectively selects for a subclass of the type referred to by the second word) (Jiang et al, 2004).

Some techniques for creating lists of phrases derive the candidates from existing data collections. In particular, WordNet and Wikipedia have been popular sources of candidate phrases. However, these approaches typically have not shown a significant advantage over other techniques.

Counterintuitively, human selection of phrases also generally has not produced significant improvements in retrieval performance. In 2010, Broschart et al conducted experiments using the TREC GOV2 collection (25 million documents), three sets of TREC queries, and five users. For all five users, incorporation of their chosen phrases as indexing units yielded overall *lower* average precision than the baseline comparison (BM25F). They found that users frequently disagreed on phrases in the TREC queries. On average, two users highlighted the same phrase only 47% of the time (Broschart, Berberich, and Schenkel, 2010). Such inconsistencies also were noted by Kim and Chan. They had ten human subjects read articles averaging 1300 words in length and choose the ten most meaningful phrases from each one. On average, only 1.3 phrases matched those chosen by others (Kim and Chan, 2004).

There has been little reported testing of phrase pre-processing for LSI. In 2000, Lizza and Sartoretto incorporated two-word phrases into LSI indexes for small document collections. They reported a 9% improvement in average precision for the TIME collection (423 documents), but no useful increase for the MEDLINE (1033 documents) and CISI (1460 documents) collections (Lizza and Sartoretto, 2000).

Weimer-Hastings investigated the effect of incorporating noun and verb phrases when using LSI to grade student answers to questions. He employed manually identified phrases in comparing student answers with expected answers. He found a decrease in correlation with human evaluations when the phrases were used (Weimer-Hastings, 2000).

Wu and Gunopulos tested LSI with two-word phrases, selected based on a threshold for either document frequency or information gain. In their testing, they employed the R118 subset of the Reuters 21578 test set. They only were able to increase F1 for that test set from .8417 to .8449, which was not statistically significant (Wu and Gunopulos, 2002).

Nakov, Valchanova, and Angelova examined the impact of different text pre-processing steps (lemmatization, stopword removal, etc) in using LSI for document categorization. They employed a test set of 702 Bulgarian-language documents assigned to 15 categories. Altogether they tested 120 combinations of local weight, global weight, and LSI dimensions. In 87 of these cases, incorporating phrases into the LSI index yielded no change in microaveraged categorization accuracy for the collection. In 27 cases there was an improvement; for 9 cases the results were worse. Most improvements were 1% or less. In all cases where the best-performing local and global weights were used, phrase processing had no effect (Nakov, Valchanova, and Angelova, 2003).

Grönqvist examined LSI performance on a synonym test of 440 Swedish queries, with 10.9% of the queries and 35.5% of the answers consisting of phrases. Results from creating the LSI index with and without phrases are shown in Table 1. He noted that the difference between the first case and the second was not statistically significant, but that the difference between the first and the third was (Grönqvist, 2005A). Overall, incorporation of two- and three-word phrases (Grönqvist, 2005B):

- Corrected 30 queries that had been incorrectly evaluated by classical LSI.
- Incorrectly judged 15 queries that had been correctly evaluated by classical LSI

Table 1: LSI phrase indexing results.

Indexed terms	% Correct
Individual words	59.55%
Individual words + two-word phrases	61.59%
Individual words + two-word phrases + three-word phrases	62.73%

In his thesis, Grönqvist also performed document retrieval tests with 101 CLEF topics for Swedish and 133 TREC topics for English. For Swedish, there was a small decrease in performance when incorporating phrases. For English, the results were

essentially unchanged from the individual word case (Grönqvist, 2006).

Olney used LSI for determining the semantic equivalence of sentence pairs. He employed two test sets - the Microsoft Research Paraphrase Corpus and a corpus based on the novel 20,000 Leagues under the Sea. The training set consisted of the TASA corpus (~70 MB of text) combined with the Wall Street Journal section of the Penn Treebank (~50,000 words). In creating the term-document matrix he included all word bigrams that occurred more than once. This increased the number of rows in the matrix by an order of magnitude (from 75,640 to 756,741). He found a negligible difference in task performance for both test sets. A variant in which word trigrams were used as context, rather than documents, also yielded no significant difference Olney, 2009).

Collectively, these studies suggest that pre-selection of phrases for LSI indexing has limited utility, at least using current phrase selection techniques. This result is consistent with the previously-cited studies of use of phrases with other vector-space-based information retrieval techniques.

