Approximate Dimension Reduction at NTCIR

Fan JIANG Michael L. LITTMAN
Department of Computer Science
Duke University
Durham, NC USA
{fan,mlittman}@cs.duke.edu

Abstract

We carried out a comparison of cross-language retrieval methods on the NTCIR-1 data based on dimension reduction (latent semantic indexing). These methods all use a collection parallel documents (translations or approximate translations) and very little, if any, linguistic knowledge. In NTCIR-1, we compared latent semantic indexing, local LSI, and approximate dimensional equalization (ADE). We found that local LSI and ADE performed the best on this collection and were comparable to the best performing systems reported elsewhere. We also ran ADE on the NTCIR-2 and found it fared considerably less well.
Keywords: *Cross-language retrieval, approximate dimension equalization, latent semantic indexing, local LSI.*

1 Introduction

Cross-language information retrieval (CLIR) is the problem of using ad-hoc queries in one language to retrieve documents in another language. Its importance has increased enormously in recent years because the information super-highway is reachable from virtually all over the world. The key to enable retrieval of documents in a language different from that of the queries is to establish word associations across those languages. Corpus-based IR systems achieve this by learning from bilingual parallel corpora, where corresponding documents in two collections are translations of each other or are on the same or related subjects. Vector-based dimension-reduction methods represent parallel documents as vectors in high dimensional space and "translate" words from one language into another by performing matrix computations. The advantage of these methods over others such as machine translation is that they use little, if any, linguistic knowledge. In our experiments, we compared latent semantic indexing (LSI), approximate dimension equalization (ADE), and local LSI on NTCIR-1 data. Our results submitted to NTCIR-2 were obtained with ADE.

2 Retrieval Methods

We compared a number of different vector-based retrieval methods, described in this section.

2.1 Latent Semantic Indexing

When using the original monolingual latent semantic indexing (LSI) [3], we first create a term–document matrix A from the training corpus in the same way as in the vector space model [12]. We then use singular value decomposition (SVD) [5] to factor A into three parts

$$A = U\Sigma V^T = U\ \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r)V^T,$$

where U and V are unitary matrices (i.e., $U^TU = I$ and $V^TV = I$) whose columns are the left and the right *singular vectors* of A, Σ is a diagonal matrix whose diagonal elements are non-negative and arranged in descending order, and r is the rank of A. The values $\sigma_1, \ldots, \sigma_r$ are known as the *singular values* of A, and are the square roots of the eigenvalues of A^TA and AA^T. The first k ($k \leq r$) columns of U and V and the k largest singular values of A together are the training matrix for LSI:

$$A_k = U_k\Sigma_kV_k^T.$$

We call A_k the reduced-dimensional form of A. The LSI retrieval formula is then

$$\mathrm{Sim}_{\text{LSI-ML}}(d, q) = (A_k^T\vec{d}) \cdot (A_k^T\vec{q}), \qquad (1)$$

where $\mathrm{Sim}_P(d, q)$ represents the similarity between a document and a query, $\vec{d}$ is the vector representation of document d, and $\vec{q}$ is the vector representation of document q.

Extending the idea of LSI into cross-language retrieval, we compute the SVDs of the matrices A and B of the parallel training corpora, and use their reduced-dimensional form:

$$\mathrm{Sim}_{\text{LSI-CL}}(d, q) = (A_k^T\vec{d}) \cdot (B_k^T\vec{q}).$$

Note that this formulation is different from the original cross-language LSI formulation by Landauer and Littman [8]. Their cross-language LSI formula uses a training matrix computed from the matrix that combines aligned documents from both training corpora into single documents:

$$\left[\begin{array}{c} A \\ B \end{array} \right]_k = U_{AB,k} \Sigma_{AB,k} V_{AB,k}^T.$$

Compared to this traditional application of LSI, our approach computes two separate SVDs of smaller matrices. This is certainly more useful when the combined matrix becomes too large to analyze via SVD.

