Enhancing Multilingual Latent Semantic Analysis with Term Alignment Information

Brett W. Bader
Computer Science & Informatics Department
Sandia National Laboratories
P. O. Box 5800, MS 1318
Albuquerque, NM 87185-1318, USA
bwbader@sandia.gov

Peter A. Chew
Cognitive Systems Research & Applications Department
Sandia National Laboratories
P. O. Box 5800, MS 1011
Albuquerque, NM 87185-1011, USA
pchew@sandia.gov

Abstract

Latent Semantic Analysis (LSA) is based on the Singular Value Decomposition (SVD) of a term-by-document matrix for identifying relationships among terms and documents from co-occurrence patterns. Among the multiple ways of computing the SVD of a rectangular matrix X, one approach is to compute the eigenvalue decomposition (EVD) of a square 2×2 composite matrix consisting of four blocks with X and X^T in the off-diagonal blocks and zero matrices in the diagonal blocks. We point out that significant value can be added to LSA by filling in some of the values in the diagonal blocks (corresponding to explicit term-to-term or document-to-document associations) and computing a term-by-concept matrix from the EVD. For the case of multilingual LSA, we incorporate information on cross-language term alignments of the same sort used in Statistical Machine Translation (SMT). Since all elements of the proposed EVD-based approach can rely entirely on lexical statistics, hardly any price is paid for the improved empirical results. In particular, the approach, like LSA or SMT, can still be generalized to virtually any language(s); computation of the EVD takes similar resources to that of the SVD since all the blocks are sparse; and the results of EVD are just as economical as those of SVD.

1 Introduction

It is close to two decades now since Deerwester et al. (1990) first proposed the application of the Singular Value Decomposition (SVD) to term-by-document arrays as a statistics-based way of representing how terms and documents fit together within a semantic space. Since the approach was supposed to 'get beyond' the terms themselves to their underlying semantics, the approach became known as Latent Semantic Analysis (LSA).

Soon after this application of SVD was widely publicized, it was suggested by Berry et al. (1994) that, with a parallel corpus, the approach could be extended to pairs of languages to allow cross-language information retrieval (IR). It has since been confirmed that LSA can be applied not just to pairs of languages, but also simultaneously to *groups* of languages, again given the existence of a *multi*-parallel corpus (Chew and Abdelali 2007).

In this paper, we return to the basics of LSA by examining its relationship with SVD, and, in turn, the mathematical relationship of SVD to the eigenvalue decomposition (EVD). These details are discussed in section 2. It has previously been suggested (for example, in Hendrickson 2007) that IR results could be improved by filling in information beyond that available directly in the term-by-document matrix, and replacing SVD with the more general EVD. To our knowledge, however, these suggestions have not been publicized outside the mathematics community, nor have they been empirically tested in IR applications. With multilingual information retrieval as a use case, we consider alternatives in section 3 for implementation of this idea. One of these re-

Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pages 49–56
Manchester, August 2008

lies on no extraneous information beyond what is already available in the multi-parallel corpus, and is based entirely on the statistics of cross-language term alignments. 'Regular' LSA has been shown to work best when a weighting scheme such as log-entropy is applied to the elements in the term-by-document array (Dumais 1991), and in section 3 we also consider various possibilities for how the term alignments should best be weighted. Section 4 recapitulates on a framework that allows EVD with term alignments to be compared with a number of related approaches (including LSA without term alignments). This is a recapitulation, because the same testing framework has been used previously (for other linear-algebra based approaches) by Chew and Abdelali (2007) and Chew et al. (2007). The results of our comparison are presented and discussed in section 5, and we conclude upon these results and suggest further avenues for research in section 6.

2 The relationship of SVD to EVD, and its application to information retrieval

In the standard LSA framework (Deerwester et al. 1990) the (sparse) term-by-document matrix X is factorized by the singular value decomposition (SVD),

$$X = USV^T \qquad (1)$$

where U is an orthonormal matrix of left singular vectors, S is a diagonal matrix of singular values, and V is an orthonormal matrix of right singular vectors (Golub and van Loan 1996).