One consistent result in all of the above studies is that, although *overall* task performance was not notably enhanced through use of phrases, the performance for *certain queries* was significantly enhanced when relevant phrases were employed. In light of the difficulties in pre-selecting phrases, this result, together with the user predilection to employ phrases noted in section 3, provides strong motivation for development of an *ad hoc* phrase query capability for vector space retrieval. The following discussion presents a novel method of providing such a capability in LSI spaces.

5 AD HOC PHRASE PROCESSING METHOD

The fundamental characteristics of an LSI space enable a query-time phrase processing capability for LSI. The key relevant characteristic of the technique is the duality condition described below.

5.1 Duality Condition

In any LSI space, the following duality condition holds:

- The LSI representation vector for any *document* corresponds to the weighted average of the LSI vectors for the *terms* contained in that document.
- The LSI representation vector for any *term* corresponds to the weighted average of the LSI vectors for the *documents* that contain that term.

5.2 Phrase Vector Creation Method

For a given LSI space, the proposed procedure for generating a representation vector for an arbitrary given phrase is as follows:

1. Identify the documents from the indexed collection that contain the given phrase.
2. Retrieve the LSI vectors corresponding to those documents.
3. Calculate the weighted sum of those representation vectors. (Using the notation of section 2, let P be the 1*d vector whose components indicate frequency of occurrence of the phrase in the documents of the collection, weighted by the same global and local weighting algorithms as used in creating the original LSI index. The new phrase vector is then given by $PV_k\Sigma_k^{-1}$).

The resulting vector will be a good approximation of the vector that *would have been created* if that phrase had been treated as a unit during the LSI pre-processing stage. It is not *exactly* the same vector, because of the complex balance that is created between term and document vectors when the SVD algorithm is applied to the collection as a whole during the creation of the LSI space. Changing what are considered to be terms within a set of documents has transitive impacts that affect the entire LSI space created from those documents. These changes will create differences in retrieval results. However, as shown in the following examples, the approximation vector produced using this method is quite accurate, at least for an LSI space of reasonable size.

In order for this technique to be applied at query time, the process of creating the approximation vector for a phrase must be quite fast (on the order of seconds). Thus all three steps described above must be carried out rapidly:

- For a given phrase, it is necessary to rapidly determine which documents contain that phrase. This necessitates use of an inverted index of term occurrences. This easily can be implemented using a variety of open source or commercial software packages. With modern hardware and software implementations, providing response times on the order of seconds for this function is straightforward,

even for large document collections. Moreover, modern LSI applications frequently incorporate a Boolean retrieval capability for added flexibility. In such cases, the needed inverted index will already be available. For example, the LSI engine employed in the testing described here comes bundled with the DT Search commercial Boolean text retrieval system. Many extant LSI applications employ the Lucene open source text retrieval engine for Boolean retrieval. Such software can retrieve postings lists for terms (and combine them into occurrence lists for phrases) with sub-second response, even for large document collections.

- Once the set of documents containing the phrase has been identified, retrieving the corresponding LSI vectors (and their term weights) is a straightforward database lookup process, using the same database used for other LSI retrieval functions.
- Combining the vectors can be carried out rapidly in local memory.

In contemporary hardware environments it is quite feasible for this series of operations to be carried out in near-real time (seconds) while processing a user query.

5.3 Examples

The most straightforward method of demonstrating the efficacy of an ad hoc technique such as this is to directly compare the results of example LSI queries involving phrases for three cases:

1. Where there is no attempt at specific phrase processing (classical LSI processing, where the terms of the phrase are treated independently).
2. Where the phrase is treated as a unit during pre-processing of the text of the document collection (so that the phrase is treated as a single feature in creating the term-document matrix).
3. Where the phrase is processed at query time using the procedure described above.

The following examples were generated using LSI spaces constructed from a collection of 1.6 million news articles from the time frame 2012-2013. This document set was a convenient size for experimentation, although at the lower end of collection size for contemporary enterprise LSI applications, most of which tend to range from millions to tens of millions of documents (Bradford, 2011). The LSI indexes were created using the following processing parameters: logarithmic weighting locally, entropy weighting globally, 300 dimensions, and pruning of terms that only occurred once. (These are typical parameters for contemporary LSI applications).

