JustSystems in Japanese monolingual information retrieval at NTCIR-6

Tetsuya Tashiro
JustSystems Corporation
Aoyama Bldg. 1-2-3 Kita-Aoyama, Minato-ku,
Tokyo 107-8640, Japan
tetsuya_tashiro@justsystem.co.jp

Abstract

At the NTCIR-6 workshop, JustSystems Corporation participated in the Cross-Language Retrieval Task(CLIR). We submitted results to the track of monolingual information retrieval (Japanese to Japanese). The major goal of our participation is to evaluate performance and robustness of phrasal indexing and phrase down weighting combined with Language Modeling retrieval model.

Keywords: *Information Retrieval, Language Modeling, Phrasal Indexing*

1 Introduction

At the NTCIR-6 workshop, JustSystems Corporation participated in the Cross-Language Retrieval Task (CLIR). The major goal of our participation is to evaluate performance and robustness of phrasal indexing and phrase down-weighting combined with Language Modeling retrieval model.

Language Modeling retrieval model has become as standard and popular as Okapi BM25 in evaluation workshops like TREC and NTCIR. JustSystems Corporation and Clairvoyance Corporation examined the effectiveness of phrasal indexing and phrase down-weighting approaches combined with Okapi BM25 in Cross-Lingual Information Retrieval at NTCIR-4.

At the NTCIR-6 workshop, we developed a straightforward combination of Language Modeling retrieval model and phrasal indexing and phrase down-weighting approaches. We also built the system based on Okapi BM25 and phrasal indexing and phrase down-weighting approaches.

At the first stage of CLIR task, we submitted the result from the system based on Okapi BM25 retrieval model.

We did comparable study of Language Modeling and BM25 with the test collections used at the second stage. This means our explorations in Language Modeling retrieval model were done with the test collections from NTCIR-3, NTCIR-4 and NTCIR-5.

2 System description

In this section, we describe an outline of our retrieval system.

2.1 Bag-of-Words model

The bag-of-words model is probably the most widely used for modeling documents in information retrieval. In this model, each document is represented as a feature vector counting the number of occurrences of different words as features and the positional and ordinal information of word occurrences is ignored. Our system employs this model.

2.2 Indexing term

Our system handles individual noun words, noun phrases (sequence of noun words) and attested subphrases as document features. An attested subphrase is constituent of a longer noun phrase that also appears independently as a full noun phrase elsewhere in the document collection. The effectiveness of the phrasal indexing was examined in the past research [2].

2.3 Feature extraction

Our system uses natural language processing methods to extract noun phrases from documents.

We employed our internally developed morphological analyzer called JPOT (Java Part-Of-Speech Tagger) for tokenization and part-of-speech tagging. Built upon statistical bigram models, JPOT can process many types of languages such as Japanese, Chinese, English, French, and Spanish.

For noun phrase identification, we apply finite state machine based grammar to the result of morphological analysis. During this process, several types of normalization are performed such as numeric normalization, normalization of long vowel markers in Katakana characters and dictionary-based normalization.

Example of normalization
Numeric normalization {二百万 = 200万 = 2,000,000}
Long vowel marker normalization { コンピューター = コンピュータ }
Dictionary-based normalization { プリインストール = プレインストール }

After noun phrase extraction is performed, the number of different feature occurrences is counted. These statistics are used in each machine learning framework.

2.4 Stop words

At the runs on the *description(D)* field,we eliminated very frequent and non-informative terms from query term vectors, such as "記事"(article), "検索"(retrieval) or "内容"(content).

2.5 Index Implementation

We developed the indexing program and searching program for the BM25 retrieval model and Language modeling retrieval model based on Apache Lucene.

3 Retrieval Model

In this section, we describe retrieval models used for this task.

3.1 BM25TF*IDF

The vector space similarity between a given document d and a given query q is used to score candidate documents [3], and goes as

$$sim(q,d)=\sum_{w\in q\cap d} W_q(w)\,W_d(w)$$

where $w\in q\cap d$ is either a term of phrase found in both *q* and *d*.

$W_q(w)$ is the weight associated with *w* in *q*, and goes as

$$\omega_w\,\delta\,TF_q(w)\,IDF(w)$$

$W_d(w)$ is the weight associated with *w* in *d*, and goes as

$$TF_d(w)\,IDF(w)$$

where ω_w is the term weight of *w* given by pseudo relevance feedback if used (always 1.0 if no pseudo relevant feedback is employed), δ is a parameter used to change the weight of phrases (1.0 for non-phrase terms), and $IDF(w)$ is the standard inverse document frequency).

