POSTECH at NTCIR-5: Combining Evidences of Multiple Term Extractions for Mono-lingual and Cross-lingual Retrieval in Korean and Japanese

Seung-Hoon Na In-Su Kang Jong-Hyeok Lee
Div. of Electrical and Computer Engineering
Pohang University of Science and Technology (POSTECH)
Advanced Information Technology Research Center (AITrc)
San 31, Hyoja-Dong, Pohang, Republic of Korea, 790-784
{nsh1979, dbaisk, jhlee}@postech.ac.kr

October 20, 2005

Abstract

This paper describes methodologies for NTCIR-5 CLIR involving Korean and Japanese, and reports the official result as well as retrieval results using NTCIR-3 and NTCIR-4 data. We participated in four tasks: K-K and J-J monolingual tracks and K-J and J-K cross-lingual tracks. Unlike English, in Asian languages such as Korean and Japanese term extraction is nontrivial because of segmentation ambiguities. In this regard, we prepared multiple term representations for documents and queries, of which ranked results are merged to generate final ranking. In preliminary experiments using NTCIR-3 and NTCIR-4 data, our model showed the best performances for description queries in Korean and Japanese. In offline results using NTCIR-5 data, our methodology in Korean showed the best performance by archieving 0.5680 for description queries and 0.6159 for others.

Keywords: Information Retrieval, Cross-lingual Information Retrieval, Multiple Evidence Combination, Unsupervised Segmentation, Query Translation, Probabilistic Retrieval Model, Language Modeling Approach

1 Introduction

Unlike English, Chinese and Japanese do not use word delimiters in a normal text. In Korean, no word boundaries exist within *Eojeol*. [1] Thus, word segmentation is nontrivial for the three Asign languages. Compared with Japanese, segmentation problem of Korean is more difficult because the basic character unit used in Korean is *Hangul* character not *Hanzi*: the number of different *Hangul* characters is much smaller than that of *Hanzi*s.

To avoid word segmentation problem, one can use character n-gram method which produces overlapping n-character strings as index terms. In Korean, the character n-gram method shows stable and robust retrieval performance although it is very simple term extraction method. However, the use of character n-grams has a limitation that they do not make semantically consistent units. Sometimes, the extraction of character n-grams may be dangerous because the method generates a sequence of semantically un-related terms from a given *Eojeol* which may have negative effects on the retrieval performance.

On the other hand, dictionary-based word segmentation can extract semantically consistent units, however, it has the difficulty in segmenting unknown words. Thus, the adaptation of a dictionary is fundamental for higher retrieval performance. However, the hand-driven adaptation of a dictionary is time-consuming. Specially, a dictionary manager may hesitate to decide what is a content word. For example, from "불린함수" (Boolean function), one may extract two content words such as "불린" (Boolean) and "함수" (fuction), and the other may consider "불린함수" as a single content word. This problem is similar to the phrase extraction problem in English.

To relax such an adaptation problem of dictionary-based word segmentation, we have developed an unsupervised segmentation algorithm without requiring any dictionaries. The algorithm sets a statistical lexicon from a given collection and performs a hybrid segmentation algorithm

[1] Eojeol indicates a Korean spacing unit as well as a syntactic unit.

based on a rule and statistics on query and documents.

As preliminary experiments, we have performed retrievals using three different term extractions for NTCIR-3 and NTCIR-4 data. Then, from their query-by-query analyses, we have found that the best term extraction scheme is different for each query. This observation makes us build the retrieval system to reflect multiple evidences of different term extractions. For combination of multiple evidences, we used a fusion-based approach which merges retrieval results from multiple representations. We expect that the combination covers some deficits of other extraction methods. For Japanese, we used two term extractions based on Japanese morphological analyzer - COBALT-JK [4] and ChaSen. [2]

For cross-lingual information retrieval, we use a naive query translation method (NQT) which does not use any word sense disambiguation method based on statistics such as co-occurrence information.