2.2 Generalized Vector Space Method

The generalized vector space model (GVSM) [14], also known as "the dual space" approach [13], is a method that captures term–term correlations from the documents (or matching document pairs, in the case of cross-language retrieval) they co-occur in. Matching the vector elements in $\vec{d}$ and rows of A by the terms they represent, $A^T\vec{d}$ transforms $\vec{d}$ into a new vector whose elements correspond to the n documents in the training collection. The query vector $\vec{q}$ can also be transformed by A in the same way. Then, the query–document similarity is measured between the transformed vectors:

$$\mathrm{Sim}_{\text{GVSM-ML}}(d,q) = (A^T\vec{d}) \cdot (A^T\vec{q}) = \vec{d}^T A A^T \vec{q},$$

where the $m \times m$ matrix AA^T has a nonzero value in its row i and column j if and only if there is a document in A that contains both the i-th and j-th terms.

The extension of GVSM to cross-language IR was proposed by Yang et al. [16]. Using a bilingual collection for training, two matrices A and B are formed, where A is a term–document training matrix in the language of the retrieval documents, and B is a parallel term–document training matrix in the language of the queries. While the number of unique terms in the two languages are different, the number of documents in the training collection is the same, and are represented by the corresponding columns of A and B. Thus, when document $\vec{d}$ is transformed by A and query $\vec{q}$ by B, we can compute their inner product:

$$\mathrm{Sim}_{\text{GVSM-CL}}(d,q) = (A^T\vec{d}) \cdot (B^T\vec{q}) = \vec{d}^T A B^T \vec{q}.$$

2.3 Approximate Dimension Equalization

Approximate dimension equalization (ADE) [7] is a new method that mimics the effect of LSI with fewer computed singular vectors (i.e., a smaller k in Equation 1). This approximation algorithm is based on the consistent pattern observed from the distribution plots of singular values of many text collections—that the first few decrease sharply in magnitude and the middle majority stay relatively flat before a final drop. It turns out that we can take advantage of this so-called low-rank-plus-shift structure of the document matrix and use SVD and some matrix computations to create a new training matrix $\tilde{A}_k$:

$$\tilde{A}_k = \bar{A}_k + \frac{1}{\sigma_k} A - \frac{1}{\sigma_k} A_k,$$

where $\bar{A}_k$ is a matrix formed by the left and right singular vectors of A without the middle singular values:

$$\bar{A}_k = U_k V_k^T,$$

and σ_k is the k-th singular values of A. A detailed mathematical analysis of ADE and its training matrix is given elsewhere [7], which we will not repeat here. But, the basic idea of this $\tilde{A}_k$ matrix is that it it flattens out the first k very large singular values in the original matrix A, thus making itself to be like $A_{k'}$ with a very large $k' >> k$.

ADE's monolingual and cross-language retrieval formulae are similar to those of LSI except for the training matrix (matrices):

$$\mathrm{Sim}_{\text{ADE-ML}}(d,q) = (\tilde{A}_k^T\vec{d}) \cdot (\tilde{A}_k^T\vec{q}),$$

$$\mathrm{Sim}_{\text{ADE-CL}}(d,q) = (\tilde{A}_k^T\vec{d}) \cdot (\tilde{B}_k^T\vec{q}).$$

Experimentally, ADE have been shown to improve on LSI and inch close to VSM every time with a limited number of dimensions. ADE becomes especially useful in cross-language retrieval, where VSM is not applicable; it obtains state-of-the-art results on some of the standard test collections [7].

2.4 Local LSI

Local LSI [6] is a powerful method that combines a sampling approach to reducing the expense of SVD computation in LSI and the effectiveness of local feedback approaches [15]. Local feedback is an automated version of the *relevance feedback* method [10], wherein the top-ranked documents from an initial query-document match are "added" to the query vector for further retrieval. Basically, for each query, this method selects the top-ranked documents from an initial retrieval, computes the SVD space of this "focused" sample collection, and runs the retrieval again with the query mapped in the new space. For cross-language retrieval, we simply run the query against the training collection, and the top-ranked parallel training documents will be used for SVD computation. Note that the initial retrieval step is monolingual and can be achieved with any known retrieval method. In our experiments, we used the simplest and most efficient method: VSM.

