Waterloo at NTCIR-3: Using Self-supervised Word Segmentation

Xiangji Huang Fuchun Peng Dale Schuurmans Nick Cercone
School of Computer Science, University of Waterloo
Waterloo, Ontario, Canada, N2L 3G1
{jhuang, f3peng, dale, ncercone}@cs.uwaterloo.ca

Abstract

In this paper, we describe the system we use in the NTCIR-3 CLIR (cross language IR) task. We participate the SLIR (single language IR) track. In our system, we use a self-supervised word-segmentation technique for Chinese information retrieval, which combines the advantages of traditional dictionary based approaches with character based approaches, while overcoming many of their shortcomings. This method is completely language independent and unsupervised, which provides a promising avenue for constructing accurate multi-lingual or cross-lingual information retrieval systems that are flexible and adaptive.

1 Introduction

In this paper, we propose an EM-based method for Chinese information retrieval which has many of the advantages of both the character based approach and the dictionary based approach, while overcoming many of the shortcomings of both methods. We call our approach *self-supervised segmentation*. In *self-supervised segmentation*, no predefined lexicon is required. Instead, all that is needed is a large unsegmented training corpus—which is almost always easy to obtain. We automatically learn a lexicon and lexical distribution from the training corpus by using the EM algorithm [10], and then segment the collections using the Viterbi algorithm [24]. Unlike previous EM word segmentation methods [11], where one lexicon is learnt, we learn two lexicons (for reasons outlined below). Since our segmentation approach is completely unsupervised and language independent, it can be easily adapted to other languages.

The rest of the paper is organized as follows. Section 2 first introduces the self-supervised word segmentation algorithm, and Section 3 then briefly describes the weighting methods we use in the experiments. Section 4 then presents the experiments we have conducted on the NTCIR data set. Conclusions are given in Section 5.

2 Self-supervised Segmentation

In a general word segmentation task where there are no identifying markers between words, one could effectively exploit *known* words to guide the segmentation of unknown words. For example, if the word "*computer*" is already known then upon seeing the text "*computerscience*" it is natural to segment "*science*" as a possible new word. To exploit this observation, we develop an EM based word discovery method that is a variant of standard EM training, but avoids getting trapped in local maxima by keeping two lexicons: a *core* lexicon which contains words that are judged to be trustworthy, and a *candidate* lexicon which contains all other candidate words that are not in the core lexicon.

Assume we have a sequence of characters $C = c_1c_2...c_T$ that we wish to segment into chunks $S = s_1s_2...s_M$, where T is the number of characters in the sequence and M is the number of words in the segmentation. Here chunks s_i will be chosen from the core lex-

icon $V_1 = \{s_i, i = 1, ..., |V_1|\}$ or the candidate lexicon $V_2 = \{s_j, j = 1, ..., |V_2|\}$. If we already have the probability distributions $\theta = \{\theta_i | \theta_i = p(s_i), i = 1, ..., |V_1|\}$ defined over the core lexicon and $\phi = \{\phi_j | \phi_j = p(s_j), j = 1, ..., |V_2|\}$ over the candidate lexicon, then we can recover the most likely segmentation of the sequence $C = c_1 c_2 ... c_T$ into chunks $S = s_1 s_2 ... s_M$ as follows. First, for any given segmentation S of C, we can calculate the joint likelihood of S and C by

$$prob(S, C|\theta, \phi) = \prod_{i=1}^{M_1} \frac{p(s_i)}{2} \prod_{j=1}^{M_2} \frac{p(s_j)}{2} = \frac{1}{2^M} \prod_{k=1}^{M} p(s_k)$$

where M_1 is the number of chunks occurring in the core lexicon, M_2 is the number of chunks occurring in the candidate lexicon, and s_k can come from either lexicon. (Note that each chunk s_k must come from exactly one of the core or candidate lexicons.) Our task is to find the segmentation S^* that achieves the maximum likelihood:

$$S^* = \underset{S}{argmax}\{prob(S, C|\theta, \phi)\} \qquad (1)$$

Given a probability distribution defined by θ and ϕ over the lexicon, the Viterbi algorithm can be used to efficiently compute the best segmentation S of character string C. Estimation of the probabilities can be done by the EM algorithm. The parameter re-estimation formulas are as follows.

$$\theta_i^{k+1} = \frac{\sum_S \#(s_i, S) \times prob(S, C|\theta^k, \phi^k)}{\sum_{s_i} \sum_S \#(s_i, S) \times prob(S, C|\theta^k, \phi^k)} \qquad (2)$$

$$\phi_j^{k+1} = \frac{\sum_S \#(s_j, S) \times prob(S, C|\theta^k, \phi^k)}{\sum_{s_j} \sum_S \#(s_j, S) \times prob(S, C|\theta^k, \phi^k)} \qquad (3)$$

where $\#(s_i, S)$ is the number of times s_i occurring the segmentation S.