Typically for LSA, a *truncated* SVD is computed such that equality in (1) no longer holds and that the best rank-R least-squares approximation to matrix X is formed by keeping the R largest singular values in S and discarding the rest. This also means that the first R vectors of U and V are retained, where R indicates the number of concept dimensions in LSA. Each column vector in U maps the terms to a single arbitrary concept, such that terms which are semantically related (as determined by patterns of co-occurrence) will tend to be grouped together with large values in columns of U.

There are many ways to compute the SVD of a sparse matrix. One expedient way is to compute the eigenvalue decomposition (EVD) of either X^TX or XX^T, depending on the largest dimension of X, to obtain U or V, respectively. With U or V, one may compute the rest of the SVD by a simple matrix-matrix multiplication and renormalization.

Another way to compute the SVD is to compute the eigenvalue decomposition of the 2-by-2 block matrix

$$B = \begin{bmatrix} 0 & X \\ X^T & 0 \end{bmatrix}.$$

The eigenvalues of B are the singular values of X, replicated as both positive and negative, plus a number of zeroes if X is not square. The left and right singular vectors are contained within the eigenvectors of this composite matrix B. Assume that X is of size $m \times n$ and that $m \geq n$, with left singular vectors $U = [U_n \quad U_{m-n}]$, where U_n corresponds to the n positive singular values and U_{m-n} corresponds to the remaining m-n zero singular values. Let Q denote the orthogonal matrix of eigenvectors corresponding to the nonnegative eigenvalues of B, then the matrices of left and right singular vectors are stacked on top of each other, U on top of V, as follows:

$$Q = \frac{1}{\sqrt{2}} \begin{bmatrix} U_n & \sqrt{2} \times U_{m-n} \\ V & 0 \end{bmatrix}.$$

Hence, one may compute the truncated SVD of X by computing only the eigenvectors corresponding to the largest R eigenvalues and then extracting and rescaling the U and V matrices from Q.

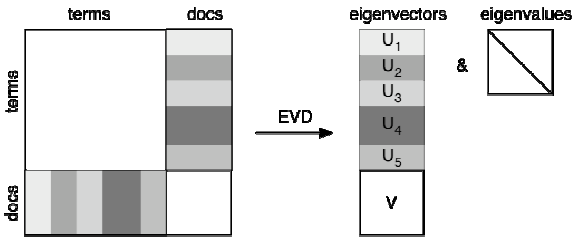

Figure 1. Eigenvalue decomposition in multilingual information retrieval

In the context of multilingual LSA using a parallel corpus, the block matrix B is depicted in Figure 1, where the terms are shaded according to each language. Each language may have a different number of terms, so the language blocks are not expected to be the same size as one another. The eigenvectors and eigenvalues of B are also shown.

We may obtain a pair of U and S matrices for each language by extracting the corresponding partition of U from the eigenvectors. We desire each language-specific U matrix to have columns of unit length, which we accomplish by computing the length of each of its columns and then

rescaling the columns of U by the inverse length and multiplying the eigenvalues by these lengths for our S matrix. We call this approach 'Tucker1' because the result is identical to creating a U and S matrix for each language from the general Tucker1 model found by three-way analysis of the terms-by-documents-by-language array (Tucker 1966).

For applications in information retrieval, we usually want to compute a measure of similarity between documents. Once we have U and S, we can estimate similarities by computing the cosine of the angle between the document vectors in the smaller 'semantic space' of the R concepts found by LSA. New documents in different languages can be projected into this common semantic space by multiplying their document vectors (formed in exactly the same way as the columns for X) by the product US^{-1}, to yield a document-by-concept vector.

3 From SVD to term-alignment-based EVD

If we compute just the SVD of a term-document matrix X, then the technique we use to accomplish this (whether computing the EVD of the block matrix B or otherwise) is immaterial from a computational linguist's point of view: there is no advantage in one technique over another. However, the technique of EVD allows one to augment the LSA framework with additional information beyond just the term-document matrix. In Figure 1, the two diagonal blocks contain only zeroes, but we envision augmenting B with term alignment information such that the upper diagonal block captures any term-to-term similarities. Additional term-term alignment information serves to enhance the term-by-concept vectors in U by providing explicit, external knowledge so that LSA can learn more refined concepts. While not explored in this paper, we also envision incorporating any document-to-document similarities into the lower diagonal block.