This paper is organized as follows. Section 2 describes an overview of our monolingual retrieval architecture by introducing retrieval model, feedback method, a combination approach and term extraction schemes. In Section 3, we describes cross-lingual retrieval methodologies. Section 4 shows official results and compares them with retrieval results using NTCIR-3 and NTCIR-4 data. Finally, Section 5 provides our conclusion.

2 Monolingual Retrieval

2.1 Overall Architecture

Figure 1 shows the overall architecture of our system for monolingual retrieval in Korean. Basically, the system uses three different term extractions and merges retrieval results from them. The extraction methods are *Character Bi-gram*, *Dictionary-Based Word* and *Collection-Based Segment*. Our intuition is that each extraction method plays discriminative effects on retrieval performance, and can relax the problem of segmentation difficulty. In addition to the combination of term representations, two different retrieval models are combined to optimize the retrieval performance at different retrieval strategies - probabilistic retrieval model [13] and language modeling approach [12]. In pseudo relevance feedback, we use different methods according to the length of query - Model-based feedback [16] for long queries and expansion-based feedback based on likelihood ratio [12] for short queries.

[2] http://chasen.naist.jp/

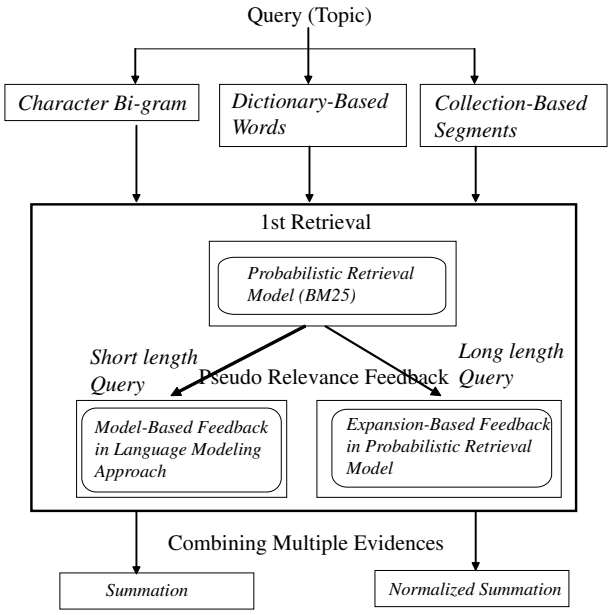

Figure 1: Overall architecture for monolingual retrieval of Korean

2.2 Retrieval Model

The initial retrieval is performed by the BM25 formula of Okapi. Pseudo relevance feedback is executed by using model-based feedback for short queries, and expansion-based feedback for long queries. In pseudo relevance feedback, the use of different strategies according to query length is motivated from our previous research [8]. Okapi's term weighting formula of term t_i in document D_j is as Eq.(1)

$$w_{ij} = w_i' \frac{tf_{ij}}{K + tf_{ij}} \frac{qtf_i}{k_3 + qtf_i} \quad (1)$$

where K is $k_1((1-b)+b\frac{dl_j}{avgdl})$ and tf_{ij} is term frequency of t_i in document D_j. w_i' is based on the Robertson-Sparck Jones weight [14], which is reduced inverse document frequency weight without relevance information ($R = r = 0$) as Eq.(2).

$$w_i' = log \frac{(r_i + 0.5)/(R_i - r_i + 0.5)}{(n_i - r_i + 0.5)/(N - n_i - R + r_i + 0.5)} \quad (2)$$

where N is the number of documents and R is the number of relevant documents, n_i is the document frequency of t_i and r_i is the frequency of documents to be relevant containing t_i. k_1, b and k_3 are set to 2.0, 0.75 and ∞, respectively.

