Data Fusion for Japanese Term and Character N-gram Search

Michiko Yasukawa
Gunma University
Gunma, Japan
michi@gunma-u.ac.jp

J. Shane Culpepper
RMIT University
Melbourne, Australia
shane.culpepper@rmit.edu.au

Falk Scholer
RMIT University
Melbourne, Australia
falk.scholer@rmit.edu.au

ABSTRACT

Term segmentation plays a vital role in building effective information retrieval systems. In particular, languages such as Japanese and Chinese require a morphological analyzer or a word segmenter to identify potential terms. The alternative approach to indexing a segmented collection is n-gram search, where every n-length sequence of symbols is indexed. Both approaches have strengths and weaknesses when applied to non-English collections. In this study, we explore data fusion techniques to answer the following question: if there are multiple ranked lists of documents from both word and n-gram indexes, can we improve overall effectiveness by combining them? We consider three empirical methods for combining search results using eight different search indexes and twenty-one different search models with and without automatic query expansion. Our approach is language independent; however, we focus on Japanese test collections – NTCIR IR4QA – as our testbed for the current experiments. Our experimental results demonstrate that the combination of the two different segmentation approaches has the potential to significantly outperform the best word-segmented search methods.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

Keywords

term segmentation; morphological analysis; n-gram search

1. INTRODUCTION

Identifying words to index is a fundamental problem in information retrieval. In segmented languages such as English and Spanish, white spaces between words provide strong indicators of term boundaries. By virtue of this explicit word segmentation, terms in text are more easily recognized and indexed by a search system. This approach will be referred to as *word search*. Two potential problems arise for the word search approach. First, there are many trivial words and affixes, but stemming and stopping words during the indexing process can help mitigate this problem somewhat. Second, a single word may be ambiguous, and a combination of multiple words may be a more meaningful linguistic unit. Hence, the associativity between terms can be captured using term proximity and phrasal components [14].

Let us consider "W_1 W_2 W_3" as an example sequence of English words, meaning "International Film Festival" as shown in Figure 1. While each word has a meaning, each pair of neighboring words may also have a meaning. Even a knowledgeable human might not recognize the best term-segment for the sequence without additional context. For example, "International Film Festival" can be divided into "International," "Film," and "Festival," in which each of the words is meaningful. On the other hand, compound words such as "Film Festival" and "International Film Festival" can have another meaning derived from the combination of words.

In non-segmented languages, words are written with character sequences without white spaces. In Japanese, the above example would look like "$X_1X_2X_3X_4X_5$" with "X_1X_2" for "International," "X_3X_4" for "Film," and "X_5" for "Festival," respectively, as shown in Figure 1.

To recognize meaningful words or phrases from such character sequences, linguistic tools such as morphological analyzers are generally used. A morphological analyzer uses a word dictionary to insert plausible segments into character sequences. Dictionaries and algorithms can vary widely among different morphological analyzers. Therefore, different morphological analyzers can produce different index terms. To avoid the uncertainty in term segmentation, an overlapping n-length character sequence or n-gram can also be used as a surrogate for terms in the indexing process. For the above example, "(X_1), (X_2), (X_3), $\cdots$" for a 1-gram index, "(X_1X_2), (X_2X_3), $\cdots$" for a 2-gram index, "$(X_1X_2X_3)$, $(X_2X_3X_4)$, $\cdots$" for a 3-gram index, respectively, are generated as terms. We refer to this approach as an *n-gram search*.

In this study, we investigate several approaches to combining both methods using data fusion, empirically evaluating the best performance that would be achievable compared to using the methods individually. This serves as a useful benchmark, and clarifies the feasibility of obtaining substantial benefit from using the proposed combined technique. The following sections present a discussion of data fusion techniques and how to apply it in merging multiple rankings. We also discuss our experimental methodology and improvements in search effectiveness.

2. RELATED WORK

Data fusion is a common technique used in Information Retrieval to perform *meta-search* [23], and can be used to aggregate ranked document lists using multiple search models and linguistic processes. Here, linguistic processes include stemming, stopping, term segmentation, and morphological analysis. Search models include

ADCS, December 08 - 09, 2015, Parramatta, NSW, Australia

ISBN 978-1-4503-4040-3/15/12…$15.00
DOI: http://dx.doi.org/10.1145/2838931.2838939

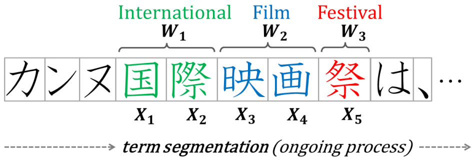

Figure 1: An example sequence of Japanese characters. X_i presents a character and W_j presents a corresponding word. The connected characters, X_1 and X_2 mean "International." $X_3X_4X_5$ and $X_1X_2X_3X_4X_5$ mean "Film Festival," and "International Film Festival," respectively, and are also meaningful.

the standard vector space model, language models, unsupervised divergence from randomness models, and many others.