Let D_1 and D_2 denote symmetric matrices. We augment the block matrix B and redefine it as a more general symmetric matrix,

$$B = \begin{bmatrix} D_1 & X \\ X^T & D_2 \end{bmatrix}.$$

If both D_1 and D_2 are equal to the identity matrix, then the eigenvalues of B are shifted by one, but the eigenvectors are not affected.

Since our use case here is multilingual information retrieval, imagine for the moment that an oracle provides dictionary information that matches up words in each of our language pairs (Arabic-English, Arabic-French, etc.) by meaning. Thus, for example, we might have a pairing between English *house* and French *maison*. This information may be encoded in the diagonal block D_1 by replacing zeroes in the cells for (*house*, *maison*) and its symmetric entry with some nonzero value indicating the strength of association for the two terms. Completing all relevant entries in D_1 in this fashion serves to strengthen the co-occurrence information in the parallel corpus that LSA normally finds via the SVD.

In the simplest approach, if the oracle indicates a match between two terms *i* and *j*, then a one could be inserted in D_1 at positions (*i*,*j*) and (*j*,*i*). If D_1 were filled with such term alignment information, the matrix B would still be sparse. Without any document-document information, then D_2 could be either the identity matrix or the zero matrix. Our experience has shown that $D_2 = 0$ works slightly better in practice. Figure 2 shows a block matrix augmented with term alignments in this fashion.

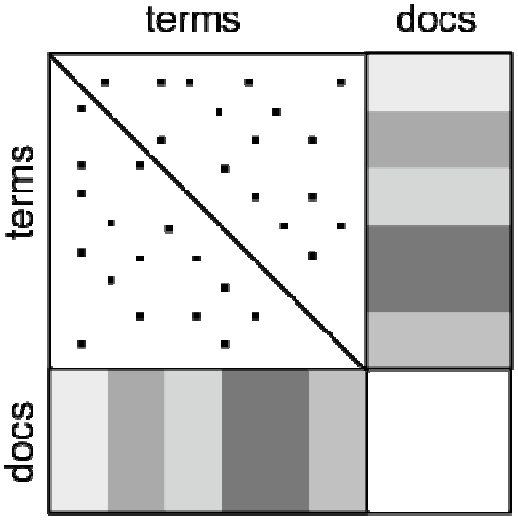

Figure 2. Augmented block matrix with term alignments

The eigenvalue decomposition of B now incorporates this extra term information provided in D_1, and the eigenvectors show stronger correspondence between those terms indicated. However, with each term aligned with one or more other terms, the row and column norms of D_1 are unequal, which means that some terms may be biased to appear more heavily in the eigenvectors. In addition, the magnitude or 'weight' of D_1 relative to X needs to be considered, otherwise the explicit alignments in D_1 and the co-

occurrence information in X may be out of balance with one another. Properly normalizing and scaling D_1 may mitigate both of these risks.

There are several possibilities for normalizing the matrix D_1. Sinkhorn balancing (Sinkhorn 1964) is a popular technique for creating a doubly stochastic matrix (rows and columns all sum to 1) from a square matrix of nonnegative elements. Sinkhorn balancing is an iterative algorithm in which, at each step, the row and column sums are computed and then subsequently used to rescale the matrix. For balancing the matrix A, each iteration consists of two updates

$A \leftarrow W_R A$

$A \leftarrow A W_C$

where W_R is a diagonal matrix containing the inverse of row sums of A, and W_C is a diagonal matrix containing the inverse of column sums of A. This algorithm exhibits linear convergence, so many iterations may be needed. The algorithm may be adapted for normalizing the row and column vectors according to any norm. Our experience has shown that normalizing D_1 with respect to the Euclidean norm works well in practice.

In terms of scaling D_1 relative to X, we simply multiply D_1 by a positive scalar value, which we denote with the variable β. The optimal value of β appears to be problem dependent.

Let us return for the moment to the question of how we populate D_1 in the first place, and what each entry in that block represents. In the simple case described above, the existence of a 1 at position (*i,j*) indicates that an alignment exists between terms *i* and *j*, and a zero indicates that no alignment exists. But in reality, a binary encoding like this may be too simplistic. In this respect, it is instructive to consider how we populate D_1 in the light of the weighting scheme used for X, since the latter is discussed in Dumais (1991) and is by now quite well understood.