In local feedback approaches, one of the most important issues is to figure out the number of top-ranked

documents to be used for feedback, because some top-ranked documents may not be truly relevant and thus provide a negative effect in further retrieval. The local LSI approach deals away with this problem: it lets SVD to figure out the relevance of the terms and documents in the feedback collection, be they positive or negative.

3 Results

We used the 332,918-document Japanese collection ("ntc1-j1") and 187,080-document English collection ("ntc1-e1") for our cross-language experiments. We used both the 30 training and the 53 test topics and both types of relevance judgments ("rel1" and "rel2") to demonstrate consistency in our results and simplify comparison with published results.

In processing the documents in both languages, we extracted the text from the title, abstract, and keyword fields. The English documents were stemmed and stop-word removed through the SMART system, which resulted in 208,276 unique terms. The Japanese text is coded in EUC, or Extended UNIX Code, which represents each character in Japanese by two bytes. To process the Japanese documents with our programs that handle only ASCII characters, we converted each two-byte Japanese character into a four-character "word" in ASCII. With no stemming or stop-word removal, the converted Japanese collection has 130,777 unique "words." We then indexed both collections with the SMART *Lnu* weighting:

$$\frac{1+\log(tf)}{1+\log(\text{avg } tf)} \times (1.0 - \textit{slope}) \times \textit{pivot} + \textit{slope} \times \textit{\# of unique terms},$$

where *tf* is term frequency, *slope* is some constant (we set it to 0.2), and *pivot* is the average number of unique terms across the entire collection [1].

For the topics, we extracted the title, description, narrative, and concept fields to form the queries. We discarded the concept words in English and acronyms from topics 0031–0083 as they contains both Japanese, English, and acronym concept fields. The Japanese queries were also converted into ASCII; they were then indexed with SMART *ltn* weighting:

$$(1+\ln(tf)) \times \log\left(\frac{n+1}{df}\right),$$

where n is the number of documents in the collection and *df* is the document frequency of the term.

We extracted parallel documents from the J and E collections for training. Matching documents were identified by the same document ID numbers at the beginning of a Japanese and an English document. The resulting parallel collections have 181,485 documents in each language, close to the size of the ntc1-e1 collection. We used only one-fourth, or 45,372 documents, of the parallel corpora for actual LSI and ADE training. We then applied the SMART *ntc* weighting on the training corpora, which was effective in our previous experiments on other collections:

$$tf \times \log\left(\frac{n+1}{df}\right)$$

is the "nt" part, and "c" means the above value will be normalized by the document length for each term.

The matrices created have 33,553 Japanese terms and 84,554 English terms. It took an SGI computer with four MIPS R10000 2.5 processors and 2 gigabytes of RAM approximately 9.5 hours to compute 1,400 SVD dimensions (4% of the full dimensionality and 85% of the total variance) from the matrix of Japanese documents, and 13 hours to compute 1,200 dimensions (3% of the full dimensionality and 54% of the total variance) from the matrix of the English collection.

For local LSI, we used top-50 documents returned from initial VSM retrieval in monolingual runs, and set the feedback size to 100 in cross-language runs.

3.1 NTCIR-1

The results of monolingual and cross-language LSI, ADE, and local LSI retrieval are shown in Table 1.

The monolingual results are indicative of what each method is capable of: while LSI is not as effective as VSM when the collection size is large due to the limited dimension, ADE closes the gap between them with the same number of computed dimensions; the performance improvement of local LSI over VSM was not all that great, but this is consistent with the findings of Sakai et al. [11] on using local feedback on the same collection.

