

RMIT and Gunma University at NTCIR-9 Intent Task

Michiko Yasukawa* J. Shane Culpepper† Falk Scholer† Matthias Petri†

*Gunma University, Kiryu, Japan
michi@cs.gunma-u.ac.jp

†RMIT University, Melbourne, Australia
{shane.culpepper, falk.scholer, matthias.petri}@rmit.edu.au

ABSTRACT

In this report, we describe our experimental results for the NTCIR-9 INTENT task. For our experiments, we use our experimental search engine, NewT. NewT is a ranked self-index capable of supporting multiple languages by deferring linguistic decisions until query time. To our knowledge, this is the first Information Retrieval task on the ClueWeb09-JA collection performed entirely with ranked self-indexes.

Categories and Subject Descriptors

H.3.1 [Information Storage and Retrieval]: Content Analysis and Indexing—*indexing methods*; H.3.2 [Information Storage and Retrieval]: Information Storage—*file organization*; H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—*query formulation, retrieval models, search process*; I.7.3 [Document and Text Processing]: Text Processing—*index generation*

General Terms

Text Indexing, Text Compression, Language Independent Text Indexing, Data Storage Representations, Experimentation, Measurement, Performance

1. INTRODUCTION

In this paper, we describe our experimental approach for the NTCIR-9 INTENT task. We present a novel approach to language independent, ranked document retrieval using our new self-index search engine called NewT. NewT is a ranked self-index capable of supporting multiple languages by deferring linguistic decisions until query time. Our search engine uses backwards search in a Burrows-Wheeler transform similar to the FM-index of Ferragina and Manzini [6] and a *wavelet tree* [7] for occurrence counting in a document array [13]. To our knowledge, this is the first Information Retrieval task on the ClueWeb09-JA collection performed entirely with ranked self-indexing algorithms.

For this year's task, our group only participated in the Japanese subtasks as we did not have a Chinese collaborator to assist with language specific issues. Our primary goal for NTCIR-9 was to implement and test our new class of indexing algorithms for multilingual tasks. Our initial evaluation of Japanese queries led to an unexpected problem with character-level indexing, overlapping substrings.

NTCIR-9 Workshop Meeting, 2011, Tokyo, Japan.
Copyright National Institute of Informatics

2. BACKGROUND

The CJKV language family is traditionally indexed as character n -grams or terms. Since Japanese is an unsegmented language, either of these approaches is possible, but recent approaches have focused primarily on morphological term parsing. Conversely, recent work on Chinese tokenization has used character n -grams [10, 15, 11]. Generally, character or n -gram based methods increase recall, while word-based methods improve precision [11].

The word segmentation process is generally performed on Japanese documents by using morphological analyzers, such as ChaSen¹ and MeCab². However, the output of these morphological analyzers is not always correct. Some morphemes suggested by morphological analyzers are overly segmented, especially when documents contain unknown words [22]. Furthermore, output texts are under-segmented or not segmented at all when web pages omit punctuation marks or white space in the text for the sake of simplicity or layout. When wrongly segmented morphemes are stored as index terms, the search results can be poor.

Either of the CJKV methods depend on an inverted index for term or n -gram indexing. Inverted indexes are the dominant solution for IR search [24]. However, forcing CJKV languages to rely on morphological analysis is not always the best option. Character-based methods have certain advantages over term-based methods, but the additional space overhead makes these methods infeasible for large collection. Recently, *self-indexing* algorithms have received a great deal of attention because of their efficient search capabilities and reduced space overhead [14]. These indexing algorithms have interesting theoretical and practical performance on basic pattern matching operations, but ranked search capabilities on large datasets is still open [8].

In this report, we investigate the problem of using self-indexing algorithms to solve the **ranked document search** problem. In order to solve the document search problem, basic self-indexing algorithms are not sufficient. However, using an auxiliary data structure to manage a *document array* enables basic document ranking [13, 19, 4]. For the INTENT task, we use an enhanced version of the *greedy top-k* approach described in [4]. A full discussion of prior work on applications of self-indexes to the **ranked document search** problem is beyond the scope of this report. Please refer to [4] or [5] for a more comprehensive discussion of the algorithms used in NewT.

