POSTECH at NTCIR-6: Combining Evidences of Multiple Term Extractions for Mono-lingual and Cross-lingual Retrieval in Korean and Japanese

Seung-Hoon Na Jungi Kim Ye-Ha Lee 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, yangpa, sion, jhlee}@postech.ac.kr

Abstract

This paper describes our methodologies for NTCIR-6 CLIR involving Korean and Japanese, and reports the official result for Stage 1 and Stage 2. We participated in three tracks: K-K and J-J monolingual tracks and J-K cross-lingual tracks. As in the previous year, we focus on handling segmentation ambiguities in Asian languages. As a result, we prepared multiple term representations for documents and queries, of which ranked results are merged to generate final ranking. From official results, our methodology in Korean won the top in 6 subtasks of total 9 subtasks for Stage 2,and won the top in 2 subtasks of total 3 subtasks for Stage 1. Even though our system is the same as the previous one, final performances from NTCIR-3 to NTCIR-5 are further improved over our previous results by slightly modifying the feedback parameters.

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 *Kanji*: the number of different *Hangul* characters is much smaller than that of *Kanji*s.

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

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 a very simple term extraction method. However, the use of character n-grams has a limitation that they do not produce 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. In particular, a dictionary manager may hesitate to decide on 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 based on a rule and statistics on query and documents.

We participated in three tracks: K-K and J-J monolingual tracks and J-K cross-lingual tracks. For K-K mono-lingual track, we have examined retrieval performances of three different term extractions in previous NTCIR test collections. Then, from query-by-query analysis, we have found that the best term extraction scheme is different for each query. This ob-

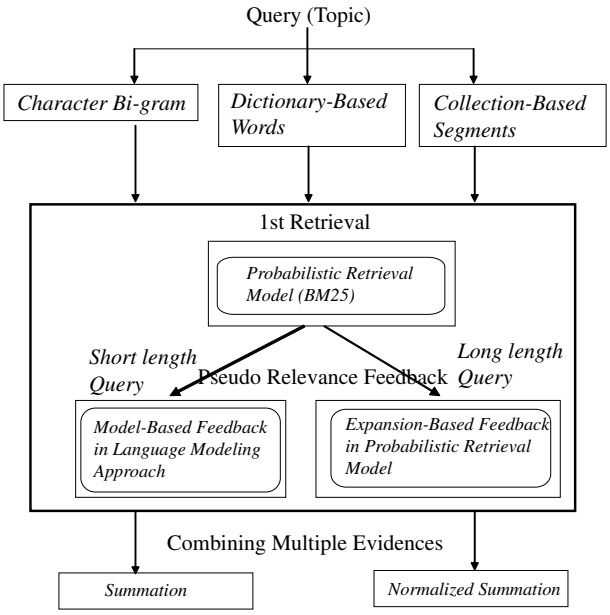

Figure 1. Overall architecture for monolingual retrieval of Korean

servation makes us build the retrieval system to reflect multiple evidences of different term extractions by using a fusion-based approach which merges retrieval results from multiple representations. For J-J mono-lingual track, we applied the single term extraction method based on Chasen, due to time limitation,

For J-K cross-lingual track, 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.

The remainder of 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 describe cross-lingual retrieval methodologies. Section 4 shows official results. 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. The architecture is the same as our previous NTCIR system [7]. 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*. We expect that each extraction method to produce discriminative effects on retrieval performance, and 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 [11] and language modeling approach [10]. In pseudo relevance feedback, we use different methods according to the length of query: Model-based feedback [14] for long queries and expansion-based feedback based on likelihood ratio [10] for short queries.

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 [6]. Okapi's term weighting formula of term t_i in document D_j is as shown in 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 [12], which has reduced the inverse document frequency weight without relevance information ($R = r = 0$) as shown in 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, 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) $\mathcal{F}$ [14]. Query model is estimated by using EM algorithm to maximize the 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.

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

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

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

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 a slightly better performance with about 1% difference.

Let $\theta_Q^{\mathcal{F}}$ be the feedback query model which is obtained by maximizing the likelihood (Eq. (3)). Then, the final query model θ'_Q is defined by linearly-combining the original query model $\hat{\theta}_Q$ and the feedback query model using interpolating parameter α as follows.

$$\theta'_Q = \alpha\theta_Q^{\mathcal{F}} + (1-\alpha)\hat{\theta}_Q \tag{4}$$

Expansion-based feedback has only been dealt heuristically in a given retrieval model. The original query is usually expanded by adding additional terms based on some criterion. Our criterion is Ponte's likelihood ratio [10] as follows.