The two lexicons are constructed automatically as follows. Let us define C_1, C_2 as the training corpus and the validation corpus respectively, and let V_1 and $V2$ be the core candidate lexicons respectively. Initially, V_1 is set to be empty and V_2 is initialized to contain all candidate "words" that are generated from the training corpus by enumerating contiguous character strings of lengths 1 to L for some predefined maximum length L. In a first pass, starting from the uniform distribution, EM is used to increase the likelihood of the training corpus C_1. When the training process stabilizes, the M words with highest probability are selected from V_2 and moved to V_1, after which all the probabilities are rescaled so that V_1 and V_2 each contain half the total probability mass. EM is then run again. The rationale for shifting half of the probability mass to V_1 is that this increases the influence of core words in determining segmentations and allows them to act as more effective guides in processing the training sequence. We call this procedure of successively moving the top M words to V_1 *forward selection*. Forward selection is repeated until the segmentation performance of Viterbi on the validation corpus C_2 leads to a decrease in F-measure (which means we must have included some erroneous words in the core lexicon). After forward selection terminates, M is decremented and we carry out a process of *backward deletion*, where the M words with the lowest probability in V_1 are moved back to V_2, and EM training is successively repeated until F-measure again decreases on the validation corpus C_2 (which means we must have deleted some correct core words). The two procedures of forward selection and backward deletion are alternated, decrementing M at each alternation, until $M \leq 0$;

3 Probabilistic Term Weighting

In an attempt to ensure that the phenomena we observe are not specific to a particular retrieval technique, we experimented with a parameterized term weighting scheme which allowed us to control the quality of retrieval performance. We considered a refined term weighting scheme based on the the standard term weighting function

$$w_0 = \log \frac{N - n + 0.5}{n + 0.5} \qquad (4)$$

where N is the number of indexed documents in the collection, and n is the number of documents containing a specific term [26].

Many researchers have shown that augmenting this basic function to take into account document length, as well as within-document and within-query frequencies, can be highly beneficial in English text retrieval [2]. For example, one standard augmentation is to use

$$w_1 = w_0 * \frac{(c_1 + 1) * tf}{K + tf} * \frac{(c_2 + 1) * qtf}{c_2 + qtf} \quad (5)$$

where

$$K = c_1 * \left(1 - c_3 + c_3 \frac{dl}{avdl}\right)$$

Here tf is within-document term frequency, qtf is within-query term frequency, dl is the length of the document, $avdl$ is the average document length, and c_1, c_2, c_3 are tuning constants that depend on the database, the nature of the queries, and are empirically determined. However, to truly achieve state-of-the-art retrieval performance, and also to allow for the quality of retrieval to be manipulated, we further augmented this standard term weighting scheme with an extra correction term

$$w_2 = w_1 \ \oplus \ k_d * y \quad (6)$$

This correction allows us to more accurately account for the length of the document. Here $\oplus$ indicates that the component is added only once per document, rather than for each term, and

$$y = \begin{cases} \ln(\frac{dl}{avdl}) + \ln(c_4) & \text{if } dl \leq rel\_avdl \\ \left(\ln(\frac{rel\_avdl}{avdl}) + \ln(c_4)\right)\left(1 - \frac{dl - rel\_avdl}{c_5 * avdl - rel\_avdl}\right) & \text{if } dl > rel\_avdl \end{cases}$$

where rel_avdl is the average relevant document length calculated from previous queries based on the same collection of documents. Overall, this term weighting formula has five tuning constants, c_1 to c_5, which are all set from previous research on English text retrieval and some initial experiments on Chinese text retrieval. In our experiments, the values of the five arbitrary constants c_1, c_2, c_3, c_4 and c_5 were set to 2.0, 5.0, 0.75, 3 and 26 respectively.