¹<http://chasen-legacy.sourceforge.jp/>

²<http://mecab.sourceforge.net/>

All prior published work on ranked self-indexes use a trivial $TF \times IDF$ ranking metric, and have generally focused on phrase queries instead of bag-of-words queries. For the INTENT task, two bag-of-words ranking functions were implemented. The first metric is referred to as *raw term frequency* ranking. For this metric, we simply compute the aggregate of raw frequency counts per document, $f_{t,d}$, for each term or substring, t .

$$RAW = \sum_{t \in q} f_{t,d}$$

However, the RAW ranking metric does not normalize for document length or IDF. Therefore, we also implemented a simple BM25 variant as follows:

$$BM25 = \sum_{t \in q} \log \left(\frac{N - f_t + 0.5}{f_t + 0.5} \right) \cdot TF_{BM25}$$

$$TF_{BM25} = \frac{f_{t,d} \cdot (k_1 + 1)}{f_{t,d} + k_1 \cdot ((1 - b) + (b \cdot \ell_d / \ell_{avg}))}$$

Here, N is the number of documents in the collection, f_t is the number distinct documents appearances of t , $k_1 = 1.2$, $b = 0.75$, ℓ_d is the number of UTF8 symbols in the documents, and ℓ_{avg} is the average of ℓ_d over the whole collection. For self-indexes, there is an efficiency trade-off between locating the top- k $f_{t,d}$ values and accurately determining d_t since the index can extract exactly k $f_{t,d}$ values without processing every document. We leave the task of finding the most efficient trade-off between these competing requirements, and devising ranking metrics that do not make term-based *i.i.d.* assumptions as a topic of future work.

2.1 Diversity Ranking

User satisfaction in web search has received significant attention in recent years. While there may be several possible “senses” for a search topic, a user typically wishes to find only one meaning for the topic. In such a situation, browsing many unrelated documents in the ranked list quickly leads to user dissatisfaction. To solve this problem, the Maximal Marginal Relevance (MMR) ranking method provides information with a diversity ranking [1], which minimizes redundancy in the search results. Since diversification was recognized as a challenging problem, experiments on diversity ranking have been conducted in the TREC diversity track [2, 3].

The NTCIR-9 INTENT task [18] focuses on the diversity ranking problem in a manner similar to the TREC Web track, but with a focus on Japanese and Chinese linguistic disambiguation. In the NTCIR-9 INTENT task, subtopic strings were not provided by the organizer, but participants were asked to prepare subtopic strings and submit them in the Subtopic Mining subtask. In the TREC diversity task, subtopics were extracted from the logs of a commercial search engine [2, 3]. Prior to the TREC task, clustering of similar queries [23, 21] and understanding query-query reformulations by users [16] were explored to obtain subtopics from a query log. Unfortunately, a query log or other additional resources were not provided in the Japanese Subtopic Mining subtask of the NTCIR-9 INTENT task.

The diversity task of the TREC Web track [2, 3] introduced two topic types: faceted and ambiguous. Faceted topics

Query Set	Description
Vanilla 1	Word segmentation of original search topics using MeCab, retaining all morphemes. (RMIT-D-J-4)
Vanilla 2	Word segmentation of original search topics using MeCab, retaining only nouns. (RMIT-D-J-2, RMIT-D-J-5)
Diversity 1	Original search topics without word segmentation, with addition of subtopic strings. (RMIT-D-J-1, RMIT-D-J-3)

Table 1: Summary of query sets.

are topics that are not ambiguous by themselves, but have different related subtopic strings. Ambiguous topics are topics that are polysemous, and need to be clarified with subtopic strings. To deal with ambiguous queries, query expansion using lexical-semantic relations was studied [20]. Lexical resource such as WordNet³ and Wikipedia⁴ have been used for resolving word sense disambiguity [12] and for named entity recognition [9]. For the INTENT task, we used the Japanese edition of Wikipedia to obtain subtopic strings for both faceted and ambiguous topics.