$TF_d(w)$ is Okapi BM25 TF, and goes as

$$\frac{(k_1+1)*c(w;d)}{k_1[(1-b)+b*(|d|/\Delta)]+c(w;d)}$$

where k_1 is the term frequency smoothing parameter, *b* is the document length smoothing parameter, $|d|$ is the document length, and Δ is the average document length in the corpus.

$c(w;d)$ is the number of occurrences of *w* in *d*.

3.2 Language Modeling

The likelihood that a given document *d* will generate a given query *q* is used to score candidate documents [1], and is given by

$$p(q|d)=\sum_{w\in q}\omega_w\log\frac{c(w;d)+\mu\,\delta\,p(w|C)}{\Sigma_w c(w;d)+\mu}$$

where $w\in q$ is either a term or phrase found in *q*.

ω_w is the term weight of *w* given by pseudo relevance feedback(always 1.0 if no pseudo relevant feedback is employed).

$c(w;d)$ is the number of occurrences of *w* in *d*.

μ is the Dirichlet prior smoothing parameter.

δ is a parameter used to change the weight of phrases(always 1.0 for non-phrase terms).

$p(w|C)$ is the term count of *w* in the corpus divided by the corpus size, and $\Sigma_w c(w;d)$ is the length of document *d* not including stop-words.

3.3 Query Expansion

Query expansion through pseudo relevance feedback has proved to be effective for improving retrieval performance. We used pseudo-relevance feedback for augmenting the queries. After retrieving some documents for a given topic from the target corpus, we took a set of top ranked documents, regarding them as relevant documents to the query, and extracted terms from the these documents. We use a formulate called Prob2 for extracting and ranking terms for expansion.

$$Prob2(t)=\log(R_t+1)\times[\log(\frac{N-R+2}{N_t-R_t+1}-1)-\log(\frac{R+1}{R}-1)]$$

where N is the number of documents in the reference corpus, Nt is the number of documents that contain the term t in the corpus, R is the number of the top n retrieved documents, and Rt is the number of documents that contain the term t in the top n documents. The k terms with the highest score according to this measure are selected and merged with original query to create the final expanded query.

$$Q_{new} = Q_{orig} + Q_{\exp}$$

Qnew, *Qorig*, *Qexp* stand for the new expanded query, the original query, and terms extracted for expansion, respectively.

Each term in *Qorig* has its original term weight ω_{orig}. (ω_{orig} is always 1.0).

$c(w;d)$ is incremented by 1 and term weight ω_{new} is assigned according to the following weighting scheme in the query expansion process.

$$\omega_{new} = \omega_{orig} + \frac{\omega_{Prob2}(t)}{\Sigma\, \omega_{Prob2}(t)}$$

Across all the experiments, we used same parameter setting for term expansion. Top 10 documents in initial retrieval are used as seed documents. We extracted the terms which appear in two documents at least and ranked the terms according to Prob2 method. We used top 35 terms for query expansion.

4 Experiments

We have submitted the results both in the first stage and second stage of formal run. We also did comparable study of Language Modeling and BM25 with the test collections used at the second stage.

In this section, we describe the result of our experiments.

4.1 First Stage - Usual Ad Hoc Searches -

At the first stage, we employed BM25TF*IDF retrieval model and submitted four runs. Table 1 shows the profiles of the submitted runs.

Parameter setting for BM25TF is like below.

Term frequency smoothing parameter $k_1 = 1.2$
Document length smoothing parameter $b = 0.2$
Phrase down-weighting parameter $\delta = 0.2$

Run id	Field	Term Expansion
JSCCL-J-J-T-01	Title	On
JSCCL-J-J-D-02	Description	On
JSCCL-J-J-T-03	Title	Off
JSCCL-J-J-D-04	Description	Off

Table 1. Submitted Runs at the first stage

Table 2 shows the average precision of the first stage. In the title runs, query expansion improves average precision.

In the description runs, the result with query expansion (JSCCL-J-J-D-02) underperformed the result without query expansion. This is caused by the error in our submitting process.

Run id	Rigid	Relax
JSCCL-J-J-T-01	0.2983	0.3821
JSCCL-J-J-D-02	0.1981	0.2554
JSCCL-J-J-T-03	0.2647	0.3485
JSCCL-J-J-D-04	0.2450	0.3206

Table 2. Result of the first stage (average precision)

4.2 Second Stage - Cross Collection Analysis -

At the second stage, we employed both BM25TF*IDF retrieval model and Language Modeling retrieval model. We submitted two runs for each retrieval model. We employed query expansion for all the runs. Table 3 shows the profiles of the submitted runs.