Basic data fusion approaches originally studied by Fox and Shaw [10] have been used for English search tasks in TREC [11, 13, 22], for European search tasks in CLEF [15, 8], and for Asian search tasks in NTCIR [12, 19, 7].

For data fusion methods in English, various types of searches including word, soundex, n-gram, skip-gram indexes have been examined [13]. However, similarly comprehensive experiments for data fusion in Japanese collections do not exist. McNamee [12] experimented on 6-grams and words in English search, but the experiments for Japanese did not combine word and n-gram search. Instead, McNamee considered only combining 2-grams and 3-grams in Japanese. Savoy [19] used test collections in Chinese, Japanese and Korean, and performed a variety of experiments including data fusion using 2-grams, and two different search models. They did not explore combining word and n-gram searches.

For the IR4QA task in NTCIR7/8 [17, 18], which is a simple ad-hoc search task in Japanese, some participant groups have investigated word search and n-gram search [21, 20]. In their experiments, they used technologies developed more than a decade ago, and significant improvements were not reported for data fusion.

Abdulahhad et al. [8] introduce an effective indexing approach using "concepts." To acquire the concepts, they use a medical thesaurus. In their experiments, a basic data fusion method [10] is applied to obtain better results. For term segmentation in Japanese, Ogawa and Matsuda [16] use 2-grams to obtain statistical word segments in test collections. Their approach does not require any dictionary maintenance for morphological analyzers, but needs a training corpus to calculate the probability of each 2-gram to determine if the two neighboring symbols should be disconnected. Both of the above approaches require specialists tools and training sets in order to achieve good effectiveness. Our interest is to pursue reproducible approaches that do not require training data or dictionary maintenance by humans, and that use off-the-shelf techniques and standard test collections.

3. PROPOSED METHOD

We first employ a basic linear combination [23, pp.73–116] to combine word and n-gram searches. For simplicity, we refer to a ranked list of documents as a *run*. Wu et al. [22] demonstrated that the linear combination method (hereafter, *Linear*) can be used to combine a run from a baseline search and a run from an anchor text search in order to improve overall effectiveness. Their approach using Linear to infer a relevance score (hereafter, *RS*) is applicable to our problem of combining a run from word and n-gram searches.

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

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

where α is a weighting parameter.

Different runs may contain different ranges of relevance scores. Therefore, score normalization [23, pp.19–42] is effective when combining runs. One straightforward normalization method is to normalize the scores of the documents in each run into the range of $[0, 1]$. Wu et al. [22] present the effectiveness of this linear normalization method (hereafter, *LNorm*). We apply LNorm in our data fusion techniques.

LNorm normalization: The normalized relevance score is calculated using the maximum and minimum document relevance score of a run. Specifically, the normalized relevance score, NRS for a retrieved document d is calculated as

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

where MAX and MIN represent the maximum and minimum RS of a run, respectively. Because the methods without LNorm are ineffective, we discuss only methods with LNorm for our study. When identifying the parameter α, we explore the following three empirical weighting methods.

Constant weighting: The weight parameter α is chosen from 0.0 to 1.0 in increments of 0.1. The same parameter is applied to all queries, and the value is determined to obtain the maximum MAP value.

Binary weighting: The weight parameter α is 1 or 0, depending on whether a word search or n-gram search is used. Either search is chosen to obtain the maximum AP value for each individual query.

Variable weighting: The weight parameter α is varied from 0.0 to 1.0, and the value is determined to obtain the maximum AP value for each query.

This short paper sets out to explore the maximum benefit that can be expected from such an approach, exploring an empirical upper bound on the effectiveness from combining both word and n-gram ranking.

4. EXPERIMENTS

We use the NTCIR7/8 IR4QA test collections in Japanese [17, 18] consisting of 797,700 documents from the Mainichi News Paper 1998–2005, the search topics, and the judgment files. To obtain a data fusion run, we created multiple word and n-gram search runs using a variety of ranking algorithms. 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 the Terrier IR Platform [6] version 3.5 and different search models[1]. To increase the variety of runs, we also apply automatic query expansion to each of the search models. For the term segmentation process in word search, we use five Japanese linguistic tools. The five word searches are referred to as ChaSen [1], Juman [2], KaKaSi [3], KyTea [4], MeCab [5] word searches, respectively, based on the names of the tools. For different n-gram searches, we considered the feasibility of the experiment and determined to use 1-gram, 2-gram, and 3-gram searches.