In the simplest case, an entry of 1 in X at position (*i,j*) can denote that term *i* occurs in document *j*, just as in our simple case with D_1. A slightly more refined alternative is to replace 1 with $f_{i,j}$, where $f_{i,j}$ denotes the raw frequency of term *i* within document *j*. But, as Dumais (1991) shows, it is significantly better in practice to use a 'log-entropy' weighting scheme. This adjusts $f_{i,j}$ first by 'dampening' high-frequency terms (using the log of the frequency), and secondly by giving a lower weight to terms which occur in many documents.[2] The former adjustment is related to an insight from Zipf's law, which is that the dampened term frequency will be in proportion to the log of the term's rank in frequency. The latter adjustment is based on information theory; a term which is scattered across many documents (such as 'and' in English) has a high entropy, and therefore lower intrinsic information content.

Suppose, therefore, that our 'dictionary' oracle could not only indicate the existence of an alignment, but also provide some numerical value for the strength of association between two aligned terms. (In practice, this is probably more than one could hope for even from the best published bilingual dictionaries.) This information could then replace the ones in D_1 prior to Sinkhorn balancing and matrix weighting.

While one cannot expect to obtain this information from published dictionaries, there is in fact a statistical approach to gathering the necessary information, which we borrow from SMT (Brown et al. 1994). All that is required is the existence of a parallel corpus, which we already have in place for multilingual LSA.

Here, an entry $f_{i,j}$ in D_1 is based on the mutual information of term *I* and term *J*, or I(*I;J*) (capitals are used to indicate that the terms are treated here as random variables). It is an axiom that:

$$I(I;J) = H(I) + H(J) - H(I,J) \quad (2)$$

where H(*I*) and H(*J*) are the marginal entropies of *I* and *J* respectively, and H(*I,J*) is the joint entropy of *I* and *J*. Properties of H(*I,J*) include the following:

$$H(I,J) \geq H(I) \geq 0$$
$$H(I,J) \geq H(J) \geq 0$$
$$H(I,J) \leq H(I) + H(J) \quad (3)$$

Considering (2) and (3) together, it should be clear that I(*I;J*) will range between 0 and the maximum value for H(*I*) or H(*J*).

For the purposes of populating D_1, we compute the entropy of a term *i* by considering the number of documents where *i* occurs, and the number of documents where *i* does not occur, and express these as probabilities. For the joint entropy H(*I,J*), we need to compute four probabilities based on all the possibilities: documents where both terms occur, those where *I* occurs without *J*, those where *J* occurs without *I*, and

[2] One can also raise the global weight in the log-entropy scheme to a power (which we denote with the variable α). Selecting $\alpha \neq 1$ can, in practice, yield better results in the applications we have tested.

those where neither occur. The result of this is that a numerical value is attached to each alignment: higher values indicate that terms are strongly correlated, and lower values indicate that one term predicts little about the other. For each pair of words (*i*,*j*) which co-occur in any text chunk in the parallel corpus, we can say that an alignment exists if, among all the possibilities, mutual information for *i* is maximized by selecting *j, and* vice versa. (Since the maximization of mutual information is not necessarily reciprocal, the effect of this is to be conservative in postulating alignments.) The weight of this alignment is its mutual information (equivalent to the 'global weight' of log-entropy) multiplied by the log of one plus the number of text chunks in which that alignment appears (equivalent to the 'local weight' of log-entropy).

Some examples of English-French pairs at either end of this spectrum (where mutual information is non-zero) are given in Table 1.

I(*I*;*J*)	Alignment weight	*I*	*J*
0.000176	0.000176	hearing	écoutait
0.000217	0.000217	misery	misérable
…			
0.270212	2.884297	house	maison
0.321754	3.506663	king	roi
0.415702	6.025456	and	et
0.472925	5.798080	I	je

Table 1. Term alignment and mutual information

We believe that this approach, which weights alignments based on mutual information, fits very well with the log-entropy scheme used for X, since both are solidly based on the same foundation of information theory.

All together, we call this particular process LSATA, which stands for LSA with term alignments.