The key constant is the quantity k_d, which is the new tuning constant that we manipulate to control the influence of correction factor, and hence control the retrieval quality. By setting k_d to different values, we have different term weighting methods in our experiments. In our experiments, we tested k_d set to values of 0, 6, 8, 10, 15, 20, 50.

4 Experiments and Analyses

4.1 Experiment setup and data sets

The document collection used in NTCIR03 include CIRB011 and CIRB020. CIRB011 documents come from Central Daily News, China Daily News Chinatimes Commercial, Chinatimes Express and China Times. The statistics of CIRB011 are listed in Table 1.

News Agency	# of Document	Per.
Chinatimes	38,163	28.8%
Chinatimes Commercial	25,812	19.5%
Chinatimes Express	5,747	4.4%
Central Daily News	27,770	21.0%
China Daily News	34,728	26.3%
Total	132,173	200MB

Table 1: statistics of CIRB011

CIRB020 are news articles published by United Daily News (udn.com) from 1998-01-01 to 1999-12-31. CIRB020 contains 249,508 news articles in total. In total, the NTCIR Chinese collection consists of 381,681 documents and 50 topics [1] encoded using BIG5 coding scheme and marked with XML. The minimum, maximum and average document sizes are 0, 29,540 and 1,223 bytes. Each topic has several fields: T (Topic) field, D (Desc) filed, N (Narr) filed and C (Conc) field. Keywords can be extracted from the combination of these fields.

The NTCIR relevance judgments for each topic came from the human assessors The relevance judgments will be done in four grades, highly relevant, relevant, partially relevant and irrelevant. Several measures are used to evaluate the retrieval result which is an ordered set of retrieved documents. The measures include *Average Precision*: average precision over all 11 recall points (0.0, 0.1, 0.2,...,

[1] Actually, there are only 42 valid topics

1.0); *R-Precision*: precision after the number of documents retrieved is equal to the number of known relevant documents for a query; and Precision at # docs: precision after # documents have been retrieved. Detailed descriptions of these measures can be found in [28].

Our segmenter is trained on the training set C_1 with validation set C_2 (see section 2), where C_1 is 90M data which contains one year of *People's Daily* news service stories (www.snweb.com) and corpus C_2 used here is a randomly selected 2000 sentence subset of the Chinese Treebank from LDCwhich is has been segmented by hand. Our segmentation accuracy is around 70-74% on the Chinese Treebank.

4.2 Experimental Results

We submitted 4 runs: WATERLOO-C-C-TDNC-01 (Run 1), WATERLOO-C-C-C-02 (Run 2), WATERLOO-C-C-D-03 (Run 3) and WATERLOO-C-C-C-04 (Run 4). However, due to some reason, our WATERLOO-C-C-C-02 did not have evaluation results, so we are not discussing it here. The name format of the runs follows the NTCIR3 dry run submission instruction.

4.2.1 WATERLOO-C-C-TDNC-01

In this run, the query keywords are extracted from the combination of fields title (T), desc (D), narr (N) and conc (C). In Table 2, we show the results of this run and also include the overall statistics of NTCIR3: maximum value (Max.), minimum value (Min.) and average value (Avg.).

In summary, the average precision is a little below the average, and P-Precision and the Precisions at a specific number are above the averages.

4.2.2 WATERLOO-C-C-D-03

In this run, the query key words are extracted from the *desc* (D) field. The evaluation results are given in Table 3. The overall statistics are not available.

Rigid Evaluation				
	Max.	Min.	Avg.	WATERLOO
Avg. Pre.	0.3435	0.0347	0.2263	0.2251
R-Pre.	0.3463	0.0327	0.2493	0.2499
PreAt10	0.4405	0.0405	0.3130	0.3190
PreAt100	0.2017	0.0162	0.1452	0.1526
PreAt1000	0.0408	0.0050	0.0317	0.0335
Relax Evaluation				
	Max.	Min.	Avg.	WATERLOO
Avg. Pre.	0.4165	0.0443	0.2806	0.2763
R-Pre.	0.4330	0.0526	0.3134	0.3262
PreAt10	0.5833	0.0619	0.4298	0.4381
PreAt100	0.2867	0.0293	0.2142	0.2248
PreAt1000	0.0660	0.0093	0.0513	0.0541

