Applying Data Fusion to IR4QA in Japanese using Word Search and Character-based N-gram Search

Michiko Yasukawa — Gunma University
J. Shane Culpepper — RMIT University
Falk Scholer — RMIT University
michi@gunma-u.ac.jp, {shane.culpepper, falk.scholer}@rmit.edu.au

Abstract

This paper presents discussion of data fusion methods using word search and character-based n-gram search in Japanese. We show how to improve search effectiveness using data fusion on the NTCIR IR4QA task. For the experimental evaluation, we use five important Japanese linguistic tools and 21 state-of-the-art search models. Experimental results demonstrate that the combination of the two different approaches can reliably improve search effectiveness.

1 Introduction

Data fusion technique in information retrieval is a sort of meta-search [1]. We apply data fusion methods to answer the following question: If there are two ranked lists of documents from word search and character-based n-gram search in Japanese, is a better list obtainable by combining them?

Two searches are used because of the nature of the Japanese language. Different from English texts, Japanese texts include no word segment markers such as white spaces. Therefore, identification of words and index terms in the search system are primary issues for Japanese information retrieval. To collect index terms from documents, Japanese morphological analyzers are used for the word-breaking process. However, morphological analysis often presents uncertainty when processing words that are not listed in the dictionary, such as unknown words or compound words. We consider "大仏像" as one example sequence of Japanese characters. Even a knowledgeable native speaker of Japanese would not recognize the best word-segment for this example because it might be "大仏, 像" (Great Buddha, statue) or " 大, 仏像" (huge, Buddha statue) according to the meaning of the words in documents or search topics. Different Japanese morphological analyzers may produce different word segments for such ambiguous character sequences. In contrast, character-based n-gram does not confront this ambiguity because the two meaningful bi-grams, " 大仏" (Great Buddha) and "仏像" (Buddha statue) are collected on a definitive manner. Both terms are indexed.

The following sections present discussion of the application of data fusion methods to the two searches, and discuss our experimental methodology and improvements in search effectiveness.

2 Related Work

Data fusion produces aggregate searched document lists using different search models and linguistic processes. Here, linguistic processes include stemming, word-breaking, and morphological analysis. Search models include the standard vector space model, the cutting-edge language model, and many others. Several data fusion approaches have been used in English search tasks in the TREC collections [2] [3], and Asian search tasks in the NTCIR collections [4]. For data fusion methods in English, significant improvements have been reported [3]. However, small improvements for the Japanese search task have been reported [4]. Our contribution is to demonstrate significant improvements using data fusion methods for Japanese with the most up-to-date linguistic tools and state-of-the-art search models that have not been tested in earlier works.

For the IR4QA task in NTCIR7/8 [5][6], which is a simple ad-hoc search task in Japanese, some participant groups discussed character-based n-gram search and word search [7] [8] [9]. In fact, word search was effective. Different from those approaches, we use both searches to surpass the baseline word search. The data fusion methods applied to our study are described in the next section.

3 Data Fusion Methods

For simplicity of explanation, we refer to a ranked list of documents as **run**. In addition, character-based n-gram is simply referred to as ***n*-gram**. To combine runs, there are widely various possible methods [1]. The work reported by Wu and others [3] demonstrates the effectiveness of seminal data fusion methods proposed by Fox and Shaw [2]. The data fusion methods used in the work [3] are applicable to our approach to infer a relevance score (hereinafter, **RS**) to a document in the data fusion run using the RSs in the two runs to be merged. The five methods applied in our approach calculate the combined RS in the following ways:

1. CombMIN: Minimum of RSs to the document.
2. CombMAX: Maximum of RSs to the document.
3. CombANZ: Average of the non-zero RSs. It is calculated using CombSUM divided by the number of nonzero RSs. CombSUM is the summation of RSs to the document.

4. CombMNZ: CombSUM multiplied by the number of nonzero RSs. It raises the rank of a document in both runs.
5. Linear: The weighted summation of RSs to the document. It is calculated using the following equation.

For any searched document d_i, its combined linear relevance score $\mathrm{RS}_c(d_i)$ in the merged run is a weighted linear combination of its original $\mathrm{RS}_w(d_i)$ from the word-search and $\mathrm{RS}_n(d_i)$ from n-gram search

$$\mathrm{RS}_c(d_i) = \alpha * \mathrm{RS}_w(d_i) + (1 - \alpha) * \mathrm{RS}_n(d_i), \quad (1)$$

where α is the weight given to the word-search run.