In cross-language retrieval, it is clear that local LSI appears to greatly outperform other two methods. We were also very interested in the results of ADE in comparison with other published results that used language specific tools such as a bilingual dictionary or a machine translation system. Here, we take a closer look. Oard and Wang [9] from the University of Maryland used the freely available "edict" Japanese/English dictionary to automatically translate the queries and obtained an average precision of 0.1534 (compared to ADE's score of 0.1703) for the 39 test queries with the same extracted topic fields as ours. Results by Sakai et al. [11] at Toshiba R&D Center—"Toshiba" in Table 1—are 0.2910 in 11-point average precision for the 21 training queries and 0.1820 for the 39 test ones with an automatic machine translation system. While their scores are higher than ours (0.1734 in 11-point average precision) for the training queries, their test query results are actually lower than the 11-point

			topic0001–0030		topic0031–0083	
			rel1	rel2	rel1	rel2
MLIR	VSM	AvgP	0.3452	0.3295	0.2601	0.2870
	LSI	AvgP	0.2555	0.2569	0.2413	0.2702
	ADE	AvgP	0.2807	0.2795	0.2576	0.2874
	Local LSI	AvgP	0.3530	0.3315	0.2643	0.2923
CLIR	LSI	AvgP	0.1327	0.1375	0.1162	0.1322
	ADE	AvgP	0.1554	0.1606	0.1703	0.1898
		11pt AP	0.1734	0.1783	0.1870	0.2043
	Local LSI	AvgP	0.2620	0.2581	0.2383	0.2568
CLIR Published	UMD	AvgP	–	–	0.1534	–
	Toshiba	11pt AP	0.2910	–	0.1820	–
	ULIS	11pt AP	–	0.1930	–	–
	Berkeley	AvgP	–	–	–	0.1925

Table 1. Monolingual and cross-language results of VSM, LSI, ADE, and local LSI on the NTCIR-1 collection

average precision of 0.1870 that we obtained from ADE. This may be partially due to the fact that they tuned the dictionary of their MT system for the training queries by adding new phrases into it. Fujii and Tetsuya [4] at the University of Library and Information Science in Japan used a compound word translation method with a bilingual dictionary on NTCIR-1. They achieved an 11-point average precision of 0.1930 with the first 21 queries and "rel2" judgments (shown in the row labeled "ULIS" in Table 1). Our result of 0.1783 is obtained with similar settings except for the topic fields being used for query—they used only the description. Finally, one of the best sets of results reported in the First NTCIR Workshop was by Chen et al. [2]. With non-interpolated average precisions as high as 0.3755 for the 39 test queries, they achieved their results with a bilingual lexicon that was built from aligning and matching Japanese and English keyword fields in NACSIS documents. Their pure MT-based run had an average precision of 0.1925 with rel2 judgments on the 39 test topics (shown in the row labeled "Berkeley" in Table 1); our result at 0.1898 is close with similar fields selected from the topics and the documents for indexing.

In summary, all the published results discussed here were obtained with the incorporation of certain type of language-specific knowledge; for example, the ability to perform word segmentation on Japanese queries is needed for the machine translation system to work. On the contrary, we accomplished comparable results with ADE merely by unigram indexing; this clearly demonstrates the ability of ADE, or if possible, a high-dimension LSI, to derive term associations and translations from bilingual corpora.

	rel1	rel2
J to E	0.0724	0.0686
E to J	0.0829	0.0773

Table 2. Cross-language results of ADE at NTCIR-2.

3.2 NTCIR-2

For NTCIR-2, we participated in the cross-lingual IR tasks E-J and J-E. The retrieval collection consists of four collections—ntc2-e0g, ntc2-e0k, ntc2-j0g, ntc2-j0k—in addition to ntc1-j1 and ntc1-e1. Due to time constraints, we did not extract parallel documents from the new collections for LSI and ADE training; we still used those 45,372 parallel ones as described above. The documents were tokenized in the same way as in NTCIR-1, and we used the description field of the topics to form the queries.