Table 2: Results of WATERLOO-C-C-TDNC-01

Rigid Evaluation	
Avg. Pre.	0.1764
R-Pre.	0.2118
PreAt10	0.2976
PreAt100	0.1198
PreAt1000	0.0257
Relax Evaluation	
Avg. Pre.	0.2138
R-Pre.	0.2631
PreAt10	0.3952
PreAt100	0.1800
PreAt1000	0.0423

Table 3: Results of WATERLOO-C-C-C-03

4.2.3 WATERLOO-C-C-C-04

In this run, the query keywords are extracted only from *conc* (C) field. The evaluation results are given in Table 4. The overall statistics are not available.

4.2.4 Comparison

Figures 1 and 2 show the comparison results in terms of Precision at 10, 100 and 1,000 for the Run 1, 2 and 3 with respect to the Relax and Rigid evaluations respectively. Each figure contains three groups. In each group, the first and second bars represent the results from Run 1 and Run 2. The third bar repre-

Rigid Evaluation	
Avg. Pre.	0.1831
R-Pre.	0.2057
PreAt10	0.2595
PreAt100	0.1295
PreAt1000	0.0325
Relax Evaluation	
Avg. Pre.	0.2406
R-Pre.	0.2727
PreAt10	0.3786
PreAt100	0.1967
PreAt1000	0.0530

Table 4: Results of WATERLOO-C-C-C-04

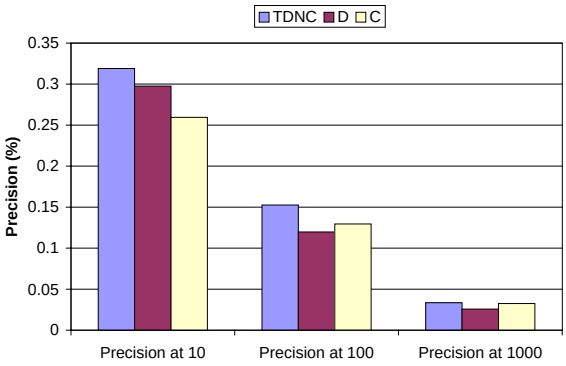

Figure 2: Precision at 10, 100 and 1000 for Rigid Evaluation

sents the results from Run 3.

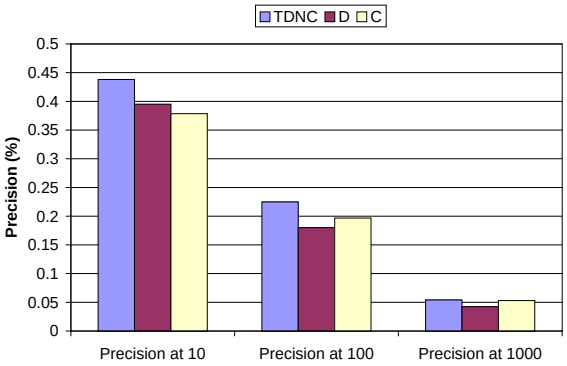

Figure 1: Precision at 10, 100 and 1000 for Relax Evaluation

4.3 Discussion

Our work is most related to the work of Chen [7]. There too it was proposed that Chinese IR could be conducted without using a dictionary. In their method, one first collects occurrence frequencies for uni-grams and bi-grams from the corpus, and then uses a mutual information based criterion to segment Chinese text [27]. To use mutual information, they limit the word length to at most 2 characters. Similarly, we also use the frequencies from the corpus and also use mutual information during the process. However, our work differs from theirs in many respects [20, 21]. First we do not limit the word length to 2 characters. The maximum word length could be set arbitrarily to suit the application. In fact, our best results are achieved when $L = 3$. Second, the statistics we used were optimized by an iterative EM process, which is guaranteed to achieve at least a local optimum. This approach should be more reliable than the statistics direct from the corpus. Our further experiments on TREC data sets shows our EM approach outperforms the mutual information based approach under the same experimental environment setting (results are not shown here).

Using EM for word segmentation has many advantages, and has in fact been considered in previous research [11, 22]. However, due to the low segmentation accuracies these methods obtain, they still do not tend to be regarded as good methods for Chinese IR. However, we have shown that the accuracies obtained by EM word segmentation is enough to achieve good results in Chinese IR systems [20].

