Evaluation of Stop Word Lists in Text Retrieval Using Latent Semantic Indexing

A N K Zaman, Pascal Matsakis
School of Computer Science
University of Guelph
Guelph, ON, Canada
azaman@uoguelph.ca, pmatsaki@uoguelph.ca

Charles Brown
Computer Science
University of Northern British Columbia
Prince George, BC, Canada
brownc@unbc.ca

***Abstract*—The goal of this research is to evaluate the use of English stop word lists in Latent Semantic Indexing (LSI)-based Information Retrieval (IR) systems with large text datasets. Literature claims that the use of such lists improves retrieval performance. Here, three different lists are compared: two were compiled by IR groups at the University of Glasgow and the University of Tennessee, and one is our own list developed at the University of Northern British Columbia. We also examine the case where stop words are not removed from the input dataset. Our research finds that using tailored stop word lists improves retrieval performance. On the other hand, using arbitrary (non-tailored) lists or not using any list reduces the retrieval performance of LSI-based IR systems with large text datasets.**

Keywords—Information Retrieval; Latent Semantic Indexing; stop words; recall-precision.

I. INTRODUCTION

Latent Semantic Indexing (LSI) was first proposed by Deerwester et al. [1]. LSI-based text retrieval systems consider that some words are not influential during the execution of the semantic analysis process. These words, which are called stop words, include articles (e.g., a, an, the), prepositions (e.g., at, by, in, to, from, with) and conjunctions (e.g., and, but, as, because). Proper stop word identification and removal are a central problem for many text processing applications in different domains. Stop words of a given text dataset might not be stop words of other datasets [2]. Stop words have significant impact on the text retrieval processes in different languages. In [3], Dolamic et al. evaluated two stop word lists with lengths 571 and 9 respectively on the Cross-Language Evaluation Forum (CLEF) dataset. According to the authors, the Divergence from Randomness (DFR) model shows lower retrieval performance when a short or no stop word list is removed from the input dataset. However, in case of a revised Okapi (Information Retrieval—or IR—system) implementation, retrieval performance does not show any significant difference whether a short, long or no stop word list is removed from the dataset. The authors also draw the same conclusion for other natural languages such as French, Hindi, and Persian. The main weaknesses of their work are that they use arbitrary stop word lists and do not use the *tf-idf* (term frequency-inverse document frequency) weighting scheme [4]. Zou et al. [5] show that the removal of stop words from Chinese text is important for Chinese word segmentation and improves the performance of Chinese text retrieval. The removal of stop words also improves systems' performance in Arabic IR and Arabic text summarization [6] [7]. The removal of stop words has positive impact on English text categorization [8]. Stop word removal also improves retrieval performance in case of cross-language IR. A number of cross-language based IR systems are reported in literature, e.g., Bengali-Hindi, Turkish-English, Japanese-English [9] [10] [11]. Schuemie et al. [12] removed stop words for cross-language IR for biomedical literature. In the end, the removal of stop words plays an important role in different text processing domains in different languages.

Although the Text REtrieval Conference (TREC) encourages text retrieval research, it does not provide any evidence or rule to use stop words in IR research. Different IR research groups use different stop word lists, and the size of these lists vary. As there is no standard stop word list for English text, one open question is the following: What are the effects of tailored (based on a certain dataset) vs. arbitrary (not tailored) stop word lists on LSI-based text retrieval systems with large datasets? In this study, the input dataset is the TREC-8 LA Times dataset. Two existing, arbitrary stop word lists are considered, as well as our own tailored stop word list. The retrieval results when removing, or not removing, the three different sets of stop words from the input dataset are compared.

The rest of the paper is organized as follows. Section II gives an overview of the LSI technique. Section III presents some characteristics of the input dataset and describes the way our stop word list is compiled. The experimental setup and retrieval results are presented in Section IV. Concluding remarks are in Section V.

II. LATENT SEMANTIC INDEXING

A. Overview

LSI is a method that exploits the idea of vector space model and Singular Value Decomposition (SVD). SVD is an effective dimensional reduction scheme. It has been proved to be a very good choice for uncovering latent semantic structure [1]. SVD can be applied with an arbitrary rectangle matrix with the entries on the rows and columns. The matrix is then decomposed into three matrices containing singular vectors and/or singular values. These three matrices with special forms

978-1-4577-1539-6/11/$26.00 ©2011 IEEE

show a breakdown of the original matrix into linearly independent components or factors. Many of these components are very small, leading to an approximate model that contains many fewer dimensions. Thus, for IR purposes, SVD provides a reduced model for representing the *term-to-term*, *document-to-document* and *term-to-document* relationships. By dimension reduction, it is possible for documents with somewhat different profiles of term usage to be mapped into the same vector of factor values. This property helps to eliminate the noise in the original data, thus improving the reliability of the algorithm. Suppose we obtained a $t \times d$ term-by-document matrix M from the collection indexing process of the traditional vector space method. We can apply SVD on M, which is then decomposed into three special matrices U, S and V. The decomposition can be written as:

$$M = USV^T \tag{1}$$

U is the $t \times t$ orthogonal matrix ($UU^T=I_t$) having the left singular vectors of M as its columns; V is the $d \times d$ orthogonal matrix ($VV^T=I_d$) having the right singular vectors as its columns; S is the $t \times d$ diagonal matrix having the singular values $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{\min(t,d)}$ of M in order along its diagonal. It should be noted that such a factorization exists for any arbitrary matrix [13].

