Table 3: iSTD performances

Method (run)	Rank 100			Specified				Maximum			
	R [%]	P [%]	F [%]	R [%]	P [%]	F [%]	rank	R [%]	P [%]	F [%]	rank
(BL-1)	73.00	73.00	73.00	81.00	65.85	72.65	123	73.00	76.04	74.49	96
dist-DTW (akbl-1)	72.00	72.00	72.00	89.00	66.92	76.39	133	95.00	65.97	77.87	144
altdist-LD (akbl-2)	67.00	67.00	67.00	87.00	65.41	74.68	133	95.00	63.33	76.00	150
dist-LD (akbl-3)	68.00	68.00	68.00	90.00	65.69	75.95	137	94.00	65.28	77.05	144

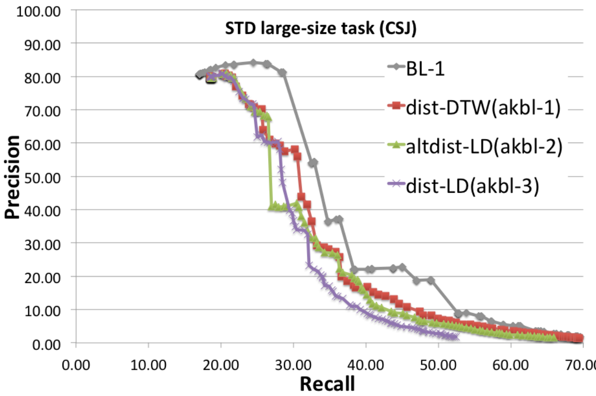

Figure 3: The recall–precision curves for the result on the large-size task (micro average)

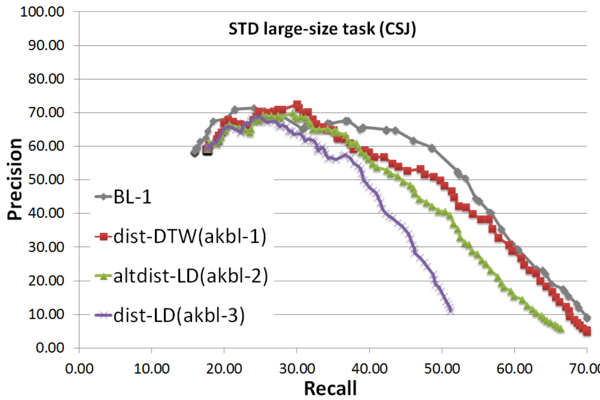

Figure 4: The recall–precision curves for the result on the large-size task (macro average)

	$v(d)$			$v_c(d)$		
	w_1	w_2	w_3	w_1w_2	w_1w_3	w_2w_3
d_1	3	0	1	0	1	0
d_2	1	2	2	1	1	1
d_3	0	0	1	0	0	0

Figure 8: *Example of document vector by extended vector space model*

more likely to appear at the same time in the document. We attempt to make use of word co-occurrence as an additional feature that is used in computing the similarity between the query and each document.

For document d, document vector $v(d)$ is computed as follow,

$$v(d) \quad = \quad [tf(w_1, d), tf(w_2, d), \dots], \tag{7}$$

where $tf(w, d)$ is frequency of word w in document d. Additionally, we also calculate co-occurrence information vector $v_c(d)$ as follows.

$$v_c(d) \quad = \quad [\delta(w_1, w_2, d), \delta(w_1, w_3, d), \dots, \delta(w_i, w_j, d), \dots] \tag{8}$$

$$\delta(w_i, w_j, d) \quad = \quad \begin{cases} 1 & (w_i \in d \text{ and } w_j \in d) \\ 0 & (\text{otherwise}) \end{cases}, \tag{9}$$

The size of the co-occurrence vector $v_c(d)$ is $|v(d)|^2 - |v(d)|$. Figure 8 shows example of document vector for query $q = w_1, w_2, w_3$.

Given query topic q, firstly, we obtain document vector $v(d)$ using word frequency that contain query topic. Than, co-occurrence vector $v_c(d)$ is computed based on $v(d)$, and extended vector $v_e(d) = [v(d), v_c(d)]$ is obtained by concatenating the $v(d)$ and $v_c(d)$. Similarly, the extended query vector $v_e(q)$ is also calculated. Finally, the similarity between $v_e(d)$ and $v_e(q)$ is calculated as same as the vector space model with TF-IDF term weighting.