Model-based feedback is performed on top retrieved documents (feedback documents) $\mathcal{F}$ [16]. Query model is estimated by using EM algorithm to maximize likelihood of top-retrieved documents given a mixture model which consists of unknown

query model θ_Q and background collection language model θ_C. Unlike original Zhai's approach, we modified the likelihood of feedback documents by reflecting the score of retrieved documents as follows.

$$\mathcal{L} = \sum_i \sum_{d_j \in \mathcal{F}} tf_{ij} rel_j \log \left(\begin{array}{l} (1-\lambda) P(t_i|\theta_Q) \\ +\lambda P(t_i|\theta_C) \end{array} \right) \quad (3)$$

where rel_j is the relevance score of d_j. Given query Q and document model θ_{D_j}, rel_j is formulated as

$$rel_j = \kappa + (1-\kappa) \frac{\log P(Q|\theta_{D_j})}{\max_j \log P(Q|\theta_{D_j})} \quad (4)$$

where κ is a tuning parameter. In our preliminary experimentation ($\kappa = 0.7$) using NTCIR-3 and NTCIR-4 Korean test sets, the modified likelihood showed slightly better performance with about 1% difference.

Expansion-based feedback has only been dealt with heuristically in a given retrieval model. The original query is usually literally expanded by adding additional terms to it based on some criterion. Our criterion is Ponte's likelihood ratio [12] as follows.

$$Score(t_i) = \sum_{d_j \in F} \log \left(\frac{P(t_i|\theta_{D_j})}{P(t_i|\theta_C)} \right) \quad (5)$$

After adding terms into original query, all of them are entered as an input to probabilistic retrieval model without re-weighting.

2.3 Term Extraction

For Korean, we prepared three different methods for term extraction as follows.

Character Bi-gram *Character Bi-gram* is the well-known term extraction method for Asian languages such as Korean, Japanese and Chinese [6]. Character bi-gram consists of two consequent Korean characters (*Emjeols* in Korean). Special characters such as numeric and English characters are pre-extracted. For example, for Eojeol '배아줄기세포' (embryonic stem cell), terms of '배아' (embryonic), '아줄' (non-sense syllables), '줄기' (stem), '기세' (spirit) and '세포' (cell) are extracted.

Dictionary-Based Word *Dictionary-Based Word* is produced by applying our Korean morphological analyzer. Our morphological analyzer selects content nouns and numerical words by using compound-noun segmentation based on longest-matching rule [3]. The size of dictionary is about 230,000 nouns, and its entries contains most Korean words and modern foreign words.

Collection-Based Segment *Collection-Based Segments* are extracted by applying unsupervised segmentation algorithm without dictionary. This problem is related to automatic lexicon construction [1, 15, 10]. In information retrieval, unsupervised method is motivated from the fact that there are many unknown words in a given test collection, thus, the segmentation performance for the given corpus is not acceptable without hard-tuning to the domain of collection. By using unsupervised method, unknown terms can be automatically learned based on collection statistics. As a result, we can expect that segmentation accuracy will be improved. Our unsupervised method is different from incremental approaches [1, 15] and iterative approaches [10]. Our method basically employs global search, but does not attempt to learn the statistical dictionary. [3] Instead, we focus on pruning unhelpful segmentation candidates over the search space based on simple principle. The unsupervised segmentation algorithm will be described in the next sub-section.

For Japanese, we prepared two methods for term extractions. One method is based on Japanese morphological analyzer of COBALT-JK, and another method is based on Chasen. In Japanese, we did not apply unsupervised segmentation.

2.4 Unsupervised Segmentation Method

Let us assume that we have a raw corpus $\mathcal{C}$ and we want to segment an n-character string $T = c_1...c_n$ (c_i is the i-th character). As an alternative notation for $c_1...c_n$, we use c_{1n}. First, we create the statistical dictionary D that is a set of all-length character n-grams of each string in $\mathcal{C}$. In order to find the most likely segmentation candidate $S*$ of T, we should calculate Eq.(6), where k-th segmentation candidate is represented as $S_k = s_1...s_{m(k)}$ (s_i is the i-th segment which belongs to D, and $m(k)$ is the index of the last segment of S_k, and $m(k) \leq n$). Note that a segment covers one or more contiguous characters in T. We interpret $P(S_k)$ as the probability that T is decomposed into a sequence of $s_1, s_2, ..., s_{m(k)}$.

$$S* = \underset{S_k = s_1 ... s_{m(k)}}{\operatorname{argmax}} \; P(S_k) \quad (6)$$

The calculation of $P(S_k)$ is simplified to Eq.(7) by assuming the independence between segments which has been adopted by most unsupervised segmentation methods.