However, our overall results are only around the average. It is much worse than the best results where query expansion and interactive retrieval methods are used. In our retrieval system, we did not use any query expansion and our keywords extraction is simple and completely automatically. Because our previous experiments [21] show that EM based word segmentation can achieve comparable results to other segmentation methods, we conjecture the the reason why our perfor-

mance is not good here is due to the query processing.

Figure 3 compares TDNC, C and D runs in terms of average precision and R-precision with respect to Relax evaluation. Figure 4 compares TDNC, C and D runs in terms of average precision and R-precision with respect to Rigid evaluation. We can observe that the results from the TDNC run are better than the results from the C run or the D rune only. This means the information provided by T (Topic) and N (Narr) fields can provide positive contribution to retrieval performance.

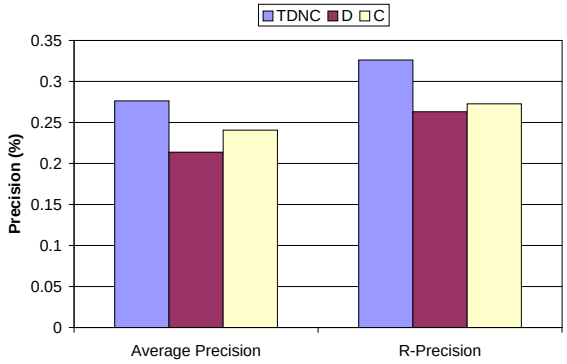

Figure 3: Average Precision and R-Precision for Relax Evaluation

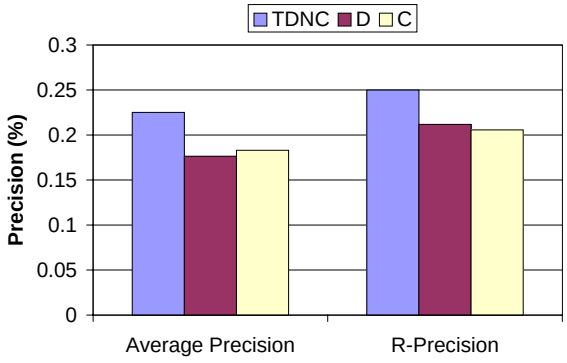

Figure 4: Average Precision and R-Precision for Rigid Evaluation

Another thing worth pointing out is that our results are far worse than those we got on the TREC data sets [21]. Further investigation needs to be conducted.

5 Conclusions and Future work

We have proposed a novel EM based method for segmenting Chinese words for the purposes of Chinese information retrieval, and presented experimental results on recent NTCIR data. Our method has the advantages of both the character based and dictionary based methods, while overcoming many of their shortcomings. However, our overall results on the NTCIR data set are not good. We are analyzing our experimental results.

We have experimented with a single word weighting strategy, and are currently investigating alternative strategies involving word-pair weighting, which can greatly increase the performance [7, 14, 16]. Building a Chinese IR system encompasses many research problems, and the performance of such systems can be influenced by several factors. As the overall NTCIR3 results show, keyword extraction also plays an important role in IR systems. Our current keyword extraction method is very rough, and we are investigating more sophisticated extraction methods such as those used in [7, 8].

6 Acknowledgements

This Research is supported by Mathematics of Information Technology and Complex Systems, and Bell University Labs.

References

[1] R. Ando and L. Lee. Mostly-Unsupervised Statistical Segmentation of Japanese: Application to Kanji. In *Proceedings ANLP-NAACL*, 2000.

[2] M. Beaulieu, M. Gatford, X. Huang, S. Robertson, S. Walker, and P. Williams. Okapi at TREC-5. In *D.K.Harman (ed): Proceedings of TREC-5*, pages 143–166, 1997.

[3] M. Brent and X. Tao. Chinese Text Segmentation With MBDP-1: Making the Most of Training Corpora. In *Proceedings of ACL2001*, France, 2001.

[4] C. Buckley, A. Singhal, and M. Mitra. Using Query Zoning and Correlation within SMART: TREC-5. In *Proceedings of TREC-5*, pages 105–118, 1997.

[5] C. Buckley, J. Walz, M. Mitra, and C. Cardie. Using Clustering and Super-Concepts Within SMART: TREC-6. In *Proceedings of TREC-6*, pages 107–124, 1998.