3.3 Combination of CSCR and STD-SCR

The proposed method will be effective for the query including the words that are OOV or misrecognized in the spoken documents, while the conventional SCR will be effective for the query that consists of the words recognized correctly in the spoken documents. It is expected that the two systems can complement each other, so it is worth investigating the hybrid system of them.

Table 4: Evaluation results for the lecture retrieval task

Quality of trancription	run	transcription(s) word	transcription(s) syllable	retrieval model	MAP
MATCHED	baseline-1	√		SMART	0.268
	baseline-2	√		VSM	0.231
	AKBL-5		√	E-VSM	0.223
	AKBL-4		√	SMART	0.212
	AKBL-1	√	√	E-VSM	0.365
	AKBL-7	√	√	SMART	0.401
UNMATCHED	baseline-3	√		SMART	0.238
	baseline-4	√		VSM	0.225
	AKBL-6		√	E-VSM	0.223
	AKBL-3		√	SMART	0.208
	AKBL-2	√	√	E-VSM	0.341
	AKBL-8	√	√	SMART	0.367

Not only simply combining the both systems, we tried to further boost the performance by making a distinction between OOV and IV words in a given query topic. Firstly, we divide the words in a query topic q into the OOV words q_{OOV} and IV words q_{IV} by consulting the recognition dictionary of the LVCSR system used in the conventional SCR system. Then, we combine the two systems as follows.

$$\begin{aligned} sim(q, d) \quad = \quad & (1 - \alpha)\, sim_{cSCR}(q, d) \\ & + \alpha\{(1 - \beta)\, sim_{STD-SCR}(q_{IV}, d) \\ & \quad + \beta sim_{STD-SCR}(q_{OOV}, d)\}, \end{aligned} \tag{10}$$

where, α and β are weighting coefficients of the linear combination, sim_{cSCR} and $sim_{STD-SCR}$ are relevance scores calculated in the conventional SCR system and the proposed STD-SCR system, respectively. Using higher α means that we prefer the STD-SCR system to the conventional SCR system. Note that $\alpha = 0$ ($\alpha = 1$) corresponds to just using only the conventional SCR (STD-SCR) system. On the other hand, using higher β means that OOV words are taken more importance than IV words when using the STD-SCR system. Note that $\beta = 0.5$ means that we make no distinction between OOV and IV words.

3.4 Experiments

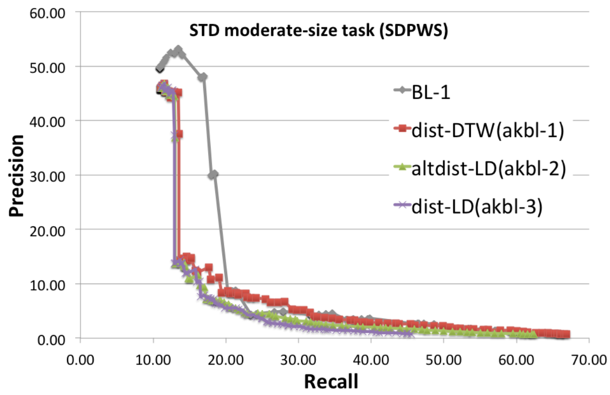

Figure 5: The recall–precision curves for the result on the moderate-size task (micro average)

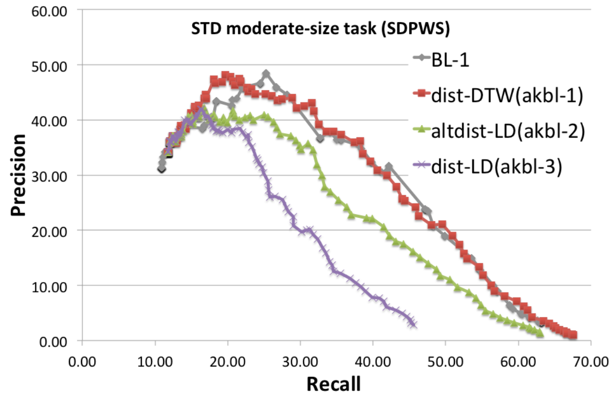

Figure 6: The recall–precision curves for the result on the moderate-size task (macro average)

We submitted eight runs for the lecture retrieval task. The four kinds of approaches were applied to either the matched or unmatched transcription.