[3] Global search considers all possible segmentation candidates to select the most likely one

Symbol	*Segments*	$P(S_k)$
S_1	abcd	0.05
S_2	a+bcd	0.03
S_3	abc+d	0.02
S_4	ab+cd	0.04
S_5	a+b+cd	0.01
S_6	ab+c+d	0.005
S_7	a+bc+d	0.005
S_8	a+b+c+d	0.001

Table 1: Sorted results of feasible segmentation candidates with $K = 4$

Symbol	*Segments*	$P(S_k)$
S_4	ab+cd	0.04
S_5	a+b+cd	0.01
S_6	ab+c+d	0.005
S_7	a+bc+d	0.005
S_8	a+b+c+d	0.001
S_1	abcd	0.05
S_2	a+bcd	0.03
S_3	abc+d	0.02

Table 2: Sorted results of feasible segmentation candidates with $K = 4$ when applying length principle

$$S* = \underset{S_k=s_1 \ldots s_{m(k)}}{\operatorname{argmax}} \prod_{i=1}^{m(k)} P(s_i) \qquad (7)$$

However, Eq.(7) tends to produce the segmentation candidate that has the smaller number of segments. Eq.(7) would divide the input string T into a few large segments. This means that the naive application of Eq.(7) may under-segment the input. To prevent under-segmentation, we attempt to obviate this problem by applying following segmentation principle to Eq.(7).

Length Priciple: Given K and the set of feasible segmentation candidates, segmentation prefers the result in which the length of all segments is smaller than K. A parameter K indicates a minimum character length of the substring. A feasible segmentation candidate is a segment sequence S_k of which $P(S_k)$ is positive. According to this principle, our segmentation prefers segments of which all lengths are smaller than K. For example, for a string *abcd*, Table 1 enumerates feasible segmentation candidates with $K = 3$.

If we use only Eq.(7) without length principle, then S_1 will be selected because $P(S_1)$ have largest segment probability. However, when applying length principle, we re-organize above candidates by their preferences as Table 2.

Now, *abcd*, which is top ranked in Table 1, is low-ranked, showing lower preference than $a+b+c+d$. As a result, $ab + cd$ is selected for the best segmentation result. If $P(ab + cd)$ is 0 in collection statistics, then other candidate will be selected. To implement Eq.(7) with length principle, we modify standard CYK algorithm. The complete procedure for finding the best segments can now be stated as follows.

1) Initialization : $(q - p + 1) < K$

$$\begin{aligned} \delta_{pq} &= P(c_{pq}) \\ \psi_{pq} &= q \end{aligned}$$

2) Recursion : $(q - p + 1) \geq K$

$$\begin{aligned} \hat{\delta}_{pq} &= \max_{1 \leq r \leq q-1} \delta_{pr}\delta_{r+1q}P(r|p,q) \\ \hat{\psi}_{pq} &= \underset{1 \leq r \leq q-1}{\operatorname{argmax}} \delta_{pr}\delta_{r+1q}P(r|p,q) \\ \delta_{pq} &= \begin{cases} P(c_{pq}) & \text{if } \hat{\delta}_{pq} = 0 \\ \delta_{pr}\delta_{r+1q}P(r|p,q) & \text{otherwise} \end{cases} \\ \psi_{pq} &= \begin{cases} q & \text{if } \hat{\delta}_{pq} = 0 \\ \hat{\psi}_{pq} & \text{otherwise} \end{cases} \end{aligned}$$

3) Termination

$$\begin{aligned} P(S*) &= \delta_{1n} \\ S* &= backtrack(\psi_{1n}) \end{aligned}$$

4) Backtracking

$$S_{pq}* = \begin{cases} c_{pq} & \text{if } \psi_{pq} = q \\ (S_{p\psi_{pq}*})(S_{(\psi_{pq}+1)q}*) & \text{otherwise} \end{cases}$$