[6] J.-S. Chang and K.-Y. Su. An Unsupervised Iterative Method for Chinese New Lexicon Extraction. *International Journal of Computational Linguistics & Chinese Language Processing*, 1997.

[7] A. Chen, J. He, L. Xu, F. C. Gey, and J. Meggs. Chinese Text Retrieval Without Using a Dictionary. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 42–49. ACM, 1997.

[8] L.-F. Chien, T.-I. Huang, and M.-C. Chien. Pat-tree-based Keyword Extraction for Chinese Information Retrieval. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 50–58. ACM, 1997.

[9] D. Dahan and M. Brent. On the Discovery of Novel Word-like Units from Utterances: An Artificial-language Study with Implications for Native-language Acquisition. *Journal of Experimental Psychology: General*, 128:165–185, 1999.

[10] A. Dempster, N. Laird, and D. Rubin. Maximum-likelihood from Incomplete Data via the EM algorithm. *J. Royal Statist. Soc. Ser.*, B(39), 1977.

[11] X. Ge, W. Pratt, and P. Smyth. Discovering Chinese Words from Unsegmented Text. In *Proceedings of the 22th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 271–272, 1999.

[12] T. Hofmann. Probabilistic Latent Semantic Indexing. In *Proceedings of the 22th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 1999.

[13] X. Huang and S. Roberton. Okapi Chinese Text Retrieval Experiments at TREC-6. In *Proceedings of TREC-6*, pages 137–142, 1998.

[14] X. Huang and S. Roberton. A Probabilistic Approach to Chinese Information Retrieval: Theory and Experiments. In *Proceedings of the BCS-IRSG 2000: the 22nd Annual Colloquium on Information Retrieval Research*, Cambridge, England, 2000.

[15] W. Jin. Chinese Segmentation and its Disambiguation. In *MCCS-92-227*, Computing Research Laboratory, New Mexico State University, Las Cruces, New Mexico, 1992.

[16] K. L. Kwok. Comparing Representations in Chinese Information Retrieval. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 34–41. ACM, 1997.

[17] C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, Massachusetts, 1999.

[18] J. Nie and F. Ren. Chinese information retrieval: using characters or words? *Information Processing and Management*, 35:443–462, 1999.

[19] J. Nie, X. Ren, and M. Brisebois. On Chinese text retrieval. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 225–233. ACM, 1996.

[20] F. Peng, X. Huang, D. Schuurmans, and N. Cercone. Investigating the Relationship of Word Segmentation Performance and Retrieval Performance in Chinese

IR. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING2002)*, Taipei, Taiwan, 2002.

[21] F. Peng, X. Huang, D. Schuurmans, N. Cercone, and S. Robertson. Using Self-supervised Word Segmentation in Chinese Information Retrieval. In *Proceedings of 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR2002)*, Tampere, Finland, 2002. ACM.

[22] F. Peng and D. Schuurmans. Self-supervised Chinese Word Segmentation. In *F. Hoffman et al. (Eds.): Advances in Intelligent Data Analysis, Proceedings of the Fourth International Conference (IDA-01), LNCS 2189*, pages 238–247, Cascais, Portugal, 2001. Springer-Verlag Berlin Heidelberg.

[23] J. Ponte and W. Croft. Useg: A Retargetable Word Segmentation Procedure for Information Retrieval. In *Symposium on Document Analysis and Information Retrival 96 (SDAIR)*, 1996.

[24] L. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of IEEE*, 77(2), 1989.

[25] S. E. Robertson and S. Walker. Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval. In *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 232–241, 1994.

[26] K. Sparck-Jones. Search Relevance Weighting Given Little Relevance Information. *Journal of Documentation*, 35(1), 1979.

[27] R. Sproat and C. Shih. A statistical method for finding word boundaries in chinese text. *Computer Proceedings of Chinese and Oriental Languages*, 4:336–351, 1990.

[28] E. Voorhees and D. Harman. Overview of the Sixth Text REtrieval Conference (TREC-6). In *Proceedings of the sixth Text REtrieval Conference (TREC-6)*. NIST Special Publication, 1998.

[29] R. Wilkinson. Chinese Document Retrieval at TREC-6. In *Proceedings of TREC-6*, 1998.