3. EXPERIMENTAL FRAMEWORK

In this section, we describe the experimental setup used for the NTCIR-9 INTENT task.

3.1 Collection Processing and Indexing

For the INTENT task, we indexed the ClueWeb09-JA collection in two steps. First, each document from the collection was extracted and normalized using Lynx,⁵ a text-based browser with an option to output processed and formatted texts. The extracted documents are converted to UTF8 character code. Next, all whitespace was removed from each document to create a contiguous UTF8 string, followed by a distinct `end of document` identifier. The fully processed ClueWeb09-JA collection was partitioned into blocks of 500,000 documents and indexed with NewT. Since NewT is an in-memory index, each 500k document block is serialized to disk and processed separately at query time.

3.2 Topic Processing

In order to evaluate NewT, we processed the search topics and generated two vanilla query sets (Vanilla 1, Vanilla 2) and a diversity query set (Diversity 1). Table 1 has a brief descriptions for each of our query sets. In the table, attributes in parenthesis represent the corresponding runs in Table 3.

3.2.1 Vanilla Queries

For the two vanilla query sets, word segmentation of search topics was performed using MeCab. Hence, search topics were decomposed into morphemes, or shorter substrings of the search topics. This is a standard approach for query processing in term-based document retrieval in Japanese text. Hence, we refer to these query sets as

³<http://wordnet.princeton.edu/>

⁴<http://www.wikipedia.org/>

⁵<http://lynx.isc.org/>

“vanilla”. Specific character strings in search topics are resolved into common words in relevant documents using morphological analysis. However, the morphological analysis may also produce ambiguous or irrelevant character substrings from search topics, resulting in performance degradation.

Vanilla 1 and **Vanilla 2** use a slight variation in query expansion: all morphemes were used in **Vanilla 1**, but only nouns were used in **Vanilla 2**, with prefixes, suffixes, and particles being dropped. **Vanilla 1** is therefore essentially a simple parse of the original topic, with little or no modification, and we treat **Vanilla 1** as a baseline in our analysis.

3.2.2 Query Diversification

For the diversity query set, search topics were retained without word segmentation. Hence, queries in this set gather target documents for search topics. Although the queries in this set faithfully represent the original character strings of the search topics, some of them are overly specific, resulting in low recall. At the other extreme, some queries are short and ambiguous, resulting in lower precision. In both cases, query expansion is an effective approach. Subtopic strings that were obtained from the Japanese Wikipedia⁶ for each distinct topic sense were used for expansion.

Topic Disambiguation: In order to gather distinct subtopic senses, we used “Disambiguation in Wikipedia”, which resolves the conflicts that arise when a single term is ambiguous. Specifically, we used three types of disambiguation definitions from Wikipedia: (1) pages, (2) titles, and (3) “See” in articles. For disambiguation pages, we used the Japanese character string, “曖昧さ回避” (meaning “disambiguation page” in English) as a pattern to identify disambiguation pages for search topics. To identify “Disambiguation in Wikipedia” in titles, we also used a set of round parentheses (“(” and “)”) as tokens for pattern matching. For example, a reader of Wikipedia may find the page “Orange”, which is a disambiguation page because it shows “disambiguation page” in the page top, and lists various meanings of “Orange” and links. In the page, major meanings of “Orange” are titled with round parentheses, “Orange (fruit),” “Orange (colour),” etc. Some ambiguous topics are not consolidated in disambiguation pages. Hence, we also used Perl Compatible Regular Expressions(PCRE),⁷ incorporated into the scripting language PHP,⁸ to gather disambiguation definitions from the headings of articles. Specifically, we used the Japanese character strings and regular expressions “.*?については「.*?」をご覧ください。” meaning “For.*?, see.*?”. We also used variant expressions, such as “「.*?」を参照” meaning “See.*?” and “「.*?」も参照” meaning “See also.*?” to obtain disambiguation definitions from articles.