The best initial runs for both test collections were produced using the search model XSqrA_M [9] with automatic query expansion.

[1] Specifically, we use the 21 search models including 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.

Data	**NTCIR7**					**NTCIR8**				
Word index	**ChaSen**	**Juman**	**KaKaSi**	**KyTea**	**MeCab**	**ChaSen**	**Juman**	**KaKaSi**	**KyTea**	**MeCab**
Initial single run (baseline)	0.6615	0.6892	0.6808	**0.7072**	0.6936	0.5158	**0.5411**	0.5362	0.5341	0.5271
n**-gram index**	**1-gram**	**1-gram**	**1-gram**	**1-gram**	**1-gram**	**1-gram**	**1-gram**	**1-gram**	**1-gram**	**1-gram**
Initial single run (to be combined)	0.3228‡	0.3228‡	0.3228‡	0.3228‡	0.3228‡	0.3092‡	0.3092‡	0.3092‡	0.3092‡	0.3092‡
Combine Cons (Word and 1-gram)	0.6647	0.6911	0.6884	0.7090	0.6969	0.5158	0.5411	0.5362	0.5341	0.5271
Combine Bin (Word and 1-gram)	0.6708	0.6989	0.6950	0.7171	0.7022	0.5219	0.5416	0.5386	0.5428	0.5307
Combine Var (Word and 1-gram)	0.6762†	0.7038†	0.7037†	0.7222†	0.7093†	0.5267†	0.5458‡	0.5446‡	0.5460	0.5351†
n**-gram index**	**2-gram**	**2-gram**	**2-gram**	**2-gram**	**2-gram**	**2-gram**	**2-gram**	**2-gram**	**2-gram**	**2-gram**
Initial single run (to be combined)	0.6356	0.6356†	0.6356†	0.6356‡	0.6356†	0.4793†	0.4793‡	0.4793‡	0.4793‡	0.4793†
Combine Cons (Word and 2-gram)	0.6738	0.7030	0.6945	0.7145	0.7037	0.5297†	0.5437	0.5427	0.5356	0.5367
Combine Bin (Word and 2-gram)	0.7160†	0.7191†	0.7125	0.7274†	0.7173†	0.5458‡	0.5557†	0.5561†	0.5539†	0.5486†
Combine Var (Word and 2-gram)	0.7208†	0.7254†	0.7183†	**0.7317**†	0.7223†	0.5535‡	0.5620‡	0.5642‡	0.5594†	0.5558‡
n**-gram index**	**3-gram**	**3-gram**	**3-gram**	**3-gram**	**3-gram**	**3-gram**	**3-gram**	**3-gram**	**3-gram**	**3-gram**
Initial single run (to be combined)	0.5995†	0.5995‡	0.5995‡	0.5995‡	0.5995‡	0.4405‡	0.4405‡	0.4405‡	0.4405‡	0.4405‡
Combine Cons (Word and 3-gram)	0.6737	0.7023	0.6934	0.7150	0.7050	0.5269†	0.5452	0.5398†	0.5366	0.5349
Combine Bin (Word and 3-gram)	0.7125†	0.7158†	0.7060†	0.7268	0.7151†	0.5451†	0.5573†	0.5524†	0.5541†	0.5518†
Combine Var (Word and 3-gram)	0.7195†	0.7235†	0.7132†	0.7314†	0.7219†	0.5566‡	**0.5660**‡	0.5617†	0.5635†	0.5621‡

Table 1: MAP values for initial single runs and the data fusion runs of word and n-gram searches. The MAP values in bold represent the best **word** and **combined** (word and n-gram) runs for NTCIR7 and NTCIR8. The best search model, XSqrA_M [9] with automatic query expansion is used for all of the single and combined runs. Combine Cons, Bin, and Var denote, respectively, the data fusion runs with Constant, Binary, and Variable weighting. † and ‡ respectively denote statistical significance at 0.05 and 0.001 levels based on a 2-tailed paired t-test against each of the MAP values for baseline word searches. Combine Var is the best for combining two runs, but not feasible for three or more runs. By applying Combine Bin to all of the eight different initial runs (from the five word and three n-gram indexes), more effective runs are obtained, with MAP values of **0.7439**‡ and **0.5810**‡ for NTCIR7 and NTCIR8, respectively.