A decision was made to carry out entity extraction for PERSON, LOCATION, and ORGANIZATION entities as a pre-processing step for the baseline LSI case. The ad hoc query technique described here is eminently suitable for dealing with named entities. If these entities had not been included in the baseline case, quite striking examples could have been presented here for ad hoc retrieval of person, organization, and location names. However, it has become routine practice in recent years for entity extraction to be applied as a pre-processing step in creating LSI indexes for applications. (Named entities are one exception to the problem of pre-determining multiword units. Even with the error rates of contemporary entity extractors, incorporating entity extraction into LSI pre-processing has been shown to have significant beneficial effect on LSI performance in most applications). Since it is so widely used, it was felt that the most meaningful baseline for comparisons for this paper would be an LSI index that included entity pre-processing.

The entity extraction was carried out using a commercial entity extraction product (Rosoka version 3.0). The LSI indexing was carried out using a commercial LSI engine (Content Analyst version 3.10). Index creation time was approximately 45 minutes employing 16 m1.xlarge Amazon AWS EC2 instances for creation of the term-document matrix and one cr1.8xlarge instance for the SVD calculation.

In each of the following examples, three sets of results are presented for each phrase query:

- The first set of results was generated using a query against a classical LSI index, with no specific phrase processing.
- The second set of results was produced by taking the given phrase and building a new LSI space, identical to that of the previous case, with the single exception of treating that one phrase as a unit during the LSI processing.
- The third set of results was obtained by employing the ad hoc phrase retrieval technique described in this paper.

In each of the following three examples, the tables show the ten terms in the LSI space that were ranked closest to the query vector corresponding to the given phrase (i.e., those terms having the highest cosine value between their representation vectors and that of the query). For the news articles used in the testing, there are many similar terms. Requiring

exact matches within the top ten terms out of the 1.5 million total terms in the collection is a rather stringent measure of performance.

Only the top ten terms are shown here, due to space limitations. However, examination of the top 100 terms demonstrated comparable degrees of similarity in each case. In general, the higher the indicated overlap for the top terms in each example, the greater the similarity would be for the results of any query related to or containing the given phrase.

5.3.1 Example 1

For this example, the query phrase was *gross national product*. Table 2 shows the closest ten term results for this phrase used as a query, for each of the three phrase processing variants.

Table 2: Top ten term results for query = gross national product.

	PHRASE PROCESSING		
	NONE	PRE-PROCESSED	AD HOC
1.	product	lithuanian_ministry_of_finance	gdp
2.	national	expenditures	gross
3.	gross	estonian_statistics__office	expenditures
4.	projected	icelandic_treasury	lithuanian_ministry_of__finance
5.	flajs	unipolarization	dependent
6.	andrej_flajs	spkef	dependency
7.	forecasts	economy	gdps
8.	projections	economic	estonian_statistics_office
9.	grow	pajula	icelandic_treasury
10.	slower	asfinag	expenditure

For the case where there was no phrase processing, the top three (closest) terms correspond to the individual terms in the phrase. This is classical LSI - treating the phrase as a straightforward combination of the meanings of its individual terms. Terms five and six refer to Andrej Flajs, who was the lead author on a study of gross national income which is frequently referenced in the articles of the collection. Terms four and seven through ten are all generic terms that occur frequently in articles in the context of discussion of gross national product (GNP).

For the case where the phrase was treated as a unit in pre-processing, the top ten results are quite different. Three of the top four terms are names of organizations. During the time frame of the articles, there was extensive news coverage of the European financial crisis. These three organizations were frequently mentioned in news articles related to GNP changes in that time frame. SpKef was the largest savings bank in Iceland, which failed. Hardo Pajula is an economic analyst who wrote extensively on the Euro Zone crisis in this time period. ASFiNAG is a corporation that was a significant contributor to high Austrian debt in this time frame.

It is interesting that there is *no* overlap among the top ten terms for these two cases. The classical implementation of the LSI query yields terms that are generically related to a discussion of gross national product. However, as indicated, identifying those three words as constituting an important phrase, and treating it as a unit in the LSI pre-processing, yields much more specific results. The results focus on key entities central to discussions of GNP in these specific documents.

For the ad hoc phrase processing approach described in this paper, four of the top ten terms overlap with the top ten terms retrieved in the full phrase pre-processing instance. In fact, these four overlap terms are the top four terms retrieved in the pre-processed case.