STD-SCR system using the E-VSM (AKBL-5,AKBL-6) The proposed STD-based SCR system was used. It used no word-based transcription and was applied only on the syllable-based transcription, where AKBL-5 and AKBL-6 used matched and unmatched transcription, respectively. For the retrieval model, we used the extended vector space model that used the word-combination features, as described in Section 3.2.

STD-SCR system using SMART (AKBL-3,AKBL-4) It was the same system as that used in the AKBL-5 and 6 except that the retrieval model was replaced by the SMART retrieval method, which uses the TF–IDF term weighting with pivoted normalization [17]. AKBL-4 and AKBL-3 used matched and unmatched transcription, respectively.

Hybrid system using the E-VSM (AKBL-1,AKBL-2) We combined the conventional word-based SCR and the syllable-based STD-SCR methods, as described in Section 3.3. For the retrieval model, we used the extended vector space model that used the word-combination features, as described in Section 3.2. The run AKBL-1 was applied on the matched word-based transcription and matched syllable-based transcription, while the AKBL-2 was applied on the unmatched word-based and syllable-based transcriptions.

Hybrid system using SMART (AKBL-7,AKBL-8) It was the same system as that used in the AKBL-1 and 2 except that the retrieval model was replaced by the SMART retrieval method, which uses the TF–IDF term weighting with pivoted normalization [17]. AKBL-7 and AKBL-8 used matched and unmatched transcription, respectively.

The experimental results are summarized in Table 4. The experiment showed that the proposed STD-based SCR method performed well, even though it relied only on the degenerated syllable-based recognition result. Especially on the unmatched condition, where the word error rate was higher, the performance of the STD-based SCR was almost same as that of the conventional SCR. It also showed that the proposed extended vector space model (E-VSM) performed better than the SMART method. Furthermore, the hybrid SCR system of the STD-SCR and the conventional SCR successfully outperformed the baseline conventional SCR methods. We also found that the SMART was more effective than the E-SVM for the hybrid system.

4. DETERMINING THE RANGE OF RELEVANT PASSAGE

4.1 Methods for Passage Retrieval

The SpokenDoc passage retrieval task differs from a conventional document retrieval task in that the segments of passage are not predefined in advance. Therefore, it is required both to determine the boundary of the passage in the document and to rank them according to their relevancy to the query topic. Thus, we extended the conventional document retrieval method to that designed for the passage retrieval.

4.1.1 *Using the Neighboring Context to Index the Passage*

Passages from the same lecture may be related to each other in the passage retrieval task, whereas the target documents are considered to be independent of each other in a conventional document retrieval task. In particular, the neighboring context of a target passage should contain related information. It would seem appropriate for the passage retrieval task to use the neighboring context to index the target passage. Normally, a passage D is indexed by its own term frequencies $TF(t, D)$ of the terms $t \in D$. This can be extended to use the neighboring context for indexing. For the context $context_n(D)$, the preceding n utterances and the following n utterances are used. Therefore, we use

$$TF_{ext}(t, D) = \beta TF(t, D) + TF(t, context_n(D)) \tag{11}$$

where β is introduced to specify the relative importance of D and $context_n(D)$. In our implementation, an utterance is used for D, n and β are set to 5 respectively through the preliminary experiments. This is the same technique we used at NTCIR-9 SpokenDoc SDR task[12].

Although above approach achieved retrieving passages, its results are fixed-length segment. For example, our implement used n=7, thus retrieved passages segment length are always 15 utterances. Since relevant passages are seemed that they have various variable-length segments, we propose automatic estimating passage boundary method.

4.1.2 *Automatic Estimation of Passage Boundary*

When Given the query Q, we first retrieve relevance utterance in spoken document set using Neighboring Context to Index approach which described in section 4.1.1, and obtain pair(D, x) which include spoken document D and relevance utterance position x in D from retrieved results. Each pair is ranked according by their similarity score, and top 4000 pairs are consisting pair set $\mathbf{P}$ in our implementation.