An earlier work [3] demonstrates a method called "LNorm," which is effective to adjust score scales from different search models. It is a linear normalization method using the maximum and minimum document relevance score of a run. The normalized relevance score, NRS for a retrieved document d is calculated as

$$\mathrm{NRS}_d = (\mathrm{RS}_d - \mathrm{MIN})/(\mathrm{MAX} - \mathrm{MIN}), \quad (2)$$

where MAX and MIN represent the maximum and minimum RS of a run. Because the methods without the normalization are ineffective for our study, we will later report the experimentally obtained results of the methods with "LNorm." The methods to be reported are LNorm CombMIN, LNorm CombMAX, LNorm CombANZ, LNorm CombMNZ, and LNorm Linear.

4 Experimental Methodology

For this experiment, we use the NTCIR7/8 IR4QA test collections in Japanese[1]. They consist of 797,700 documents from Mainichi News Paper 1998–2005, the search topics, and the judgment files. To obtain a data fusion run, we must prepare word and n-gram search single runs to be combined. Then, we obtain a data fusion run using the methods described in Section 3. To perform the initial word and n-gram search, we use Terrier IR Platform[2] ver. 3.5 and its implemented search models (BB2, BM25, DFI0, DFR_BM25, DFRee, DirichletLM, DLH, DLH13, DPH, Hiemstra_LM, IFB2, In_expB2, In_expC2, InB2, InL2, Js_KLs, LemurTF_IDF, LGD, PL2, TF_IDF, and XSqrA_M) to calculate relevance scores. For the word-breaking process in word search, we use five Japanese linguistic tools: ChaSen [3], Juman [4], KaKaSi [5], KyTea [6], and MeCab [7].

The best five MAP values for the initial word and n-gram search are shown in Table 1. The abbreviations for search models in Table 1 are explained in Table 2. As Table 1 shows, word search tends to produce better results than n-gram search when the same search model is chosen. Among the n-gram search, 2-gram search yields more effective results than 1-gram or 3-gram search does. Regarding the search models, DirichletLM is the most effective. For word search and 2-gram search, XSqrA_M is the second-best. The model, DirichletLM [10] is a Language Model. It is different from the model XSqrA_M [11], which is a Divergence from Randomness (DFR) model. Among the word searches with DirichletLM, Juman and KyTea respectively produce better results for NTCIR7 and NTCIR8.

In comparison of each average precision (AP) value per topic, word search is generally more effective than n-gram search. However, n-gram search wins against word search for some search topics. Moreover, it is non-negligible. Put more specifically, for the AP values on the 98 topics in the NTCIR7 collection with the model, DirichletLM, the word search with Juman wins against the n-gram search with 2-gram by 70 topics. The word search loses to the n-gram search by 25 topics. The two searches equally match for 3 topics. For the AP values on the 94 topics in the NTCIR8 collection with the model, DirichletLM, the word search with KyTea wins against n-gram search with 2-gram by 67 topics, and the word search loses to the n-gram search by 27 topics. Because word search and n-gram search are good at different topics, the merged runs from them are expected to be effective because of the harmonic effect of data fusion.

For the baseline search in the experiment, we use the top performing search model, DirichletLM. First, we use each data fusion method with the word and n-gram single runs with DirichletLM to identify the most effective fusion method. Second, we exploit the most effective method to obtain combined runs with the model, XSqrA_M. After single runs are tested with DirichletLM and XSqrA_M, we also investigate whether data fusion with Query Expansion (QE) is effective with these two models. To obtain QE data fusion runs, we use Terrier's default automatic query expansion model, Bo1 (a Bose–Einstein distribution query-expansion method [11]). Finally, to compare the overall effectiveness, we also test other data fusion runs with every other search model aside from DirichletLM and XSqrA_M, with and without QE, respectively.