2.5 Multiple Evidence Combination

Each term representation yields one evidence for a document. Final ranked results are obtained by combining such multiple evidences. Let the score of document D_i be $score_i$. There are two methods for multiple evidence combinations. First method is *SUM*, which is summation of scores of a document generated from each evidence ($\sum score_i$). Second method is *NORM-SUM*. Let $norm_i$ (corresponds to Max_Norm [5]) be normalized scores by maximum score value .

$$norm_i = \frac{score_i}{\max_k score_k}$$

NORM-SUM is the summation of normalized scores ($\sum norm_i$).

In our system, different combination methods are used according to the length of query. We select *SUM* for short query and *NORM-SUM* for long query because this selection was robust empirically.

	# of translation pairs	*# of source language terms*	*# dictionary ambiguity*
K-J	420,650	303,199	1.39
J-K	434,672	399,220	1.09

Table 3: Bilingual dictionaries

3 Cross-lingual Retrieval

There are two traditional approaches in cross-lingual retrieval; query-translation (QT) and document-translation (DT). It is reported that their combination improves performance due to different effects for retrieval performance of individual method. Because the process of document-translation requires large resource and high time cost for applying in real situation, we have developed pseudo document translation (PDT) method and have participated at NTCIR-4 by combining it with query translation [2]. We have found that PDT is exactly the same as Pirkola's method [11] when lengths of all documents are equal. Thus, the combination of PDT and QT will be equivalent to the combination of Pirkola's method with QT. This consideration will significantly reduce time complexity of PDT for a given collection.

However, at NTCIR-5, we did not submit such combinations of QT and Pirkola's method. Instead we performed only naive query translation (NQT) focusing on combining multiple evidences which are generated from different term extractions. If this result is combined with Pirkola's method, performance will be more improved.

3.1 Bilingual Dictionary

Table 3 shows some statistics about our bilingual dictionaries used at NTCIR-5 CLIR. These dictionaries were extracted from dictionaries created for machine translation (MT) systems. Note that ambiguity of K-J is higher than that of J-K, which might be originally caused by the difference of characters used in two languages. In regards of meaning representation, Chinese character, which is frequently used in Japanese, is less-ambiguity than Korean character. In Korean language, several different Chinese characters can be equally pronounced to single Korean character.

3.2 Naive Query Translation (NQT) Method

Naive query translation method is a simple dictionary-based translation method. For given source language query $Q_s = q_1 q_2 ... q_n$, each query term q_i is expanded to translation candidates $t_{i1} ... t_{im(i)}$ by using bi-lingual dictionary and there are no additional weights for expanded terms. This method is simple since it does not contain any other disambiguation procedure and is normally used as the baseline in BLIR research. In spite of this fact, this method provides fundamental retrieval performance because of effects of self-disambiguation. The effect of self-disambiguation is originated from characteristics of information retrieval where the score of documents is assigned according to the degree of matching of multiple query terms. Thus, it is highly plausible that feasible documents will collectively match only topically related terms.

3.3 Combination of Multiple Evidences

As like monolingual retrieval, there are multiple query representations for cross-lingual retrieval. They are merged to generate the final ranked result. Their representations are dependent on methods used in monolingual retrieval. In J-K retrieval, three representations are available such as character n-gram, dictionary-based words and collection-based segments since the target language is Korean. Similarly, in K-J retrieval, two representations are available.

Concerning J-K retrieval, we can obtain only dictionary-based word by translating the given query. Other representations such as collection-based segment cannot be obtained by using translation due to lack of bilingual dictionary. To make other representations, we perform segmentation on each translated word. In other words, collection-based segment is obtained by decomposing initial dictionary-based target term into smaller segments based on statistical dictionary in the collection (Section 2.3.1). As a result, they can be used to retrieve indexes of collection-based segments in Korean. It is more simple to convert dictionary-based word into character bi-gram by extracting character bi-grams from it.