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

After adding terms into the original query, these terms 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 [5]. 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' (nonsense 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 the longest-matching rule [3]. The size of dictionary is about 230,000 nouns, and its entries contain most of the 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, 13, 8]. In information retrieval, the 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 the unsupervised method, unknown terms can be automatically learned based on collection statistics. As a result, we can expect the segmentation accuracy to improve. Our unsupervised method is different from incremental approaches [1, 13] and iterative approaches [8]. Our method basically employs global search, but does not attempt to learn the statistical dictionary. [2] Instead, we focus on pruning unhelpful segmentation candidates over the search space based on a simple principle. The unsupervised segmentation algorithm will be described in the next sub-section.

For 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 $s_1, s_2, ..., s_{m(k)}$.

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

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

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

However, Eq. (7) tends to produce a segmentation candidate that has the smaller number of segments. Eq. (7) would divide the input string T into a few large segments, which 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 the 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

[2] 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

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 $P(S_1)$ has the largest segment probability. However, when applying length principle, we re-organize the above candidates by their preferences as in 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 another candidate will be selected. To implement Eq. (7) with length principle, we modify the 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 : $(q-p+1) \geq K$

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

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 a summation of scores of a document generated from each evidence ($\sum score_i$), and the second method is *NORM-SUM*. Let $norm_i$ (corresponds to Max_Norm [4]) 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 a short query and *NORM-SUM* for a long query because this selection was robust empirically.

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. Since 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 [9] when lengths of all documents

	# of translation pairs	*# of source language terms*	*# dictionary ambiguity*
J-K	434,672	399,220	1.09

Table 3. Bilingual dictionaries

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-6, we did not submit such combinations of QT and Pirkola's method. Instead, as in the previous NTCIR-5, we only performed naive query translation (NQT) focusing on combining multiple evidences which are generated from different term extractions. We believe that if this result is combined with Pirkola's method, then the performance can be further improved.

3.1 Bilingual Dictionary

Table 3 shows some statistics on our bilingual dictionaries used at NTCIR-6 CLIR. These dictionaries were extracted from dictionaries created for machine translation (MT) systems. Note that the ambiguity of J-K is very small. The first reason is the linguistic difference of characters used in two languages. Chinese character, which is frequently used in Japanese, is less-ambiguous than Korean character. In Korean language, several different Chinese characters can be equally pronounced by a single Korean character. Generally, when the source language is Korean (K-J or K-C), the ambiguity is much more in J-K. The second reason is due to the large ratio of proper nouns in dictionary, in which more than half of all words belong to proper nouns. In this case, there is little ambiguity. Without proper nouns, the ambiguity will increase.

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 other disambiguation procedures and is normally used as the baseline in BLIR research. Nevertheless, this method provides fundamental retrieval performance due to the effects of self-disambiguation, which 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 the topically related terms.

3.3 Combination of Multiple Evidences

As in the monolingual retrieval, there are multiple query representations for cross-lingual retrieval, which are merged to generate the final ranked result. Their representations are dependent on the 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 which are used in Korean.

A problem exists since we can only prepare the dictionary-based words by translating the given query. Other representations such as collection-based segment cannot be obtained by using direct translation due to the lack of bilingual dictionary. To build other representations, we first translate the original source word, and segment each translated word to generate consistent indexing terms according to corresponding extraction methods. The segmentation is performed by regarding all indexing terms by words in a dictionary. For example, the collection-based segment is obtained by decomposing the initial dictionary-based target term into smaller segments based on a statistical dictionary in the collection (Section 2.3.1). As a result, these segments become consistent to retrieve indexes of collection-based segments in Korean. Similarly, we can generate consistent translated terms for character bi-gram from dictionary-based translated words.

4 Experimentation

This section reports the retrieval results of our official runs submitted to NTCIR-6 CLIR task: three results of NTCIR-3, NTCIR-4 and NTCIR-5 track. Evaluation measure is the mean of non-interpolated average precision (MAP). Each topic has four fields: title (T), description (D), narrative (N) and concepts (C). Relevance judgments with relax version are used.

In Korean SLIR, we use Jelinek smoothing for language modeling approach of which parameter λ is 0.75 [15]. 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. The total number of expansion terms is restricted to 200. κ is set to 0.7.