Generally, in (1), the matrices U, S and V must all be of full rank. However, SVD offers a simple strategy for optimal approximation to fit using smaller matrices [1]. If the singular values in S are ordered by size, the first k largest values may be kept and the remaining smaller ones set to zero. The product of the resulting matrices is a matrix M_k which is only approximately equal to M, and is of rank k. Since zeros were introduced into S, the representation can be simplified by deleting the zero rows and columns of S to obtain a new diagonal matrix S_k, and then deleting the corresponding columns of U and V to obtain U_k and V_k respectively. The rank-k model with the best possible least-squares-fit to M can be written as follows:

$$M_k = U_k S_k V_k^T \tag{2}$$

where M_k is a matrix of size $t \times d$, U_k is of size $t \times k$, S_k is of size $k \times k$, and V_k is of size $k \times d$.

SVD provides an optimal solution to dimensionality reduction in that it derives an orthonormal space, where the dimensions are ordered. Therefore, projecting the set of documents onto the k lowest dimensions is guaranteed to have, among all possible projections to a k dimensional space, the lowest possible least-square distance to the original documents.

B. Weighting Scheme

One significant issue in LSI-based IR systems is term weighting, i.e., assigning weight to a term so that the assigned weight properly reflects the contribution of the term in distinguishing the considered document from other documents. Let L_{ij} be the local weight of the term i in the document j and tf_{ij} be the frequency with which the term i appears in the document j. The local weight in terms of raw term frequency is defined as follows:

$$\textit{raw term frequency:}\quad L_{ij} = tf_{ij} \tag{3a}$$

Let G_i be the global weight of the term i, let tf_i be the frequency of the term i in the entire collection, let df_i be the frequency of documents in which i occurs, and let d be the number of documents in the whole collection. The following equations define the *idf* and *tf-idf* weighting schemes:

$$\textit{idf:}\quad G_i = \log\left(\frac{d}{df_i}\right) \tag{3b}$$

$$\textit{tf-idf:}\quad G_i = tf_{ij} \times idf_i \tag{3c}$$

III. INPUT DATASET AND STOP WORD LISTS

Examples of stop word lists are the lists compiled by the IR groups at the University of Glasgow and the University of Tennessee. These lists have 319 and 439 stop words, respectively. In their book [14], Manning et al. describe a way to prepare a list of stop words: "*the general strategy for determining a stop list is to sort the terms by collection frequency (the total number of times each term appears in the document collection), and then to take the most frequent terms, often hand-filtered for their semantic content relative to the domain of the documents being indexed, as a stop list, the members of which are then discarded during indexing.*" Our own stop word list has been compiled following the above idea. It includes the University of Glasgow and University of Tennessee stop word lists, 730 TREC file names (input dataset), 22 tag names (e.g., doc, docno, etc) and other words (e.g., alphanumeric words, roman numbers). Its total length is 1911. The algorithmic steps to create this stop word list are given below:

- Consider the terms whose frequency is at least 2 (a term must be present in the document at least twice).
- Create an initial stop word list by combining the stop word lists of the IR University of Tennessee and University of Glasgow groups (without duplication of terms in the list).
- Remove all the punctuation from the input TREC-8 LA Times dataset.
- Create a list of terms from the input dataset, in descending order of term frequencies, i.e., the term with the highest term frequency will be at the top of the list.
- Manually extract the special items to be added to the initial list (those terms are not already in the initial list) to create an extended stop word list.
- Add all file names to the initial list as every file contains file names, e.g., LA123190.
- Add all tag names, e.g., doc, docno, to the initial stop word list.
- Add roman numbers to the initial list, e.g., xvii.

- Add scale units, e.g., ft, mm, etc.
- Add adjectives and adverbs, e.g., ago.
- Add prefixes from words, e.g., non (as in non-governmental).
- Add special words, e.g., haven (as in haven't), doesn (as in doesn't).
- Add dates, e.g., feb-92, 11-apr.
- Add foreign words as dataset in newspaper articles. Add suspicious words, e.g., aaftink, aachen, ora.
- Add other words, e.g., ext (telephone extension), 19th, z90, v6 (engine).