5 Results and Discussion

The mean average precision (MAP) values for data fusion runs are shown in Table 3. The MAP values in the first row (NTCIR7/8-01) are the baseline word or n-gram single run in the corresponding column. Each fusion run in the rows from NTCIR7/8-02 through NTCIR7/8-06 is a combined run with a word search run and a 2-gram search run chosen from the corresponding word/n-gram search single runs (NTCIR7/8-01). The DirichletLM search model is used for all runs (from NTCIR7/8-01 through NTCIR7/8-06). Overall, few significant improvements are apparent among the data fusion runs with the four combination methods (from NTCIR7/8-02 through 05). The runs with the LNorm Linear method (NTCIR7/8-06) are effective. Their MAP values are better than the baseline runs (NTCIR7/8-01). However, the differences

[1] http://research.nii.ac.jp/ntcir/data/data-en.html
[2] http://terrier.org
[3] http://en.sourceforge.jp/projects/chasen-legacy/
[4] http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?JUMAN
[5] http://kakasi.namazu.org/index.html.en
[6] http://www.phontron.com/kytea/
[7] http://code.google.com/p/mecab/

Table 1: Effectiveness results based on mean average precision (MAP) for NTCIR7/8 IR4QA **word** and **ngram** search single runs. † and ‡ respectively denote statistical significance relative to the top run for each column at the 0.05 and 0.001 levels based on a 2-tailed paired t-test. Abbreviations for search models are presented in Table 2.

Dataset	**ChaSen**	**Juman**	**KaKaSi**	**KyTea**	**MeCab**	**1-gram**	**2-gram**	**3-gram**
NTCIR7	0.6346 Di	0.6769 Di	0.6590 Di	0.6888 Di	0.6667 Di	0.3151 IL	0.6220 Di	0.5954 Di
	0.6086† XS	0.6526† XS	0.6329‡ XS	0.6605‡ XS	0.6441† XS	0.3143 TF	0.6097† XS	0.5823 XS
	0.5987‡ Js	0.6419‡ DP	0.6256‡ DP	0.6485‡ DP	0.6326‡ DP	0.3131 IB	0.5996‡ FI	0.5762† DP
	0.5972‡ DP	0.6398‡ Js	0.6193‡ Js	0.6454‡ Js	0.6312‡ Js	0.3109 xB	0.5992† Js	0.5746† FI
	0.5942‡ FR	0.6347‡ FR	0.6146‡ FR	0.6405‡ FR	0.6272‡ FR	0.3051† xC	0.5979‡ DP	0.5707‡ Js
NTCIR8	0.4859 Di	0.5072 Di	0.4978 Di	0.5044 Di	0.4943 Di	0.2827 LG	0.4599 Di	0.4018 Di
	0.4533‡ XS	0.4764‡ XS	0.4696‡ XS	0.4724‡ XS	0.4606‡ XS	0.2812 XS	0.4405† XS	0.3943 XS
	0.4276‡ DP	0.4535† DP	0.4497† DP	0.4487† DP	0.4366† DP	0.2785 IB	0.4225‡ DP	0.3871† DP
	0.4175‡ Js	0.4398‡ Js	0.4346‡ Js	0.4335‡ Js	0.4218‡ Js	0.2708 TF	0.4211‡ FI	0.3703‡ Js
	0.4095‡ FR	0.4301‡ FR	0.4271‡ FR	0.4240‡ FR	0.4124‡ FR	0.2691 IL	0.4138‡ Js	0.3703‡ FI

are not statistically significant in the cases of Juman and KyTea. The data fusion runs with XSqrA_M (NTCIR7/8-08) are more effective than the corresponding single runs (NTCIR7/8-07). However, they are not significant improvements over the baseline runs (NTCIR7/8-01). Regarding the query expansion (QE) data fusion runs with DirichletLM (NTCIR7/8-09), they are worse for NTCIR7 test collection. The differences are not statistically significant for the NTCIR8 test collection. However, the QE XSqrA_M data fusion runs (NTCIR7/8-12) are significantly better than the baseline runs (NTCIR7/8-01) in all cases (ChaSen, Juman, KaKaSi, KyTea, and MeCab). To this point, we have specifically addressed the two best search models (DirichletLM and XSqrA_M), and the combination only with the 2-gram search runs. We have also confirmed MAP values for all combinations using other search models, the combination with runs from 1-gram to 3-gram search runs. Results obtained from those other data fusion runs are not significantly better than the best data fusion runs (NTCIR7/8-12).

In the NTCIR7 workshop [5], the MAP value of the run, OT-JA-JA-04-T reported by Tomlinson [7] was 0.6979. It was the best participant run. Our best data fusion runs for NTCIR7 (NTCIR7-12) with Juman, KyTea, and MeCab are superior, but the differences are not statistically significant. For the run OT-JA-JA-04-T, the participant group identified unwanted Japanese words such as “事件” meaning “accident” and “関連” meaning “relation” from the training topics. Those words were removed from their index to produce marked improvements in the task, but using such specific stop words might limit the search process generality. Our data fusion runs were obtained with no stop words. They still yielded MAP values comparable to those of the best group. In the NTCIR8 workshop [6], the MAP value of the run, LTI-JA-JA-01-T reported by Shima and Mitamura [9] was 0.4356. It was the best participant run. Our best data fusion runs for NTCIR8 (NTCIR8-12) produced statistically significant improvements over the best participant run[8].