In Japanese SLIR, remind that we did not combine multiple term extractions for Japanese task. Instead, we use a single term representation by extracting terms where only unknown words and nouns tagged by Chasen are considered and all English words are ignored. In addition, our retrieval method for Japanese is different from the architecture described in Figure 1. Basically, it follows the pure language

NTCIR-3			
Method	T	D	TDNC
BG	0.3068	0.2651	0.3811
DW	0.2750	0.2341	0.3780
CS	0.2785	0.2153	0.3819
BGp	0.3504	0.3445	0.4381
DWp	0.3939	0.3332	0.4520
CSp	0.3820	0.3241	0.4467
BGp+DWp+CSp	**0.4325**	0.3975	0.4853
Top	0.4325	**0.4116**	**0.5037**
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	0.5347	0.5170	0.5782
DWp	0.5094	0.4809	0.5453
CSp	0.5246	0.5248	0.5664
BGp+DWp+CSp	**0.5736**	**0.5571**	**0.6063**
Top	0.5736	0.5571	0.6063
NTCIR-5			
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	0.4855	0.5165	0.5777
DWp	0.5126	0.5325	0.5729
CSp	0.5392	0.5660	0.6085
BGp+DWp+CSp	0.5434	0.5725	0.6159
DWp+CSp	0.5539	**0.5829**	**0.6120**
Top	**0.5622**	0.5829	0.6120

Table 4. Official results of Korean SLIR at NTCIR-3, NTCIR-4 and NTCIR-5

modeling framework. For initial retrieval, The language modeling approach is first applied based on Jelinek-Mercer smoothing and then model-based feedback is performed regardless of the type of query. The smoothing parameter λ is fixed to 0.1. For the feedback, *R* is set to 13 for T, to 7 for D, and to 3 for DN. For expansion terms, all terms in feedback documents are included. κ is set to 0.0.

4.1 SLIR Track in Stage 2

Table 4 shows the official results of Korean retrieval on NTCIR-3, NTCIR-4 and NTCIR-5 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, then symbol "p" is attached to the 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. BGp and DWp indicate similar meanings. Bold face indicates that the run has achieved the best performance at the given task. N/A means that the retrieval result is not available at current status.

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 the 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 9.8% ((0.4325 - 0.3939) / 0.3939) and 15.4% ((0.3975 - 0.3445) / 0.3445) in T and D, respectively, and 8.6% ((0.4853 - 0.4457) / 0.4457) for TDNC. This final result is top-ranked on T at this year.

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 final result is top-ranked for all topics (T, D and Other) at this year

At NTCIR-5, BG completely fails on short length query, which is a different behavior from NTCIR-3 and NTCIR-4. Thus, the full combination method does not obtain synergy effects, of which performances are almost the same to CSp. Due to the failure of BG, we only submitted combining results of DWp and CSp without BGp. This combination method (DWp+CSp) shows better performances on triple combination (BGp+DWp+CSp) on T and D. This final result (DWp+CSp) is top-ranked for two topics (D and Others) at this year.

Different from previous NTCIR, note that there is a minor change on the setting of interpolating parameter α. The previous NTCIR system fixes α to 0.9, however, we found that the more α the final query model uses, the more retrieval performance we obtain. This year, α is modified into the value between 0.95 and 0.99. As a result, the final performance is slightly further improved from 1% to 2% for all test collections.

Table 5 shows the official results on Japanese retrieval results on NTCIR-3, NTCIR-4 and NTCIR-5 test set. We use notation CHA for Japanese extraction method. Overall, the performance of our system is not good, which is inferior to one of top system. Our final result is middle-ranked at this year.

NTCIR-3			
Method	T	D	DN
CHA	0.3105	0.3272	0.3926
CHAp	0.3848	0.3506	0.3808
Top	**0.4651**	**0.4707**	**0.4762**
NTCIR-4			
Method	T	D	DN
CHA	0.3296	0.3394	0.4223
CHAp	0.4281	0.4052	0.4134
Top	**0.5069**	**0.5082**	**0.4955**
NTCIR-5			
Method	T	D	DN
CHA	0.3022	0.3052	0.472
CHAp	0.4475	0.4118	0.4822
Top	**0.5259**	**0.4961**	**0.5380**

Table 5. Official results in Japanese SLIR at NTCIR-3, NTCIR-4 and NTCIR-5

NTCIR-3			
Method	T	D	TDNC
BG	0.1964	0.2056	0.2461
DW	0.1769	0.1910	0.2467
CS	0.1263	0.1233	0.1736
BGp	0.3016	0.2744	0.3007
DWp	0.2959	0.2985	0.3377
CSp	0.1664	0.1950	0.2529
BGp+DWp+CSp	0.3357	0.3212	0.3544
Top	**0.3725**	**0.3940**	**0.5037**
NTCIR-4			
Method	T	D	TDNC
BG	0.3119	0.3127	0.4064
DW	0.3040	0.2923	0.3961
CS	0.3193	0.3400	0.4446
BGp	0.4177	0.3720	0.4427
DWp	0.4021	0.3863	0.4539
CSp	0.4044	0.4273	0.4930
BGp+DWp+CSp	**0.4584**	**0.4345**	**0.5150**
Top	0.4584	0.4345	0.5150
NTCIR-5			
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
DWp+CSp	0.4722	0.5020	0.5572
Top	**0.5441**	**0.5571**	**0.5799**