Run id	Field	Retrieval Model
JSCCL-J-J-T-01-{N3-N5}	Title	BM25
JSCCL-J-J-D-02-{N3-N5}	Description	BM25
JSCCL-J-J-T-03-{N3-N5}	Title	LM
JSCCL-J-J-D-04-{N3-N5}	Description	LM

Table 3. Submitted Runs at the second stage

Parameter setting for BM25TF is like below.

Term frequency smoothing parameter $k_1 = 1.2$
Document length smoothing parameter $b = 0.2$
Phrase down-weighting parameter $\delta = 0.2$

Parameter setting for Language Modeling is like below.

Dirichlet prior smoothing parameter $\mu = 1000$
Phrase down-weighting parameter $\delta = 2000$

4.3 Parameter Analysis: BM25TF*IDF

Figure 1 shows the results of experiments where p*hrase down-weighting parameter* δ was varied and the other two kept fixed. This estimations came from preliminary experiments where many combinations of input parameters were changed (i.e. none were fixed) in order to get a rough idea of where the true optimized values might be. Results for all three corpora are displayed both with and without PRF. The query type for these results is always title, and the evaluation method always relaxed. The *y*-axis always represents the mean average precision.

It is clear that the use of pseudo relevance feedback improves results in essentially all cases. However, one of the first things one notices about these curves is that the use of PRF generates a lot of unpredictability and jaggedness compared to trials without it. This may be attributed to the straying of the topic when many more terms are added to the query. Overall more relevant documents are returned, but the range of topics generally increases and makes for a less predictable MAP.

As for δ, MAP peaks are at (0.1, 0.4179), (0.25, 0.4100), and (0.05, 0.3956) without PRF. With PRF, they can be found at (0.15, 0.4705), (.45, 0.4684), and (.05, 0.4801). PRF renders the curves less smooth, with N5 suffering the most and almost taking on sinusoidal characteristics. One of the most interesting aspects of this graph is that N4 is nearly flat for all values of δ, the phrase discount parameter. As for query set statistics, one potential anomaly of N4's title query set is its high term to phrase ratio of 4.08. N3 and N5 have comparatively lower ratios of 3.39 and 3.63. Thus, with fewer phrases on the average, it makes some sense that N4 would be less affected by the phrase discount parameter.

4.4 Parameter Analysis: Language Modeling

Figure 2 shows the results of experiments where *phrase down-weighting parameter* δ was varied and *Dirichlet prior smoothing parameter* μ kept fixed.

As with Okapi, use of PRF increased MAP significantly along with increasing the roughness of the curves. Phrase down-weighting parameter δ to LM tend to be far more stable than those of Okapi.

(δ, MAP) peak values occur at (5000, 0.4090), (100, 0.3944), and (1000, 0.3831) without PRF and (100, 0.4734), (100, 0.4534), and (4000, 0.4575) with it. However, to call some of these points peaks would be misleading. LM's δ is the most stable parameter in this entire project, and as long as it exceeds some threshold value (about 3000 without PRF and 5000 with it), MAP values hardly change at all. The tiny "peaks" that arise at various points can probably be attributed to experimental error and the presence of human generated data. Anomalies such as N3's seemingly sharp peak at δ of 100 are probably fortuitous points that might also be found on other curves for less round values of δ. In addition, just like in the Okapi case, N4 is almost unaffected by δ, though this δ plays a slightly different role in LM.

Reducing the weight of phrasal terms in the corpora greatly improves performance [2]. We achieved this through our δ parameter in Okapi, and achieved ideal MAP values for δ somewhere between 0.05 and 0.2, which was to be expected. However, with LM the ideal range on δ is extremely different, around 1000 or above. Actually, the δ in LM does not perform exactly the same role as it does in Okapi. If it were multiplied to each term in the summation in the LM formula as a whole (i.e. alongside ω_w), it turns out that the ranking of result documents would not change. Each score would get a uniform boost or discount, and the MAP would be the same. Therefore, δ in LM was moved so that it was only multiplied with μ in the numerator. Here it is able to improve the MAP, but must be somewhere on the order of 1000 due to the interaction with μ, which also achieves best performance around this order of magnitude.

5 Conclusion

We did comparable study of BM25 and Language Modeling combined with phrasal indexing and phrase down-weighting. BM25 outperformed the particular implementation of language modeling we tested in this project. Admittedly, the δ parameter was inserted a bit haphazardly into the LM algorithm, though it did improve performance significantly. Further adjustments to LM could quite possibly push it beyond what Okapi is capable of. These possible improvements should be explored in future experiments.