Then, we define d_x as x-th utterance in D from each pair, and find most relevance segment boundary$(\hat{f}, \hat{t})$ about each d_x as follows

$$(\hat{f}, \hat{t}) = \operatorname*{argmax}_{(f,t)\in\{(x-i,x+j)\ |\ 0\leq i\leq l, 0\leq j\leq l\}} sim(Q, d_f^t) \tag{12}$$

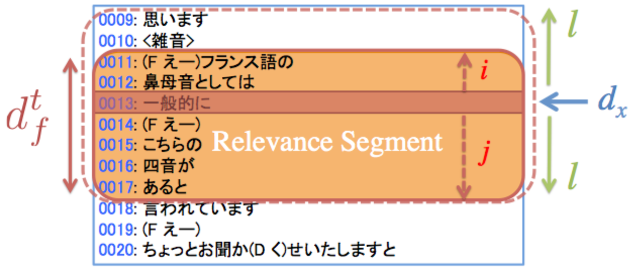

Figure 9: Automatic estimation of passage boundary

Where, $(\hat{f}, \hat{t})$ is pair of f-th and t-th utterance position in D, and d_f^t is a sequence of utterances between i-th and j-th utterances in D. As shown in Figure 9, we calculate similarity between Q and d_f^t with every possible combination of i and j, and finally we submit the most similar passage segment $[x-i, x+j]$. As a result, submitted passages have each different length. In contrast with method that no considered relevancy between Q and passage segment length described in section 4.1.1, this method also consider passage segment length and its relevancy. Therefore, we expect to obtain more relevance passage segment.

In the experiment, retrieved results are obtained using Equation(12) with every pair in **P** and ranked according by their similarity score.

4.1.3 *Penalizing Neighboring Retrieval*

In applying context indexing or automatic estimating passage boundary, neighboring passages tend to be retrieved at the same time as they share the same indexing words. This is not adequate from the perspective of retrieval systems because such systems output many redundant results. For this reason, we penalize a retrieval result that is neighbor to another result that has been output previously. In practice, the retrieved passage is discarded from the output list, if there are other results already retrieved within an utterance neighborhood of it.

As previous our run in NTCIR-9 shows[12], this approach significantly improves the performance in combination with context indexing. So we also use this approach in this round continuously.

4.2 Retrieval Models

4.2.1 *Relevance Models*

Levrenko and Croft [20] proposed relevance models as an information retrieval model. They define the relevance class R to be the subset of documents in a collection C, which are relevant to some particular information need, i.e. $R \subset C$. A relevance model is the probability distribution $P(w|R)$, where $w \in V$ is a word in a vocabulary V . $P(w|R)$ is estimated from a given query Q as follows.

$$P(w|R) = \frac{1}{P(Q)} \sum_{D \in C} P(w|D) \prod_{i=1}^{|Q|} P(q_i|D) \tag{13}$$

where $P(Q)$ is constant with respect to Q.

Then, $P(w|R)$ is used to rank the documents $D \subset C$ by using the Kullback-Leibler divergence between th e distributions $P(w|R)$ and $P(w|D)$:

$$H(R||D) = -\sum_{w \in V} P(w|R) \log P(w|D) \tag{14}$$

Relevance models can be seen as an implementation of pseudo relevance feedback, which is a sort of query-expansion technique using the target document collection, i.e. the query Q is expanded with the related words in the collection C through the estimation of the relevance model $P(w|R)$. In out implementation, we use only top-K documents D with $\prod_{i=i}^{|Q|} P(q_i|D)$ in Equation(13), and vocabulary V is consisted from top-800 w by $P(w|R)$.

Applying relevance models directly to our passage retrieval, specifically the context-indexing method described in Section 4.1.1 is problematic. Because context indexing uses neighboring utterances to index a document (an utterance), several neighboring documents share the same index words. This makes the estimated $P(w|R)$ inaccurate.

In order to deal with this problem, no context-expanded documents, i.e. a set of utterances, are used in the estimation of $P(w|R)$, but then context-expanded documents are ranked using $P(w|R)$. Namely, $P(w|R)$ is estimated as follows:

$$P(w|R) = \sum_{d \in c} P(w|d) \prod_{i=1}^{|Q|} P(q_i|d) \tag{15}$$

where d and c are an utterance and a set of utterances, respectively. Then, the context-expanded documents $\hat{D} \subset \hat{C}$ are ranked by the following equation:

$$H(R||D) = -\sum_{w \in V} P(w|R) \log P(w|\hat{D}) \tag{16}$$

In a similar way, we regard d_t^f as $\hat{D}$ for calculating $sim(Q, d_t^f)$ in Equation(12).