Table 2: Weighting models for terms and documents.

Abbr.	**Model**	**Parameter**	**The best**
Di	DirichletLM	$mu = 2500$ (default)	1st
XS	XSqrA_M	Parameter free	2nd
DP	DPH	Parameter free	3rd or 4th
Js	Js_KLs	Parameter free	3rd or 4th
FR	DFRee	Parameter free	5th
FI	DFI0	Parameter free	OTHER
TF	TF_IDF	Parameter free	OTHER
IL	InL2	$c = 1.0$ (default)	OTHER
IB	InB2	$c = 1.0$ (default)	OTHER
xB	In_expB2	$c = 1.0$ (default)	OTHER
xC	In_expC2	$c = 1.0$ (default)	OTHER
LG	LGD	$c = 1.0$ (default)	OTHER

6 Conclusion

We investigated the effectiveness of data fusion methods on the ad-hoc search task, NTCIR IR4QA in Japanese. For the experiment, we used major Japanese linguistic tools including ChaSen, Juman, KaKaSi, KyTea, and MeCab, and cutting-edge search models including Dirichlet Language Model (DirichletLM) and Divergence from Randomness using Pearson’s chi-square (XSqrA_M).

Our experiment verified that word search is generally more effective than character-based n-gram search. However, the latter wins against word search in some cases. Moreover, it is non-trivial. When we merged the two searched document lists from the two searches, the merged list produced significant improvement by virtue of the effect of data fusion technique. This result implies that word-breaking in Japanese still has room for improvement. Presently used Japanese linguistic tools have not been able to manage the task independently.

Future work should be undertaken to identify what character sequences in Japanese must be collected as index terms for effective information retrieval.

[8] Differences were tested using a 2-tailed paired t-test, with p-value less than 0.001 used as a threshold for statistical significance

Table 3: Effectiveness results based on mean average precision (MAP) for NTCIR7/8 IR4QA data fusion runs. † and ‡ denote statistical significance relative to the baseline run for each column at the 0.05 and 0.001 levels, respectively, based on a 2-tailed paired t-test. **Boldface** denotes significant improvement. α is the parameter for LNorm Linear.