Similarly, in K-J retrieval, two representations are available. Since bilingual dictionary was used in COBALT-JK system, translated terms can be used to retrieve COBALT-JK index, but not on Chasen index. To make these terms be terms used in Chasen, they are decomposed into smaller one or merged into larger one by checking whether resulting terms are included or not in index terms in Chasen.

4 Experimentation

This section reports the retrieval results of our official runs submitted to NTCIR-5 CLIR

NTCIR-3			
Method	T	D	TDNC
BG	0.3068	0.2651	N/A
DW	0.2750	0.2341	N/A
CS	0.2785	0.2153	N/A
BGp	0.3718	0.3668	N/A
DWp	0.3887	0.3378	N/A
CSp	0.3792	0.3241	N/A
BGp+DWp+CSp	**0.4285**	**0.3859**	N/A
Top	0.3317	0.3602	N/A
NTCIR-4			
Method	T	D	TDNC
BG	0.4403	0.4191	0.5279
DW	0.3894	0.3838	0.5009
CS	0.4412	0.4385	0.5382
BGp	0.5328	0.5165	0.5782
DWp	0.5012	0.4789	0.5453
CSp	0.5242	0.5267	0.5664
BGp+DWp+CSp	**0.5682**	**0.5570**	0.6063
Top	0.5361	0.5097	**0.6212**

Table 4: Preliminary Korean SLIR experiments at NTCIR-3 and NTCIR-4

task, involving preliminary experimentation of same NTCIR-3 and NTCIR-4 track. Evaluation measure is non-interpolated average precision (AvgPr). Each topic has four fields: title (T), description (D), narrative (N) and concepts (C). We submitted our runs using T, D and TDNC. Relevance judgements with relax version are used.

For language modeling approach, we use Jelinek smoothing of which parameter λ is 0.75 [17]. For unsupervised segmentation, K is set to 3 which is tuned in Korean language. For pseudo relevance feedback, we use top R documents where R is set to 15 for Korean and 20 for Japanese, respectively. The total number of expansion terms and original query terms are limited to 200. κ is set to 0.7 for Korean and 0.6 for Japanese.

4.1 SLIR Track

4.1.1 Preliminary Experimentations at NTCIR-3 and NTCIR-4

Table 4 shows the preliminary Korean retrieval results using NTCIR-3 and NTCIR-4 test set. We use notation for each term extraction method - character bi-gram (BG), dictionary-based word (DW) and collection-based segment (CS). If pseudo relevance feedback (PRF) is performed, symbol "p" is attached to tag name of initial retrieval. Thus, CSp means that initial retrieval is performed by using term extraction method of collection-based segments and then pseudo relevance feedback is applied. Bold face indicates that the run achieves the best performance at the given task. N/A means that the retrieval result is not available at this time.

Method	T	D	TDNC
COB	0.3486	0.3557	N/A
CHA	0.3441	0.3432	N/A
COBp	0.4690	0.4980	N/A
CHAp	0.4693	0.4732	N/A
COBp+CHAp	0.4821	**0.5026**	N/A
Top	**0.4864**	0.4831	N/A

Table 5: Preliminary Japanese SLIR experiments at NTCIR-4

At NTCIR-3, in initial retrieval, BG shows superior performance to DW and CS on T and D. After PRF, in Title (T) DWp is better than BGp, reversing results of initial retrieval. In Description (D) BGp preserves superior performance to other methods. Remarkably, the combining method (BGp+DWp+CSp) significantly improves the best of individual method, showing that the improvement over the best is about 10.2% ((0.4285 - 0.3887) / 0.3887) and 5.2% ((0.3859 - 0.3668) / 0.3668) in T and D, respectively. In addition, this result surpasses the performances of top system at NTCIR-3.

At NTCIR-4, the results are somewhat different from NTCIR-3. In initial retrieval, CS is superior to DW on T, D and TDNC, to BG on D and TDNC. After PRF, BGp becomes better than CSp on T and TDNC. On D, CSp preserves the best performance over other methods. As like NTCIR-3, the combination method significantly improves all of individual methods, showing that the improvement over the best is about 6.64%, 5.75% and 4.85% on T, D and TDNC, respectively. This results also are better than performances of the best system at NTCIR-4 except for TDNC.