4.2.2 *Query Likelihood Model*

We also applied the query likelihood model[6] as our retrieval model for SCR for similarity between query and document. We use the probability $P(Q|D)$ that a query Q is constructed from a relevant document D:

$$P(Q|D) = \prod_{q \in Q} P(q|D)^{TF(q,Q)} \tag{17}$$

$p(q|D)$ is estimated with dirichlet smoothing[22]:

$$P(q|D) = \frac{TF(q,D)}{|D| + \mu} + \frac{\mu}{|D| + \mu} \frac{TF(q|C)}{|C|} \tag{18}$$

where $|D|$ is number of words in D, $|C|$ is number of document in C, μ is smoothing parameter. The $P(Q|D)$ is used to rank the documents $D \subset C$. In experiment, query likelihood model based similarity was obtained from Equation(17) instead of relevance models based similarity Equation(14), and the context-expanded document $\hat{D} \in \hat{C}$ or d_t^f are used as D.

4.3 Experiments

We submitted three types of runs. One detects variable length passage segments estimated automatically, as described in Section 4.1.2. The others have fixed-length segments, as described in Section 4.1.1. The latter two are the same method used in previous NTCIR-9 SpokenDoc. As the retrieval model, either Query Likelihood Model or Relevance Model is applied, so that we submitted six runs in total. All runs used neighboring penalty, as described in Section 4.1.3. All runs target the *matched* transcription.

The runs akbl-1,2, and 3 are using Relevance Model as the similarity calculation between query and passage. The run akbl-4, 5, and 6 are using Query Likelihood Model. The runs akbl-1 and 4 target variable-length segments, each of which consists of 1 to 25 utterances and are tuned against fMAP score. The akbl-2, 3, 5, and 6 target fixed-length segments, each of which consists of 15 utterances constantly. The akbl-2, and 5 are tuned against fMAP score while the akbl-3, and

Table 5: Experimental results of passage retrieval

run	uMAP	pwMAP	fMAP	retrieval model	definition of passage	tuning against
baseline-1	0.133	0.100	0.087	GETA(SMART)	fixed length	-
baseline-2	0.092	0.082	0.068	GETA(TFIDF)	fixed length	-
AKBL-1	0.102	0.088	0.063	Relevance Model	variable length	fMAP
AKBL-2	0.129	0.125	0.086	Relevance Model	fixed length	fMAP
AKBL-3	0.131	0.137	0.093	Relevance Model	fixed length	pwMAP
AKBL-4	0.131	0.123	0.087	Query Likelihood	variable length	fMAP
AKBL-5	0.122	0.132	0.083	Query Likelihood	fixed length	fMAP
AKBL-6	0.126	0.139	0.089	Query Likelihood	fixed length	pwMAP

5 are against pwMAP score. We used Spoken-Doc1 dryrun 39 queries for training these runs.

The results are shown in Table 5. It showed that there was no significant difference between the two retrieval models, i.e. Query Likelihood and Relevance Model. It also showed that the estimation of the variable length passage did not improve the performance of the passage retrieval. On the other hand, our language modeling approach performed well, compared with the baseline runs and the runs from the other SpokenDoc-2 participants.

5. CONCLUSIONS

In this paper, We investigated three methods for SpokenDoc-2, i.e. the Distance-ordered spoken term detection, the STD-based spoken content retrieval, and the automatic determination of the passage boundaries, which were applied to the STD, the SCR lecture retrieval, and the SCR passage retrieval tasks, respectively.

6. REFERENCES

[1] T. Akiba and K. Honda. Effects of query expansion for spoken document passage retrieval. In *Proceedings of International Conference on Speech Communication and Technology*, pages 2137–2140, 2011.

[2] T. Akiba and K. Honda. Effects of query expansion for spoken documnet passage retrieval. In *Proceedings of International Conference on Speech Communication and Technology*, pages 2137–2140, 2011.