5.3.2 Example 2

For this example, the query phrase was *rare earth element*. (Rare earth elements are members of a subgroup of the periodic table of elements, plus scandium and yttrium). Table 3 shows the closest ten terms for this phrase for each of the three phrase processing variants.

Table 3: Top ten term results for query = rare earth element.

	PHRASE PROCESSING		
	NONE	PRE-PROCESSED	AD HOC
1.	earth	praseodymium	dysprosium
2.	earths	dysprosium	rare
3.	rare	rhodia	bastnasite
4.	planetary	association_of_china_rare_earth	praseodymium
5.	planets	molycorp	molycorp
6.	comets	scandium	superfund
7.	asteroids	nechalacho	molycorp_inc
8.	jpl	molycorp_inc	association_of_china__rare_earth
9.	hi_tech_co	jia_yinsong	nechalacho
10.	asteroid	bastnasite	su_bo

For the case of no phrase processing, it is clear that the results are driven by the term *earth*. The terms *rare* and *element* are used in multiple contexts, but, in this collection, *earth* is discussed almost always in the context of being a planet. Accordingly, seven of the top ten terms in this case have the context of *celestial bodies*. JPL is an acronym for Jet Propulsion Laboratory, a leading US organization for space exploration.

As in the first example, there is *no* overlap between the top ten results for the case of no phrase processing and the results for the pre-processed case. In the pre-processed case, the top two terms and the sixth term are names of rare earth elements. *Rhodia* is the name of a leading rare earth production company. The fourth term in the listing is the *Association of China Rare Earth Industry*, from which the entity extractor has dropped the word *industry*. *Molycorp* is a mining company that is a major supplier of rare earth ores. *Nechalacho* is the site of a major mining activity for rare earth ores. *Jia Yinsong* is chief of the Rare Earth Office of the Ministry of Industry and Information Technology in China. Bastnasite is the most abundant rare earth element mineral.

In this example, the contexts of the two result lists for the no-phrase-processing and the phrase-pre-processing instances are completely different. Whereas in the previous case, a user might have been willing to work with the more general results given by classical LSI for *gross national product*; in this case, the results with no phrase processing are completely unrelated to the intent of the query *rare earth element*.

In this example, the ad hoc approach has performed very well. Seven of the top ten terms from the phrase-pre-processing instance occur among the top ten terms for the ad hoc case. This includes the top two terms from the pre-processed case. Su Bo is Vice Minister of Industry in China. The news articles contain excerpts from a number of speeches that he made about rare earth elements in this time frame.

In this case, the phrase is highly non-compositional; i.e., its meaning is not a simple combination of the meanings of its constituent words. (Actually it is not a simple combination of the dominant senses of the constituent terms in this collection). The technique described here has its greatest impact for such phrases. For this type of query, use of the technique described here would have a significant beneficial impact on user satisfaction.

5.3.3 Example 3

For this example, the query phrase was *highly enriched uranium*. Table 4 shows the closest ten terms for this phrase for each of the three phrase processing variants.

For the case with no phrase processing, the terms are generally related to uranium enrichment. Fordo is the site of a uranium enrichment plant in Iran.

For the case where the phrase was treated as a unit in pre-processing, the top ten results are quite different – there is only one term that also occurred in the top ten for the no-phrase-processing case.

With phrase pre-processing, the acronym for *highly enriched uranium* (HEU) is the top term, as might be expected. Nuclear Threat Initiative is a nonprofit organization dedicated to reducing the spread of weapons of mass destruction. CPPNM is an acronym for Convention on the Physical Protection of Nuclear Material. Robert Gallucci is president of the MacArthur foundation. Miles Pomper is a Senior Research Associate at the Center for Nonproliferation Studies. Frank N. von Hippel is Co-Director of the Program on Science and Global Security at Princeton University. Multiple news articles from the collection dealt with papers that these individuals had written and speeches that they had given related to nuclear proliferation.

Table 4: Top ten term results for query = highly enriched uranium.

	PHRASE PROCESSING		
	NONE	PRE-PROCESSED	AD HOC
1.	uranium	heu	plutonium
2.	enriched	plutonium	heu
3.	enrichment	fissile	fissile
4.	weapons-grade	nuclear_threat_initiative	weapons-grade
5.	fordo	weapons-grade	nonproliferation
6.	international_atomic_energy_agency	robert_gallucci	enriched
7.	iaea	nonproliferation	nuclear
8.	centrifuges	cppnm	gary_samore
9.	centrifuge	miles_pomper	nuclear_threat__initiative
10.	enriching	hippel	siegfried_hecker

The ad hoc approach performed well for this phrase. Six of the top ten result terms from full phrase pre-processing occur in the ad hoc result list,

including the top five terms from the pre-processed case.