4 Testing framework

Since the inception of the Cross-Language Evaluation Forum (CLEF) in 2000, there has been growing interest in cross-language IR, and a number of parallel corpora have become available (for example through the Linguistic Data Consortium). Widely used examples include the Canadian Hansard parliament proceedings (in French and English). Harder to obtain are *multi*-parallel corpora – those where the same text is translated into more than two parallel languages.

One such corpus which has not yet gained wide acceptance, perhaps owing to the perception that it has less relevance to real-world applications than other parallel corpora, is the Bible. Yet the range of languages covered is unarguably unmatched elsewhere, and one might contend that its relevance is in some ways greater than, say, Hansard's, as its impact on Western culture has been broader than that of Canadian government debates. Similarly, the Quran, while not translated into as many languages as the Bible, has had a significant impact on another large segment of the world's population.

But the relevance or otherwise of the Bible and/or Quran, and the extent to which they have been accepted by the computational linguistics community at large as parallel corpora, are somewhat beside the point for us here. Our interest is in developing theory and applications which have universal applicability to as many languages as possible, regardless of the subject matter or whether the languages are ancient or modern. One might compare this approach to Chomsky's quest for Universal Grammar (Chomsky 1965), except that the theory in our case is based on lexical statistics and linear algebra rather than rule-based generative grammar.

The Bible and Quran have in fact previously been used for experiments similar to ours (e.g., Chew et al. 2007). By using these texts as parallel corpora, therefore, we facilitate direct comparison of our results with previous ones. But besides this, the Bible has some especially attractive properties for our current purposes. First, the carefulness of the translations means that we are relatively unlikely to encounter situations where cross-language term alignments are impossible because some text is missing in one of the translations. Secondly, the relatively small size of the parallel text chunks (by and large, each chunk is a verse, most of which are about a sentence in length) greatly facilitates the process of statistical term alignment. (This is based on the combinatorics: the number of possible term-to-term alignments increases approximately quadratically with the number of terms per text chunk.)

Thus, our framework is as follows. In our term-by-document matrix X, the documents are verses, and the terms are distinct wordforms in any of the five languages used in the test data in Chew et al. (2007): Arabic (AR), English (EN), French (FR), Russian (RU) and Spanish (ES). As in Chew et al. (2007), too, our test data consists of the text of the Quran in the same 5 languages. In this case, the 'documents' are the 114 parallel suras (or chapters) of the Quran. We obtained all translations of the Bible and Quran from openly-

available websites such as that of Biola University (2005-2006) and http://www.kuran.gen.tr.

As already mentioned, SVD of a term-by-document matrix is equivalent to EVD of a block matrix in which two of the blocks (the non-diagonal ones) are X and X^T. As described in section 3, we fill in some of the values of D_1 with nonzeroes (from term alignments derived from the Bible). In all cases (both SVD and EVD), we performed a truncated decomposition in either 60, 240, or 300 dimensions.

SVD/EVD dimensions	Type of decomposition	Include term alignments? / weighting type	Term alignment settings		Global weight α*	Average P1	Average MP5
			Sinkhorn balanced?	β			
60	SVD	N/A			1.8	0.7116	0.5702
	Tucker1					0.7170	0.5770
	PARAFAC2					0.7420	0.6580
	LSATA	yes (binary)	no	N/A	1.8	0.7000	0.5691
			yes	4.0		0.7611	0.6474
				1.0	1.6	0.7716	0.5972
				4.0		0.7979	0.6467
		yes (log-MI)	no	N/A	1.8	0.6481	0.3804
			yes	1.0		0.7393	0.5972
				12.0		0.8088	0.6972
				1.0	1.6	0.7488	0.5789
				12.0		0.7933	0.6586
240	SVD	N/A			1.8	0.8761	0.6554
	PARAFAC2					0.8975	0.7853
300	SVD	N/A			1.8	0.8796	0.6575
	LSATA	yes (binary)	yes	4.0	1.6	**0.9421**	0.7695
		yes (log-MI)		12.0	1.8	0.8982	0.8000
					1.6	0.9182	**0.8067**

*See footnote 2.