Search Type	Model	Method	ChaSen	Juman	KaKaSi	KyTea	MeCab	2-gram
NTCIR7-01	DirichletLM	Single run (baseline)	0.6346	0.6769	0.6590	0.6888	0.6667	0.6220
02	DirichletLM	LNorm CombMIN	0.5983†	0.6253†	0.6075†	0.6335†	0.6177†	
03	DirichletLM	LNorm CombMAX	0.6561	0.6755	0.6653	0.6780	0.6721	
04	DirichletLM	LNorm CombANZ	0.6413	0.6668	0.6564	0.6713	0.6613	
05	DirichletLM	LNorm CombMNZ	**0.6669**‡	0.6799	0.6718	0.6821	0.6753	
06	DirichletLM	LNorm Linear	**0.6688**†	0.6926	**0.6812**†	0.6981	**0.6849**†	
			$\alpha = 0.5$	$\alpha = 0.8$	$\alpha = 0.7$	$\alpha = 0.8$	$\alpha = 0.7$	
07	XSqrA_M	Single run	0.6086†	0.6526†	0.6329‡	0.6605‡	0.6441†	0.6097†
08	XSqrA_M	LNorm Linear	0.6402	0.6675	0.6523	0.6694	0.6588	
			$\alpha = 0.5$	$\alpha = 0.8$	$\alpha = 0.6$	$\alpha = 0.9$	$\alpha = 0.7$	
09	QE DirichletLM	Single run	0.5258‡	0.5505‡	0.5431‡	0.5741‡	0.5528‡	0.5957
10	QE XSqrA_M	Single run	**0.6615**†	0.6892	**0.6808**†	0.7072	**0.6936**†	**0.6356**†
11	QE DirichletLM	LNorm Linear	0.6246	0.6285†	0.6245	0.6338†	0.6267	
			$\alpha = 0.4$	$\alpha = 0.5$	$\alpha = 0.4$	$\alpha = 0.5$	$\alpha = 0.4$	
12	QE XSqrA_M	LNorm Linear	**0.6738**†	**0.7030**†	**0.6945**†	**0.7145**†	**0.7037**†	
			$\alpha = 0.8$	$\alpha = 0.8$	$\alpha = 0.7$	$\alpha = 0.9$	$\alpha = 0.8$	
NTCIR8-01	DirichletLM	Single run (baseline)	0.4859	0.5072	0.4978	0.5044	0.4943	0.4599
02	DirichletLM	LNorm CombMIN	0.4583†	0.4680†	0.4659†	0.4704†	0.4678†	
03	DirichletLM	LNorm CombMAX	0.4799	0.4900	0.4838	0.4922	0.4854	
04	DirichletLM	LNorm CombANZ	0.4836	0.4888†	0.4877	0.4955	0.4906	
05	DirichletLM	LNorm CombMNZ	0.4998	0.5046	0.5008	0.5042	0.5031	
06	DirichletLM	LNorm Linear	**0.5024**†	0.5136	**0.5063**‡	0.5122	**0.5094**†	
			$\alpha = 0.6$	$\alpha = 0.7$	$\alpha = 0.7$	$\alpha = 0.7$	$\alpha = 0.7$	
07	XSqrA_M	Single run	0.4533‡	0.4764‡	0.4696‡	0.4724‡	0.4606‡	0.4405†
08	XSqrA_M	LNorm Linear	0.4697	0.4812†	0.4769†	0.4892†	0.4729	
			$\alpha = 0.5$	$\alpha = 0.7$	$\alpha = 0.7$	$\alpha = 0.6$	$\alpha = 0.5$	
09	QE DirichletLM	Single run	0.4461†	0.4664†	0.4650	0.4663†	0.4558†	0.4753
10	QE XSqrA_M	Single run	**0.5158**†	**0.5411**†	**0.5362**‡	**0.5341**†	**0.5271**†	0.4793
11	QE DirichletLM	LNorm Linear	0.5153	0.5243	0.5227	0.5213	0.5173	
			$\alpha = 0.5$	$\alpha = 0.5$	$\alpha = 0.5$	$\alpha = 0.5$	$\alpha = 0.5$	
12	QE XSqrA_M	LNorm Linear	**0.5297**‡	**0.5437**†	**0.5427**‡	**0.5356**†	**0.5367**‡	
			$\alpha = 0.7$	$\alpha = 0.8$	$\alpha = 0.7$	$\alpha = 0.8$	$\alpha = 0.7$	

References

[1] S. Wu. *Data Fusion in Information Retrieval*. Springer, 2012.

[2] E.A. Fox and J.A. Shaw. Combination of multiple searches. In *The Second Text REtrieval Conference*, pages 243–252, 1994.

[3] M. Wu, D. Hawking, A. Turpin, and F. Scholer. Using anchor text for homepage and topic distillation search tasks. *JASIST*, 63(6):1235–1255, 2012.

[4] S. Abdou and J. Savoy. Monolingual experiments with Far-East languages in NTCIR-6. In *Proc. of the 6th NTCIR Workshop Meeting*, pages 52–59, 2007.

[5] T. Sakai, N. Kando, et al. Overview of the NTCIR-7 ACLIA IR4QA task. In *Proc. of the 7th NTCIR Workshop Meeting*, pages 77–114, 2008.

[6] T. Sakai, H. Shima, et al. Overview of NTCIR-8 ACLIA IR4QA. In *Proc. of the 8th NTCIR Workshop Meeting*, pages 63–93, 2010.

[7] S. Tomlinson. Experiments in finding Chinese and Japanese answer documents at NTCIR-7. In *Proc. of the 7th NTCIR Workshop Meeting*, pages 177–184, 2008.

[8] N. Lao, H. Shima, et al. Query expansion and machine translation for robust cross-lingual information retrieval. In *Proc. of the 7th NTCIR Workshop Meeting*, pages 140–147, 2008.

[9] T. Shima and T. Mitamura. Bootstrap pattern learning for open-domain CLQA. In *Proc. of the 8th NTCIR Workshop Meeting*, pages 37–42, 2010.

[10] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22(2):179–214, 2004.

[11] G. Amati. *Probability Models for Information Retrieval based on Divergence from Randomness*. PhD thesis, University of Glasgow, Glasgow, 2003.