The other terms presented also have the appropriate context. Gary Samore is the Executive Director for Research at the Belfer Center for Science and International Affairs. Siegfried Hecker was co-director of the Center for International Security and Cooperation from 2007-2012. Multiple articles from the collection discussed speeches given and papers written by these two individuals that dealt with nuclear proliferation.

A summary of the term overlap results from these examples is shown in Table 5.

An interesting aspect of the term results is their specificity. Considering the three examples taken together, only 23% of the terms produced by classical LSI corresponded to named entities. More than three-fourths were general terms. The pre-processed results were much more specific. More than half (57%) corresponded to named entities. The ad hoc technique described here produced results intermediate between these – 40% of those results were named entities.

Table 5: Comparison of Result Sets.

Phrase	Overlap of Result Sets with those for full phrase pre-processing			
	No Phrase Pre-processing		Ad Hoc Technique	
	Top 10 Terms	Top 100 Docs	Top 10 Terms	Top 100 Docs
gross national product	0	12%	40 %	52%
rare earth element	0	15%	70%	79%
highly enriched uranium	10%	3%	60%	47%

5.3.4 Document Result Comparisons

Examination of the results for retrieved documents showed patterns similar to those for terms. In these tests, each of the three example phrases was used as a query and the 100 highest-ranked documents were retrieved. These results also are shown in table 5. They show differences similar to those for terms. For the news articles used in the testing, there are many similar documents. Requiring exact matches within the top 100 documents out of 1.6 million is a relatively strict measure of performance.

Although space limitations preclude showing other examples, all phrases tested have shown results similar to those in Table 5. In general, the results are most striking when the query phrase is highly non-compositional. Overall, for term retrieval, the average overlap between the no-phrase-processing results and those for full phrase pre-processing is only a few percent. For document retrieval, the average overlap is only about 10%. For the technique described in this paper, the average overlap with full phrase pre-processing for both types of retrieval is close to 60%.

The test data demonstrate that the retrieval results generated using the proposed technique constitute a useful approximation of those that would have been obtained if the given phrases had been treated as indexing units during the creation of the LSI spaces.

6 CONCLUSIONS

The method described here provides an approach for using arbitrary phrases as queries in an LSI space after the LSI index has been created. Tests demonstrate that the results of such queries are a useful approximation of the results that would have been obtained if those phrases had been treated as indexing units during the creation of the LSI space. This is true both when retrieving closest terms and closest documents. In contemporary hardware environments, the technique is fast enough that it can be used for near-real-time processing at query time.

The implementation of this technique in LSI-based information systems can be anticipated to yield improvements in user satisfaction. Users are accustomed to being able to use phrase searches in Boolean retrieval systems and find their absence in LSI systems a limitation. There also are clear improvements in performance for many queries. In general, the precision of results obtained using the technique described here will be much greater than that obtained by simply including the individual terms of the phrase in a standard LSI query. This is particularly true for highly non-compositional phrases.

The technique described here is equally applicable for English- and for foreign-language text. It also has direct analogues for dealing with coupled features in LSI spaces used to represent non-textual data.

Some advanced knowledge discovery applications employ workflows in which LSI queries are automatically generated. Such applications typically emphasize generation of high-precision result sets. The technique described here is particularly applicable to such environments.

ACKNOWLEDGEMENTS

The author would like to thank the members of the Semantic Engineering staff at Agilex Technologies who participated in the reported testing. He would also like to thank the anonymous referees, who made suggestions that significantly improved the quality of the paper.

REFERENCES

Bradford, R., 2009. Comparability of LSI and human judgment in text analysis tasks. *Proceedings, Applied Computing Conference, Athens, Greece,* 359-366.

Bradford, R., 2011. Implementation techniques for large-scale latent semantic indexing applications. *Proceedings, ACM Conference on Information and Knowledge Management, Glasgow, Scotland, October, 2011.*

Broschart, A., Berberich, K., Schenkel, R., 2010. Evaluating the potential of explicit phrases for retrieval quality. *Proceedings, ECIR 2010,* 623-626.