Table 5 shows the preliminary Japanese retrieval results using NTCIR4 test set. We use notation for each term extraction method - COBALT-JK (COB) and Chasen (CHA). Two methods are comparative on T, while COB are robust over CHA on D. The combination method slightly improves all of individual methods. The results are comparative to the best system at NTCIR-4.

4.1.2 Official Results at NTCIR-5

Table 6 shows official Korean retrieval results in NTCIR-5. Unlike NTCIR-3 and NTCIR-4, BG fails on short length query. Thus, the combination method does not obtain marginal effects, of which performances are almost the same to CSp. However, our official result (BGp+DWp+CSp) is

Method	T	D	TDNC
BG	0.3847	0.4212	0.5381
DW	0.3748	0.3961	0.5114
CS	0.4199	0.4381	0.5639
BGp	0.4793	0.5136	0.5777
DWp	0.5031	0.5338	0.5729
CSp	0.5309	0.5638	0.6085
BGp+DWp+CSp	0.5316	0.5680	**0.6159**
DWp+CSp	0.5400	**0.5790**	0.6120
Top	**0.5441**	0.5680	0.6159

Table 6: Korean SLIR performance at NTCIR-5

Method	T	D	TDNC
COB	0.3386	0.3257	0.4529
CHA	0.2973	0.3043	N/A
COBp	0.4434	0.4044	0.5068
CHAp	0.4219	0.3976	N/A
COBp+CHAp	0.4536	0.4275	0.5068
Top	**0.5028**	**0.4707**	**0.5427**

Table 7: Japanese SLIR performance at NTCIR-5

promising, which shows the top performance on D and TDNC, and comparative performance over top performance on T. Because of failure of BG, we performed further experiment of combination of only DWp and CSp without BGp. This combination method (DWp+CSp) shows better performances on triple combination (BGp+DWp+CSp) on T and D. For short length queries, BG plays a negative effects on retrieval performance when using combination.

Table 7 shows official Japanese retrieval results in NTCIR-5. As like NTCIR-4, they shows effects of the combination. COB is robust over CHA. For some reasons, we cannot obtain the results of CHA in TDNC, so combination on TDNC is not performed. Performances of our systems on Japanese are less than top performances at NTCIR-5 showing performance difference about 5%.

4.2 BLIR Track

Table 8 shows official J-K retrieval results in NTCIR-5. Since target language is Korean, BG, DW and CS methods are available. As like monolingual retrieval, BG fails on retrieval performance. In spite of failure of BG, our official run, the combination of three methods (BGp+DWp+CSp) significantly improves all of individual methods. We further performed the combination of only DWp and CSp then the results are better than triple combination on T, D and TDNC. Similar to the monolingual result, BG plays negative effects on retrieval performance when it is combined.

Method	T	D	TDNC
BG	0.2709	0.3092	0.4358
DW	0.2903	0.3156	0.4052
CS	0.3054	0.3359	0.4767
BGp	0.3736	0.4304	0.4920
DWp	0.4218	0.4482	0.4960
CSp	0.4197	0.4502	0.5356
BGp+DWp+CSp	0.4597	0.4849	0.5509
DWp+CSp	**0.4722**	**0.5020**	**0.5572**
Top	0.4597	0.4849	0.5509

Table 8: J-K BLIR performance at NTCIR-5

Method	T	D	TDNC
COB	0.1743	0.2484	0.3711
CHA	0.1498	0.2274	0.3575
COBp	0.2842	0.3782	0.4447
CHAp	0.2712	N/A	0.4391
COBp+CHAp	**0.2923**	**0.3750**	**0.4643**
Top	0.2923	0.3750	0.4643

Table 9: K-J BLIR Performance at NTCIR-5

Table 9 shows official K-J retrieval results in NTCIR-5. Since target language is Japanese, COB and CHA methods are available. In initial retrieval, COB is superior to CHA on T, D and TDNC, however, two methods become comparative after PRF. Thus, in our official run (COBp+CHAp), there are some positive effects on retrieval performance when both are combined. The performance of this combination is the best.