Our results in NTCIR-2 are not as good as we had hoped, as shown in Table 2. This could be because of the smaller number of fields used in queries, a mismatch between the parallel documents derived from the NTCIR-1 collection and the NTCIR-2 corpus, or some other possibility we have not yet been able to identify.

4 Conclusion

Through NTCIR, we have gained valuable experience in cross-language retrieval between English and Japanese. We were encouraged by the performance of ADE and local LSI in NTCIR-1 and disappointed in ADE's performance in NTCIR-2. A more systematic comparison, including testing methods other than ADE, on the NTCIR-2 collection is warranted.

References

[1] C. Buckley, A. Singhal, and M. Mitra. New retrieval approaches using SMART: TREC 4. In D. K. Harman, editor, *Proceedings of Fourth Text Retrieval Conference (TREC-4)*, pages 25–43. Department of Commerce, National Institute of Standards and Technology, 1996.

[2] A. Chen, F. C. Gey, K. Kishida, H. Jiang, and Q. Liang. Comparing multiple methods for Japanese and Japanese-English text retrieval. In *Proceedings of the First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition*, pages 49–58. National Center for Science Information Systems, Tokyo, Japan, 1999.

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

[4] A. Fujii and T. Ishikawa. Cross-language information retrieval for technical documents. In *Proceedings of the Joint ACL SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-99)*, pages 29–37. Association for Computational Linguistics, 1999.

[5] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, Maryland, 2nd edition, 1989.

[6] D. Hull. Improving text retrieval for the routing problem using latent semantic indexing. In *Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Routing, pages 282–291, 1994.

[7] F. Jiang and M. L. Littman. Approximate dimension equalization in vector-based information retrieval. In P. Langley, editor, *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 423–430, San Francisco, 2000. Morgan Kaufmann.

[8] T. K. Landauer and M. L. Littman. Fully automatic cross-language document retrieval using latent semantic indexing. In *Proceedings of the Sixth Annual Conference of the UW Centre for the New Oxford English Dictionary and Text Research*, pages 31–38, Waterloo Ontario, October 1990.

[9] D. W. Oard and J. Wang. NTCIR CLIR experiments at the University of Maryland. In *Proceedings of the First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition*, pages 157–162. National Center for Science Information Systems, Tokyo, Japan, 1999.

[10] J. J. Rocchio. Relevance feedback in information retreival. In G. Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Retrieval*, pages 313–323. Prentice Hall, Englewood Cliffs, New Jersey, 1971.

[11] T. Sakai, Y. Shibazaki, M. Suzuki, M. Kajiura, T. Manabe, and K. Sumita. Cross-language information retrieval for NTCIR at Toshiba. In *Proceedings of the First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition*, pages 137–144. National Center for Science Information Systems, Tokyo, Japan, 1999.

[12] G. Salton, A. Wang, and C. S. Yang. A vector space model for information retrieval. *Journal of the American Society for Information Science*, 18(11):613–620, November 1975.

[13] P. Sheridan and J. P. Ballerini. Experiments in multilingual information retrieval using the SPIDER system. In H.-P. Frei, D. Harman, P. Schäuble, and R. Wilkinson, editors, *Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 58–65, New York, August 1996. Association for Computing Machinery.

[14] S. K. M. Wong, W. Ziarko, and P. C. N. Wong. Generalized vector space model in information retrieval. In *Proceedings of the Eighth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 18–25. Association for Computing Machinery, 1985.

[15] J. Xu and W. B. Croft. Query expansion using local and global document analysis. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '96)*, pages 4–11. Association for Computing Machinery, 1996.

[16] Y. Yang, J. G. Carbonell, R. D. Brown, and R. E. Frederking. Translingual information retrieval: Learning from bilingual corpora. *Artificial Intelligence*, 103(1–2):323–345, 1998.