Dumais, S., 2004. Latent semantic analysis. *ARIST Review of Information Science and Technology,* vol. 38, Chapter 4.

Dumais, S., et al, 1988. Using latent semantic analysis to improve access to textual information. *Proceedings, CHI 88, June 15-19, 1988, Washington, DC*, 281-285.

Fagan, J., 1989. The effectiveness of a nonsyntactic approach to automatic phrase indexing for document retrieval. *JASIS,* 40(2), 115-132.

Furnas, G., et al, 1988. Information retrieval using a singular value decomposition model of latent semantic structure. *Proceedings 11th SIGIR,* 465-480.

Grönqvist, L., 2005A. An evaluation of bi- and tri-gram enriched latent semantic vector models. *ELECTRA Workshop, Methodologies and Evaluation of Lexical Cohesion Techniques in Real-world Applications*, Salvador, Brazil, 19 August, 2005, 57–62.

Grönqvist, L., 2005B. Evaluating latent semantic vector models with synonym tests and document retrieval. *ELECTRA Workshop, Methodologies and Evaluation of Lexical Cohesion Techniques in Real-world Applications*, Salvador, Brazil, 19 August, 2005, 86–88.

Grönqvist, L., 2006. *Exploring Latent Semantic Vector Models Enriched With N-grams.* PhD Thesis, Växjö University, Sweden.

Harmon, D., 2005. The TREC ad hoc experiments. *In TREC: Experiment and Evaluation in Information Retrieval, Voorhees and Harmon, eds,* MIT Press.

Hulth, A., 2004. *Combining Machine Learning and Natural Language Processing for Automatic Keyword Extraction.* Thesis, Stockholm University, April, 2004.

Jiang, M., et al, 2004. Choosing the right bigrams for information retrieval. *Proceeding of the Meeting of the International Federation of Classification Societies, 2004*, 531-540.

Kim, H-R., and Chan, P., 2004. Identifying variable-length meaningful phrases with correlation functions. *Proceedings, ICTAI, 2004, 16th IEEE International Conference on Tools with Artificial Intelligence*, 30-38.

Kraaij, W., and Pohlmann, R., 1998. Comparing the effects of syntactic vs. statistical phrase indexing strategies for Dutch. *Proceedings, ECDL 98,* LNCS 1513, 605-617.

Lizza, M., and Sartoretto, F., 2001. A comparative analysis of LSI strategies. *In Computational Information Retrieval, M. Berry ed.,* SIAM, 171-181.

Manning, C., Raghavan, P., and Schütze, H., 2008. *Introduction to Information Retrieval*, Cambridge University Press, 36.

Metzler, D., Strohman, T., Croft, W., 2006. Indri at TREC 2006: lessons learned from three terabyte tracks. *Proceedings, Fifteenth Text REtrieval Conference,* NIST Special Publication SP 500-272.

Mitra, M., et al, 1997. An analysis of statistical and syntactic phrases. *Proceedings of RIAO 97, Montreal, Canada*, 200-214.

Nakov, P., Valchanova, E., and Angelova, G. 2003. Towards deeper understanding of the LSA performance. In *Proceedings, Recent Advances in Natural Language Processing*, 2003, 311-318.

Ogawa, Y., et al, 2000. Structuring and expanding queries in the probabilistic model. *Proceedings, Ninth Text REtrieval Conference (TREC-9),* NIST Special Publication 500-249, 427-435.

Olney, A. 2009. Generalizing latent semantic analysis. In *Proceedings, 2009 IEEE International Conference on Semantic Computing*, 40-46.

Salton, G., Yang, C., Yu, T., 1975. A theory of term importance in automatic text analysis. *JASIS,* 26(1), 33-44.

Turpin, A., and Moffat, A., 1999. Statistical phrases for vector-space information retrieval. *Proceedings, SIGIR 99, Berkley, CA, August 1999*, 309-310.

Weimer-Hastings, P. 2000. Adding syntactic information to LSA. In *Proceedings of the 22nd Annual Meeting of the Cognitive Science Society.*

Wu, H., and Gunopulos, D. 2002. Evaluating the utility of statistical phrases and latent semantic indexing for text classification. *Proceedings ICDM,* 713-716.

Zhai, C., et al, 1996. Evaluation of syntactic phrase indexing – CLARIT NLP track report. *In Proceedings, Fifth TExt Retrieval Conference,* NIST Special Publication 500-238, 347-358.