Effectiveness results for the initial single runs and the combined runs are shown in Table 1. The Juman parsing scheme is the best for NTCIR7 and the KyTea parsing is the best for NTCIR8. These two cutting-edge parsers have not been examined thoroughly, while the ChaSen, KaKaSi and MeCab parsers were used extensively in previous Japanese Information Retrieval experiments. In this work we compare all of the currently available methods. The KyTea parsing for NTCIR7 outperforms the other word-based methods ($p < 0.05$) and all of the n-gram-based methods ($p < 0.001$). For NTCIR8, the Juman parsing outperforms ChaSen ($p < 0.05$), MeCab ($p < 0.05$), 1-gram ($p < 0.001$), 2-gram ($p < 0.05$), and 3-gram ($p < 0.001$). However, it does not show statistical significance against the KaKaSi and KyTea parsings.

As shown in the table, n-gram searches are significantly worse than the baseline word searches in nearly every case (indicated with † or ‡). The ChaSen parsing is the worst among the word searches, and the 2-gram search is the best among the n-gram searches. As a result, the difference between the ChaSen word search and the 2-gram search was not significant for NTCIR7 ($p = 0.285$). To improve the search effectiveness for the combined runs, the contribution of 1-gram is minimal. While none of the 2-gram and 3-gram runs are effective alone, all of the combined runs using these n-gram runs outperform the baseline runs.

As shown in the Table, the Variable method produces better MAP values than the Constant and Binary methods because it adaptively tunes the parameter α when combining a word search and an n-gram search. The best MAP values for the combined runs are obtained by using the Variable method, with MAP values of 0.7317 ($p < 0.05$) and 0.5660 ($p < 0.001$) for NTCIR7 and NTCIR8 respectively.

The Binary method is more efficient than the Constant and Variable method. Applying the Constant or Variable method to an exhaustive data fusion of many runs is not feasible as they require more computationally expensive parameter tuning. On the other hand, the Binary method can be easily applied to combine three or more runs as it simply chooses the maximum AP value for each topic. By applying the Binary method to the data fusion of all of the word and n-gram search runs, the best oracle runs are obtained, with MAP values of 0.7439 ($p < 0.001$) and 0.5810 ($p < 0.001$) for NTCIR7 and NTCIR8 respectively.

To compare the properties of different search indexes, the number of unique terms in the KyTea word index for NTCIR7 and the Juman word index for NTCIR8 are, respectively, 440,105 and 231,209. The minimum and maximum length of index terms is 1 and 20 for both cases. The number of unique terms in the 1-gram, 2-gram and 3-gram indexes for NTCIR7 are 5,431, 860,181 and 8,225,898. The number of unique terms in the 1-gram, 2-gram and 3-gram indexes for NTCIR8 are 5,166, 780,604 and 7,280,608. Morphological analyzers associate known words with subsequences in unknown words, and sometimes meaningful words are broken. These broken words can be harmful or harmless, depending on how the search model performs. Specifically, the same sequence of characters $X_1X_2X_3$ is segmented differently by different morphological analysis — $((X_1X_2)\&(X_3))$, $((X_1)\&(X_2X_3))$, $(X_1X_2X_3)$ — and as such causes inconsistencies in the search query and the documents retrieved. While the n-gram indexes avoid such uncertainty in term segmentation, the word indexes are capable of retaining meaningful compound words, such as "Film Festival" and "International Film Festival" as shown in Figure 1. When term segments by morphological analysis are not useful, n-gram searches may make up for the decrease in search effectiveness of the data fusion runs.

The effectiveness results for topic-by-topic comparison are shown in Figure 2. To facilitate visualization, the two best word searches (Juman for NTCIR7 and KyTea for NTCIR8) and the best n-gram search (2-gram) are shown, accompanied by the best oracle run.