To compile this stop word list, first we manually search the high frequency, low frequency, and other special terms out of 132,785 terms in the frequency table. We repeat this in a number of cycles by removing different stop words from the TREC-8 LA Times dataset. Searching stop words is very time consuming as the dataset as well as the number of terms are large. The most difficult thing is to choose a word as a stop word. Since the TREC-8 LA Times dataset contains newspaper articles (on politics, sports, geography, history, science-technology, etc.), there are variations in the contents. Some characteristics of this dataset are presented in Table I.

IV. EXPERIMENT

This study finds out the effects of "stop words / common words" on a text-based LSI IR process for the TREC-8 LA Times dataset. Evidence is developed to indicate the most effective stop word lists for LSI-based ad hoc IR processes. We performed our experiment by removing the stop word lists mentioned in Table II, as well as without removing them. We applied Porter's stemming [15] to find the root words from the input text. 50 queries were used (associated with the TREC-8 LA Times dataset) to evaluate the retrieval performance. Our findings are presented in terms of recall-precision [14]. Table III shows the 10-point interpolated precision of the four different retrieval systems: the UNBC system which uses our stop word list developed at the University of Northern British Columbia; the UT system which uses the University of Tennessee stop word list; the UG system which uses the University of Glasgow stop word list; and the STEM system which does not use any stop word list. The recall-precision graph based on the results in Table III is shown in Fig. 1.

TABLE I. CHARACTERISTICS OF THE TREC-8 LA TIMES DATASET (1989,1990)

Number of documents	131,321
Size of the input dataset	476MB
Average vocabulary size (approximately)	500
Average document size (approximately)	40 KB
Largest file size	828 KB (LA052089_0101)
Smallest size	352 Bytes (LA070189_0001)
Number of words in the smallest file	91
Number of words in the largest file	167,045
Number of relevant files (out of 131,312 files) with respect to TREC-8 query set	1,151

TABLE II. PARAMETERS FOR THE STUDY OF STOP WORDS

Dataset	Stop Word Lists	Stemming	Weighting Scheme	Number of Queries
TREC-8 LA Times	University of Glasgow	Porter's Stemming	tf-idf	50
	University of Tennessee			
	University of Northern British Columbia			
	( )			

TABLE III. 10–POINT INTERPOLATED PRECISION OF THE FOUR SYSTEMS

10-Point Recall	UNBC	UT	UG	STEM
0.1	0.1757	0.1513	0.1221	0.1385
0.2	0.1108	0.0994	0.0864	0.1058
0.3	0.0888	0.0735	0.0799	0.0841
0.4	0.0724	0.0693	0.0722	0.0743
0.5	0.0678	0.0672	0.0694	0.0710
0.6	0.0659	0.0647	0.0671	0.0662
0.7	0.0646	0.0609	0.0660	0.0632
0.8	0.0621	0.0588	0.0628	0.0592
0.9	0.0554	0.0489	0.0578	0.0553
1.0	0.0436	0.0316	0.0374	0.0247

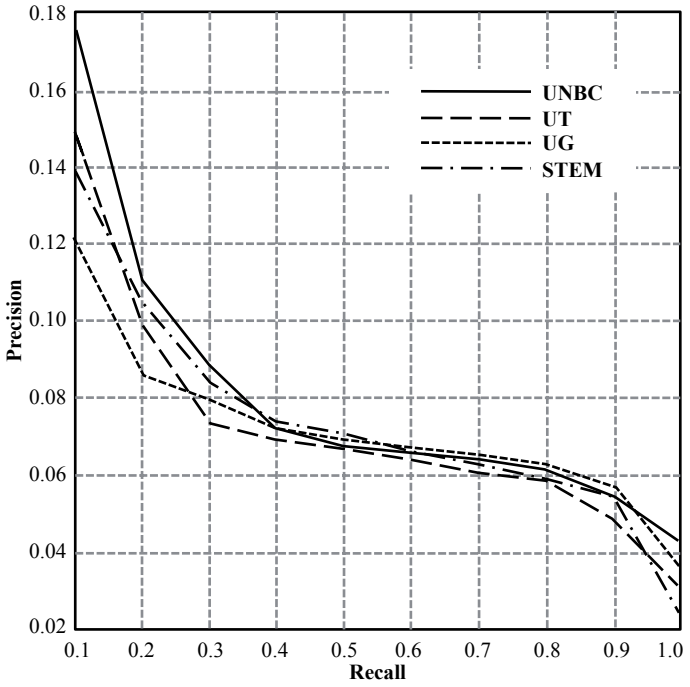

Figure 1. Recall-Precision Graph

In Table III, the recall value of 0.1 represents the top 10% of the retrieved documents (in the collection) which are relevant to a query set. As an example, using the UNBC system, the precision associated with the top 10% of the documents is 0.1757 (i.e., 17.57%). This value is calculated by interpolating the precision values of all 50 queries used for this research at the standard recall value 0.1.