Table 6. Official results in Korean BLIR at NTCIR-3, NTCIR-4 and NTCIR-5

Coll	Average of AvgPr	% SLIR
NTCIR-3	0.3371	76.89%
NTCIR-4	0.4673	81.05%
NTCIR-5	0.5105	87.57%

Table 7. Averages of AvgPr and performance ratios for corresponding K-K run of each J-K run

Method	T	D	TDNC
BG	0.4062	0.3849	0.5065
BGp	**0.5179**	0.5234	**0.5883**
Top	0.5179	**0.5375**	0.5883

Table 8. Official results in Korean SLIR at NTCIR-6 Stage 1

4.2 BLIR Track in Stage 2

Table 6 shows the official J-K retrieval results on NTCIR-3, NTCIR-4 and NTCIR-5. Since the target language is Korean, BG, DW and CS methods are available. Overall, the differences in performances according to each term extraction are almost the same as the results of K-K monolingual as mentioned in Section 4.1. For example, as like the monolingual retrieval, at NTCIR-5, BG fails on retrieval performance. Similar to the monolingual result, BG produces negative effects on retrieval performance when it is combined. There is an error in CS result in NTCIR-3, where the performance is failed. In fact, we found that there is a bug when applying CS to NTCIR-3. If the bug is fixed, then the performance of CS could be reasonably modified.

Table 7 shows the distribution of averages of AvgPr across different combinations of query fields and performance ratio of J-K for corresponding SLIR (K-K). The ratios are collection-dependent ranging from 70% to 90%. This result is not poor compared to SLIR, regarding that our system adopts not a sophisticated method but a naive translation method.

4.3 SLIR Track in Stage 1

For Korean in Stage 1, we do not combine multiple evidences of term extractions. Instead, we only use the BG method to extract terms and apply the pseudo relevance feedback as mentioned in Section 2. Note that we apply not the expansion-based feedback but the model-based feedback for long length query. For Japanese, the method used in Stage 1 is the same as one in Stage 2. Table 8 and Table 9 show the official results of NTCIR-6 Stage 1 for Korean and Japanese, respectively. Remark that for Korean SLIR, our result shows the best performance on T and TDNC.

Method	T	D	DN
CHA	0.2566	0.2505	0.3128
CHAp	0.3451	0.3151	0.3368
Top	**0.4393**	**0.4138**	**0.3898**

Table 9. Official results in Japanese SLIR at NTCIR-6 Stage 1

5 Conclusion

For NTCIR-6 SLIR, we employed a coupling strategy that combines several ranked lists generated from multiple term representations by differentiating pseudo relevance feedback and combination method according to the length of queries. We use three term extractions which consist of character n-gram and dictionary-based word and collection-based segment indexes for Korean retrieval. For NTCIR-6 BLIR, we experimented with a strategy based on a naive query translation and the same coupling strategies as target language. Remarkable observation is that collection-based segment by using unsupervised segmentation algorithm works well in all previous NTCIR tasks. In the future, we will 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.

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] J.-H. Lee. Combining multiple evidence from different properties of weighting schemes. In *SIGIR '95*, pages 180–188, 1995.

[5] J. H. Lee and J. S. Ahn. Using n-grams for korean text retrieval. In *SIGIR '96*, pages 216–224, 1996.

[6] 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.

[7] S.-H. Na, I.-S. Kang, and J.-H. Lee. Postech at ntcir-5: Combining evidences of multiple term extractions for mono-lingual and cross-lingual retrieval in korean and japanese. In *NTCIR-5: Working Notes of the Fifth NTCIR Workshop Meeting*, pages 1–8, 2005.

[8] 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.

[9] A. Pirkola. The effects of query structure and dictionary setups in dictionary-based cross-language information retrieval. In *SIGIR '98*, pages 55–63, 1998.

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

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

[12] S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR '94*, pages 232–241, 1994.

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

[14] C. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *CIKM '01*, pages 403–410, 2001.

[15] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR '01*, pages 334–342, 2001.