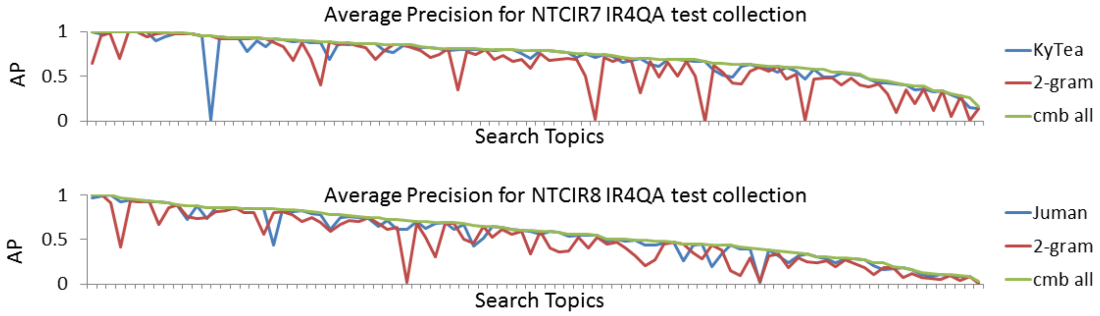

Figure 2: AP values for topic-by-topic comparison. The best search model, XSqrA_M [9] with automatic query expansion is used. The data labeled as "cmb all" is the data fusion run over all word and n-gram searches. Search topics are in descending order of the AP values for the data fusion run.

For each case, the best search model, XSqrA_M [9] is used. Downward spikes in the figure represent lower AP values, and are topic specific. As shown in the figure, word searches are effective for most topics, but occasionally encounter fatal failures. These failures demonstrate the reason for the better performance of the data fusion of word and n-gram searches. Although the n-gram searches are significantly worse than the best word search in overall effectiveness (MAP values shown in Table 1), the n-gram searches win against the word searches in some cases. Consequently, the combined run is boosted up by having the occasional assistance of the n-gram search (AP values shown in Figure 2). To investigate search effectiveness for such difficult search topics, new test collections that are sensitive to ambiguous term segments may be required.

5. CONCLUSION

We have explored empirical approaches to merging ranked retrieval results from multiple representations of a single collection. Our experiments have verified that word search is generally more effective than n-gram search but not always. It should therefore be possible to exploit the techniques in combination to achieve higher overall effectiveness, regardless of search model.

The results for data fusion, using the best oracle method, show that combining information from word and n-gram indexing approaches can be significantly better than either index in isolation. These initial results are promising and demonstrate the potential of related methods. Here we focused on word and n-gram searches in the Japanese language. Our methods are easily extended to other languages and may also be used to combine search results from multiple parsings of a single collection, including various stopping and stemming combinations.

Future work will investigate techniques such as query difficulty prediction for setting the combination parameter automatically. We will also explore the association between query difficulty and various indexing methods in unsegmented languages, in order to find new ways to improve system effectiveness in a language independent way.

Acknowledgment This work was supported partially by JSPS KAKENHI Grant Number 26330363 and the Australian Research Council's *Discovery Projects* Scheme (Project DP140102655). Shane Culpepper is the recipient of an Australian Research Council DECRA Research Fellowship (DE140100275).

References

[1] ChaSen. https://osdn.jp/projects/chasen-legacy/, 2012.

[2] JUMAN. http://nlp.ist.i.kyoto-u.ac.jp/index.php?JUMAN, 2012.

[3] KAKASI. http://kakasi.namazu.org/index.html.en, 2013.

[4] KyTea. http://www.phontron.com/kytea/, 2013.

[5] MeCab. http://taku910.github.io/mecab/, 2013.

[6] Terrier IR Platform. http://terrier.org/, 2014.

[7] 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.

[8] K. Abdulahhad, J. Chevallet, and C. Berrut. Matching fusion with conceptual indexing. In *Proc. of RISE2012*, 2012.

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

[10] E. Fox and J. Shaw. Combination of multiple searches. In *Proc. of TREC-2*, pages 243–252, 1994.

[11] J. Mayfield and P. McNamee. Combining methods for the TREC 2003 robust track. In *Working Notes of TREC 2003*, 2003.

[12] P. McNamee. Experiments in the retrieval of unsegmented Japanese text at the NTCIR-2 workshop. In *Proc. of the 2nd NTCIR Workshop Meeting*, 2001.

[13] P. McNamee, C. K. Nicholas, and J. Mayfield. Addressing morphological variation in alphabetic languages. In *Proc. of SIGIR '09*, pages 75–82, 2009.

[14] D. Metzler and W. B. Croft. A markov random field model for term dependencies. In *Proc. of SIGIR '05*, pages 472–479, 2005.

[15] C. Monz, J. Kamps, and M. de Rijke. The university of Amsterdam at CLEF 2002. In *CLEF Working Notes*, 2002.

[16] Y. Ogawa and T. Matsuda. Overlapping statistical word indexing: A new indexing method for Japanese text. In *Proc. of SIGIR '97*, pages 226–234, 1997.

[17] 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.

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

[19] J. Savoy. Comparative study of monolingual and multilingual search models for use with asian languages. *TALIP*, 4(2):163–189, 2005.

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

[21] 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.

[22] 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.

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