Table 2. Results with various linear algebraic decomposition methods and weighting schemes

To evaluate the different methods against one another, we use similar measures of precision as were used with the same dataset by Chew et al. (2007): precision at 1 document (P1) (the average proportion of cases where the translation of a document ranked highest among all retrieved documents of the same language) and multilingual precision at 5 documents (MP5) (the average proportion of the top 5 ranked documents which were translations of the query document into any of the 5 languages, among all retrieved documents of *any* language). By definition, MP5 is always less than or equal to P1; MP5 measures success in multilingual clustering, while P1 measures success in retrieving documents when the source and target languages are pre-specified.

5 Results and Discussion

Table 2 above presents a summary of our results. The main point to note is that the addition of information on term alignments is clearly beneficial. An approach based on the Tucker1 decomposition algorithm, without any information on term alignments, achieves P1 of 0.7170 and MP5 of 0.5770. With scaled term alignment information, the results improve to 0.7611 and 0.6474, respectively. Using a chi-squared test, we tested the significance of the increase in P1 and found it to be highly significant ($p \approx 1.7 \times 10^{-7}$).

The results also show, however, that one needs to be careful about how the word-alignment information is added. Without some form of balancing and scaling of D_1, there is little improvement (and often significant deterioration) in the results when alignment information is included.

In addition to comparing a block EVD approach with term alignments to one without, we also compared against another decomposition method, PARAFAC2, which has been found to be more effective than SVD in cross-language IR (Chew et al. 2007). Here, the results are more equivocal. P1 is slightly higher under the LSATA approach (with binary values in D_1) than

under PARAFAC2, while the reverse is true for MP5. The difference for P1 is significant at $p < 0.05$ but not at $p < 0.01$. In any case, there are risks in making a comparison between PARAFAC2 and LSATA. For one thing, PARAFAC2, as implemented here, includes no mechanism for incorporating term-alignment information. It is not clear to us yet whether such a mechanism could (mathematically or practically) be incorporated into PARAFAC2. Secondly, we are not yet confident that we have found the optimal weighting scheme for the D_1 block under the LSATA model. Our experiments with different weighting and normalization schemes for the D_1 block are still in relatively initial stages, though it can also be seen from Table 2 that by selecting certain settings under LSATA (replacing binary weighting in D_1 with mutual-information-based weighting, and applying scaling with beta = 12.0), we were able to improve upon PARAFAC2 under both measures.

Although we have not tested all settings, Table 2 also shows our best results to date with this dataset, which have come from applying EVD to the block matrix that includes D_1. The precise optimal settings for EVD appear to depend on whether the objective is to maximize P1 or MP5. For P1, our best results (0.9421) were obtained with binary weighting, global term $\alpha = 1.6$, and $\beta = 4.0$. For MP5, the best results (0.8067) were obtained with mutual-information based weighting, $\alpha = 1.8$, and $\beta = 12.0$. It appears in all cases that D_1 needs to be balanced if it contains term alignment information.

The evidence, then, appears to be strongly in favor of incorporating information beyond term-to-document associations within an IR approach based on linear algebra. It happens that LSATA offers an obvious way to do this, while other methods such as PARAFAC2 may or may not. Here, we have examined just one form of information besides term-to-document statistics: term-to-term statistics. However, there is no reason to suppose that the results might not be improved still further by incorporating information on document-to-document associations, or for that matter associations between terms or documents and other linguistic, grammatical, or contextual objects.

6 Conclusion

In this paper, we have discussed the mathematical relationship between SVD and EVD, and specifically the fact that SVD is a special case of EVD. For information retrieval, the significance of this is that SVD allows for explicit encoding of associations between terms and documents, but not between terms and terms, or between documents and documents.

By moving from the special case of SVD to the general case of EVD, however, we open up the possibility that additional information can be encoded prior to decomposition. We have examined a particular use case for SVD: multilingual information retrieval. This use case presents an interesting example of additional information which could be encoded on the term-by-term diagonal block: cross-language pairings of equivalent terms (such as *house*/*maison*). Such pairs can be obtained from bilingual dictionaries, but we can save ourselves the trouble of obtaining and using these. Multilingual LSA requires that a parallel corpus have already been obtained, and well-understood statistical term alignment procedures can be applied to obtain cross-language term-to-term associations. Moreover, if the corpus is *multi*-parallel, we can ensure that the statistical basis for alignment is the same across all language pairs.