[3] T. Akiba, H. Nishizaki, K. Aikawa, X. Hu, Y. Itoh, T. Kawahara, S. Nakagawa, H. Nanjo, and Y. Yamashita. Overview of the NTCIR-10 SpokenDoc-2 task. In *Proceedings of The Tenth NTCIR Workshop Meeting*, 2013.

[4] C. Chelba and A. Acero. Position specific posterior lattices for indexing speech. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, pages 443–450, 2005.

[5] B. Chen, H. min Wang, and L. shan Lee. Discriminating capabilities of syllable-based features and approaches of utilizing them for voice retrieval of speech information in mandarin chinese. *IEEE Transactions on Speeh and Audio Processing*, 10:303–314, 2002.

[6] W. B. Croft and J. Lafferty. Language modeling for information retrieval. *Kluwer Academic Publishers*, 2003.

[7] T. Hori, L. Hetherington, T. J. Hazen, and J. R. Glass. Open-vocabulary spoken utterance retrieval using confusion networks. In *Proceedings of International Conference on Acoustic, Speech, and Signal Processing*, pages 73–76, 2007.

[8] K. Iwami, Y. Fujii, K. Yamamoto, and S. Nakagawa. Efficient out-of-vocabulary term detection by n-gram array in deices with distance from a syllable lattices. In *Proceedings of International Conference on Acoustics Speech and Signal Processing*, pages 5664–5667, 2011.

[9] A. Jansen, K. Church, and H. Hermansky. Towards spoken term discovery at scalewith zero resources. In *Proceedings of International Conference on Speech Communication and Technology*, pages 1676–1679, 2010.

[10] G. J. F. Jones, J. T. Foote, K. S. Jones, and S. J. Young. Retrieving spoken documents by combining multiple index sources, 1996.

[11] T. Kaneko and T. Akiba. Metric subspace indexing for fast spoken term detection. In *Proceedings of International Conference on Speech Communication and Technology*, pages 689–692, 2010.

[12] T. Kaneko, T. Takigami, and T. Akiba. Std based on hough transform and sdr using std results: Experiments at ntcir-9 spokendoc. In *Proceedings of The Ninth NTCIR Workshop Meeting*, pages 264–270, 2011.

[13] K. Katsurada, S. Teshima, and T. Nitta. Fast keyword detection using suffix array. In *Proceedings of International Conference on Speech Communication and Technology*, 2009.

[14] Y. Pan, H. Chang, B. Chen, and L. Lee. Subword-based position specific posterior lattices (S-PSPL) for indexing speech information. In *Proceedings of International Conference on Speech Communication and Technology*, pages 318–321, 2007.

[15] Y.-c. Pan and L.-s. Lee. Performance analysis for lattice-based speech indexing approaches using words and subword units. *Trans. Audio, Speech and Lang. Proc.*, 18(6):1562–1574, Aug. 2010.

[16] M. Saraclar and R. Sproat. Lattice-based search for spoken utterance retrieval. In *Proceedings of Human Language Technology Conference*, 2004.

[17] A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *Proceedings of ACM SIGIR*, pages 21–29, 1996.

[18] K. Sugimoto, H. Nishizaki, and Y. Sekiguchi. Effect of document expansion using web documents for spoken documents retrieval. In *Proceedings of the 2nd Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, pages 526–529, 2010.

[19] M. Terao, T. Koshinaka, S. Ando, R. Isotani, and A. Okumura. Open-vocabulary spoken-document retrieval based on query expansion using related web documents. In *Proceedings of International Conference on Speech Communication and Technology*, pages 2171–2174, 2008.

[20] V.Lavrenko and W. B. Croft. Relevance models in information retrieval. *In Language Modeling for Information Retrieval*, pages 11–56, 2003.

[21] M. Wechsler, E. Munteanu, and P. Schäuble. New techniques for open-vocabulary spoken document retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, pages 20–27, New York, NY, USA, 1998. ACM.

[22] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems (TOIS)*, 22(2):179–214, 2004.

[23] Z.-Y. Zhou, P. Yu, C. Chelba, and F. Seide. Towards spoken-document retrieval for the internet: Lattice indexing for large-scale web-search architectures. In *Proceedings of Human Language Technology Conference*, pages 415–422, 2006.