References

[1] Zhai, C., Lafferty, J. 2001. A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval. From *SIGIR '01*.

[2] Fujita, S. 1999. Notes on Phrasal Indexing: JSCB Evaluation Experiments at NTCIR AD HOC. In Proceedings of the First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition.

[3] Qu, Y., Grefenstette, G., Hull, D., Evans, D., Ishikawa, M., Nara, S., Ueda, T., Noda, D., Arita K., Funakoshi, Y., Matsuda, H. Justsystem-Clairvoyance CLIR Experiments at NTCIR-4 Workshop. In Working Notes of the NTCIR-4 Workshop.

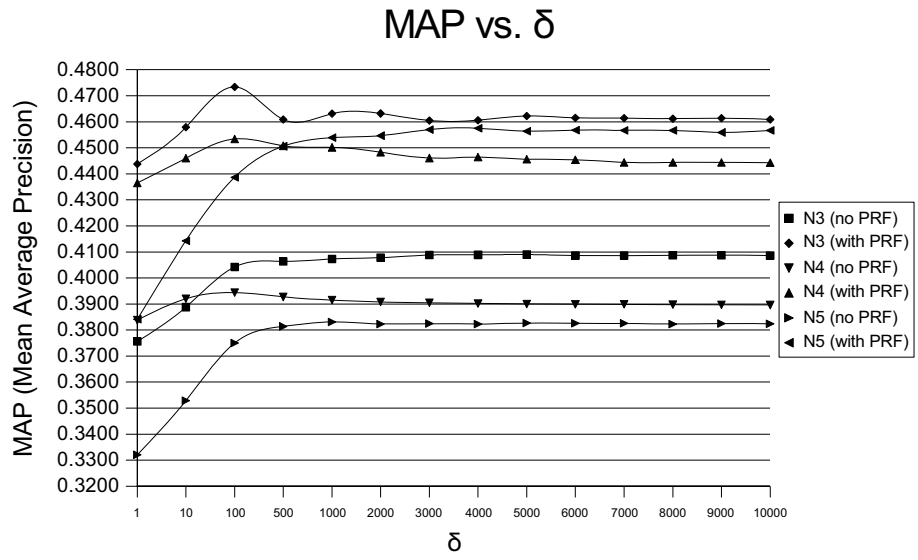

Figure 1: Okapi's Phrase Discounting Parameter, δ
(Fixed Parameters: k1=1.1, b=0.2)

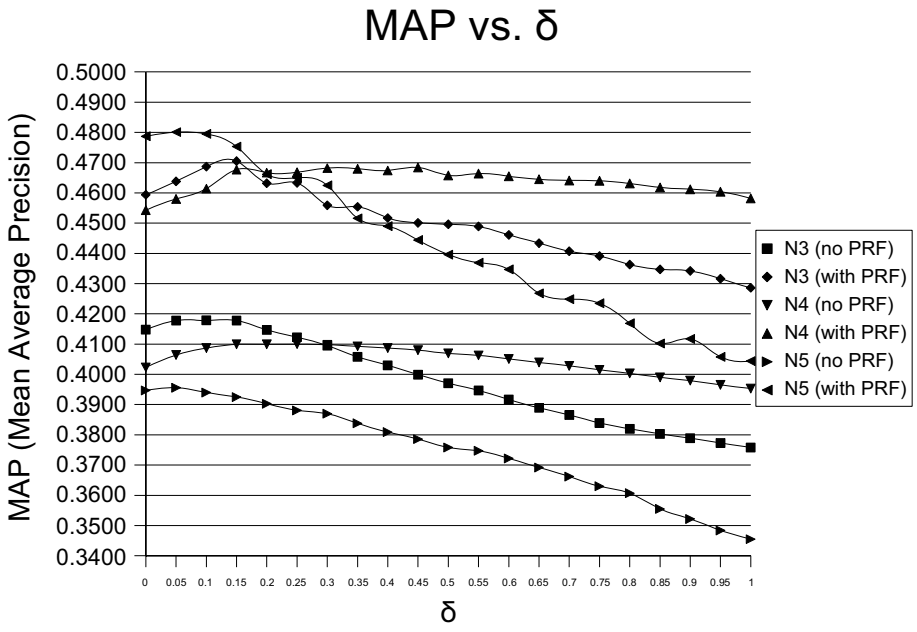

Figure 2: LM's Phrase Multiplier Parameter, δ
(Fixed Parameter: μ=1700)