Table 10 shows the distribution of averages of AvgPr across different combinations of query fields and performance ratio for corresponding SLIR. The relative difficulty of K-J over J-K may be related to dictionary ambiguity (sense ambiguity) in Table 3.

5 Conclusion

For NTCIR-5 SLIR, we employed a coupling strategy that combines several ranked lists generated from multiple term representations by differenti-

	NTCIR-5	
Run	Average of AvgPr	% SLIR
J-K	0.4985	87.18
K-J	0.3438	74.13

Table 10: Averages of AvgPr and performance ratios for corresponding SLIRs

ating pseudo relevance feedback and combination method according to the length of queries. We use three term extractions for Korean which consist of character n-gram and dictionary-based word and collection-based segment indexes, and two term extractions for Japanese which use morphological analysis in COBALT-JK and Chasen. For NTCIR-5 BLIR, we experimented with a strategy based on a naive query translation and the same coupling strategies. Remarkable observation is that collection-based segment index by using unsupervised segmentation algorithm works well in most NTCIR tasks. In the future, we use unsupervised methods based on automatic dictionary construction such as incremental or iterative approach to improve retrieval performance. We plan to apply our unsupervised segmentation method to other Asian languages such as Japanese and Chinese. In addition, it is interesting to implement the combination of multiple evidences in the context of discriminative model [7, 9] which is recently developed.

References

[1] M. R. Brent. An efficient, probabilistically sound algorithm for segmentation andword discovery. *Mach. Learn.*, 34(1-3):71–105, 1999.

[2] I.-S. Kang, S.-H. Na, and J.-H. Lee. Postech at ntcir-4: Cjke monolingual and korean-related cross-language experiments. In *NTCIR-4: Working Notes of the Fourth NTCIR Workshop Meeting*, pages 89–95, 2004.

[3] S.-S. Kang. Korean compound noun decomposition algorithm (in korean). *Journal of the Korean Information Science Society (KISS)*, 25(1):172–182, 1998.

[4] E.-J. Kim and J.-H. Lee. A collocation-based transfer model for japanese-to- korean machine translation. In *NLPRS '93: Proceedings of the 2nd Natural Language Processing Pacific Rim Symposium*, pages 223–231, 1993.

[5] J.-H. Lee. Combining multiple evidence from different properties of weighting schemes. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 180–188, 1995.

[6] J. H. Lee and J. S. Ahn. Using n-grams for korean text retrieval. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 216–224, 1996.

[7] D. Metzler and W. B. Croft. A markov random field model for term dependencies. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 472–479, 2005.

[8] S.-H. Na, I.-S. Kang, and J.-H. Lee. Improving relevance feedback in the language modeling approach: Maximum a posteriori probability criterion and three-component mixture model. In *IJCNLP-04: The First International Joint Conference on Natural Language Processing*, pages 189–194, 2004.

[9] R. Nallapati. Discriminative models for information retrieval. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 64–71, 2004.

[10] F. Peng and D. Schuurmans. A hierarchical em approach to word segmentation. In *NLPRS '01: Proceedings of the 6th Natural Language Processing Pacific Rim Symposium*, pages 475–480, 2001.

[11] A. Pirkola. The effects of query structure and dictionary setups in dictionary-based cross-language information retrieval. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 55–63, 1998.

[12] A. Ponte. *A Language Modeling Approach to Information Retrieval.* PhD thesis, University of Massachusetts, 1998.

[13] S. Robertson and K. S. Jones. Relevance weighting of search terms. *Journal of Royal Statistical Society*, 27(3):129–146, 1976.

[14] S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 232–241, 1994.

[15] A. Venkataraman. A statistical model for word discovery in transcribed speech. *Computational Linguistics*, 27(3):352–372, 2001.

[16] C. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, pages 403–410, 2001.

[17] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR '01: Proceedings of the 24nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 334–342, 2001.