Our results show that by including term-to-term alignment information, then performing EVD, we can improve the results of cross-language IR quite significantly.

It should be pointed out that while we have successfully used statistics-based information in the term-by-term diagonal block, there is no reason to suppose that similar or better results might not be achieved by manually filling in nonzeroes in either diagonal block. The additional information encoded by these nonzeroes could include associations known a priori between documents (e.g., they were written by the same author) or terms (e.g., they occur together in a thesaurus), or both. While in these examples the additional information required might not be available from the training corpus, and its encoding could involve moving away from an entirely statistics-based model, the additional effort could be justified depending upon the intended application.

In future work, we would like to examine in particular whether still further statistically-derivable (or readily available) data could be incorporated into the model. For example, one can conceive of a block EVD involving 'levels' beyond the 'term level' and the 'document level'. In a 3×3 block EVD, for example, one might include n-grams, terms, and documents; this approach should also be extensible to essentially all languages. Might the addition of further informa-

tion lead to even higher precision? Avenues for research such as this raise their own questions, such as the type of weighting scheme which would have to be applied in a 3×3 block matrix.

In summary, however, our results give us some confidence that there can be significant benefit in making more linguistic and/or statistical information available to linear algebraic IR approaches such as EVD. Cross-language term alignments are just one example of the type of additional information which could be included; we believe that future research will uncover many more similar examples.

Acknowledgement

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

References

Michael W. Berry, Susan T. Dumais., and G. W. O'Brien. 1994. Using Linear Algebra for Intelligent Information Retrieval. *SIAM: Review* 37, 573-595.

Biola University. 2005-2006. *The Unbound Bible*. Accessed at http://www.unboundbible.org/ on Jan. 29, 2008.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1994. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics* 19(2), 263-311.

Peter A. Chew and Ahmed Abdelali. 2007. Benefits of the 'Massively Parallel Rosetta Stone': Cross-Language Information Retrieval with over 30 Languages . *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, ACL 2007*. Prague, Czech Republic, June 23–30, 2007. pp. 872-879.

Noam Chomsky. 1965. *Aspects of the Theory of Syntax*. Cambridge, MA: MIT Press.

S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer and R. Harshman. 1990. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science* 41:6, 391-407.

Susan Dumais. 1991. Improving the Retrieval of Information from External Sources. *Behavior Research Methods, Instruments, and Computers* 23(2):229-236.

Gene H. Golub and Charles F. van Loan. 1996. *Matrix Computations*, 3rd edition. The Johns Hopkins University Press: London.

R. A. Harshman. 1972. PARAFAC2: Mathematical and Technical Notes. *UCLA Working Papers in Phonetics* 22, 30-47.

Bruce Hendrickson. 2007. Latent Semantic Analysis and Fiedler Retrieval. *Linear Algebra and its Applications* 421 (2-3), 345-355.

P. Koehn, F. J. Och, and D. Marcu. 2003. Statistical Phrase-Based Translation. *Proceedings of the Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computational Linguistics (HLT/NAACL)*, 48-54.

P. Koehn. 2002. Europarl: a Multilingual Corpus for Evaluation of Machine Translation. Unpublished, accessed on Jan. 29, 2008 at http://www.iccs.inf.ed.ac.uk/~pkoehn/publications/europarl.pdf.

Philip Resnik, Mari Broman Olsen, and Mona Diab. 1999. The Bible as a Parallel Corpus: Annotating the "Book of 2000 Tongues". *Computers and the Humanities*, 33: 129-153.

R. Sinkhorn. 1964. A Relation between Arbitrary Positive Matrices and Doubly Stochastic Matrices. *Annals of Mathematical Statistics* 35 (2), 876-879.

Ledyard R. Tucker. 1966. Some Mathematical Notes on Three-mode Factor Analysis, *Psychometrika* 31, 279-311.

Ding Zhou, Sergey A. Orshanskiy, Hongyuan Zha, and C. Lee Giles. 2007. Co-Ranking Authors and Documents in a Heterogeneous Network. *Seventh IEEE InternationalConference on Data Mining*, 739-744.