The retrieval systems are compared in terms of precision in different standard recall points, e.g., 0.1, 0.2. For example, at recall point 0.3 (top ranked 30% documents), the precision values for UNBC is 8.88%, and it is 7.35% for UT. So, UNBC shows 1.53% (8.88%-7.35%) better retrieval performance than UT for the top 30% retrieved documents. If we look at the recall point 0.3 in Fig. 2, we can see the differences.

In the end, the system UNBC with extended stop word list provides the best result when compared to the three other systems. For the top 10% retrieval, it shows 5.37% better retrieval performance than UG, 3.68% better retrieval performance than STEM, and 2.44% better retrieval performance than UT. However, after top 40% retrieval all the systems show almost the same retrieval performance. Note that in STEM, we just applied Porter's stemming without removing stop words, and the retrieval performance is 1.64% better than UG's. From the above results, it is clear that the use of an arbitrary set of stop words reduces retrieval performance in case of LSI-based ah hoc IR with large dataset.

V. CONCLUSION

Stop word lists are one of the key parameters in the area of IR. We have investigated the performance of LSI by using three different stop word lists, and also, without using any stop word list, i.e., without removing stop words from the input dataset. Our main finding is that for a LSI-based ad hoc IR system, the use of an arbitrary stop word list reduces retrieval performance: for better retrieval performance, a tailored stop word list must be assembled for every unique large dataset.

REFERENCES

[1] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. A.; "Indexing by latent semantic analysis". Journal of the American Society for Information Science, 41(6), pp.391-407, 1990.

[2] Dragut E.C., Fang F., Sistla A.P., Yu C.T. and Meng W., "Stop word and related problems in web interface integration", The Proceedings of the Very Large Database Endowment, PVLDB 2 (1), pp. 349–360, 2009.

[3] Dolamic, L. and Savoy, J., When stopword lists make the difference. Journal of the American Society for Information Science and Technology, 61: pp. 200–203, 2010.

[4] Zaman, A.N.K.; Brown, C.G.; , "Latent semantic indexing and large dataset: Study of term-weighting schemes," Digital Information Management (ICDIM), 2010 Fifth International Conference on, pp.1-4, 5-8 July 2010

[5] Zou F., Wang F. L., Deng X, and Han S., "Evaluation of Stop Word List in Chinese Language", the 5th edition of the International Conference on Language Resources and Evaluation (LREC), Genoa, Italy, 2006.

[6] I. El-Khiar, "Effects of Stops Words Elimination for Arabic Information Retrieval: A Comparative Study." In International Journal of Computing and Information Sciences, 4 (3), 2006.

[7] Azmi, A.; Al-thanyyan, S.; , "Ikhtasir — A user selected compression ratio Arabic text summarization system," Natural Language Processing and Knowledge Engineering, 2009. NLP-KE 2009. International Conference on , vol., no., pp.1-7, 24-27 Sept. 2009

[8] Feng Xia; Tian Jicun; Liu Zhihui; , "A Text Categorization Method Based on Local Document Frequency," Fuzzy Systems and Knowledge Discovery, 2009. FSKD '09. Sixth International Conference on , vol.7, no., pp.468-471, 14-16 Aug. 2009

[9] Mandal, D., Gupta, M., Dandapat, S., Banerjee, P., Sarkar, S.: Bengali and Hindi to English CLIR evaluation. In: Advances in Multilingual and Multimodal Information Retrieval (8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007). Number 5152 in LNCS, Budapest, Hungary Springer Verlag (2008) 95–102

[10] Celebi, E.; Sen, B.; Gunel, B.; , "Turkish — English cross language information retrieval using LSI," Computer and Information Sciences, 2009. ISCIS 2009. 24th International Symposium on , pp.634-638,14-16 Sept. 2009

[11] Li and Shawe-Taylor, Y. Li and J. Shawe-Taylor, "Using KCCA for Japanese–English cross-language information retrieval and document classification", Journal of Intelligent Information System 27(2), pp. 117–133, 2006.

[12] Schuemie M., Trieschnigg D., and Kraaij W.. "Cross Language Information Retrieval for Biomedical Literature". In Proceedings of The Sixteenth Text REtrieval Conference, TREC 2007. NIST, Gaithersburg, MD, USA.

[13] Schutze H. and Silverstein C., "Projections for efficient document clustering". In ACM/SIGIR, pp. 74–81, 1997.

[14] Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze. An Introduction to Information Retrieval. Cambridge University Press Cambridge, 2008.

[15] Porter M. An algorithm for suffix stripping, avaliable :http://tartarus.org/~martin/PorterStemmer/ Access time: May 2011