Wikipedia summaries in topics: If a search topic had no disambiguation definitions, but had an article, we obtained the last noun from the first sentence in the article. We assume that such topics are fairly specific because no disambiguation exists on Wikipedia, but the descriptions can

⁶<http://ja.wikipedia.org/>

⁷<http://www.pcre.org/>

⁸<http://www.php.net/>

Run	Description
RMIT-S-J-1	Obtain subtopic strings using pattern matching from Japanese Wikipedia.

Table 2: Subtopic mining run.

help to make ambiguous queries more specific. According to Wikipedia’s Style Manual,⁹ the introduction of a Wikipedia article is a summary of the most important aspects. Extracting all aspects from the introduction is preferred, but difficult to automate. The automation would require a precise analysis of the dependency structure in Japanese. In this task, we performed word segmentation on the first sentence by using MeCab, and simply chose the last noun from the result of morphological analysis. For example, a reader of Wikipedia may find the sentence “マカロンは、アーモンドを使ったフランスを代表する洋菓子である。” meaning “A macaroon is a western confectionery, which is made with almond and a representative of France.” as the first sentence in the page “マカロン” meaning “Macaroon.” This sentence contains the five nouns, “マカロン,” “アーモンド,” “代表,” “フランス,” “洋菓子” meaning “macaroon,” “almond,” “representative,” “France,” “western confectionery,” respectively. We obtained the last noun in the sentence, which is “洋菓子” meaning “western confectionery” as an important aspect for the topic. The expanded query for the example is “マカロン, 洋菓子” meaning, “macaroon, western confectionery” in English.

Topics as queries: Some search topics were not listed in the Japanese Wikipedia. For those topics, no diversification was applied to the queries. However, some topics were excessively specific. Consequently, the queries could not be applied directly to the collection. In these instances, the query was represented as both the search topic and all of the nouns in the search topic. For example, with the excessively specific search topic: “スターバックスのシナモンロールのレシピ”, meaning “cooking recipes for cinnamon rolls in Starbucks stores” in English is used as a query, too few results are returned. So, both the topic and all the nouns are used in the query, “スターバックスのシナモンロール のレシピ, スターバックス, シナモンロール, レシピ” meaning “cooking recipes for cinnamon rolls in Starbucks stores, Starbucks stores, cinnamon rolls, cooking recipes” in English.

3.3 Description of Runs

We submitted a single run for subtopic mining and 5 runs for document ranking. Tables 2 and 3 give brief descriptions of our submitted runs. In Table 3, attributes in parenthesis represent the corresponding query sets in Table 1.

MeCab, a dictionary based morphological analyzer, was used for topic processing. By default, MeCab uses a recommended standard dictionary, the IPA dictionary.¹⁰ Although the IPA dictionary contains a large number of entries for Japanese morphemes, some search topics were not in the vocabulary. In our preliminary experiment, some search topics were tagged as unknown words by the morphological analysis, and meaningful morphemes were segmented into shorter substrings. If morphemes in queries are overly segmented into short strings, search results

⁹http://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style

¹⁰<http://sourceforge.jp/projects/ipadic/>

Run	Description
RMIT-D-J-1	Simple BM25 ranking with NewT using round robin query diversification. (Diversity 1)
RMIT-D-J-2	Simple BM25 ranking with NewT using naive query expansion. (Vanilla 2)
RMIT-D-J-3	Simple TF ranking with NewT using round robin query diversification. (Diversity 1)
RMIT-D-J-4	Simple BM25 ranking with NewT using unmodified topic queries. (Vanilla 1)
RMIT-D-J-5	Simple TF ranking with NewT using naive query expansion. (Vanilla 2)

Table 3: Document ranking runs.

Run	Vanilla 1	Vanilla 2	Diversity 1
RMIT-S-J-1			★ Pattern Match
RMIT-D-J-1			★BM25
RMIT-D-J-2		★BM25	
RMIT-D-J-3			★TF
RMIT-D-J-4	★BM25		
RMIT-D-J-5		★TF	

Table 4: Run and query set correspondence.

tend to contain more irrelevant documents, and the search performance is notably degraded. For instance, if the word “macaroon” is segmented into three short strings, “mac,” “aro,” “on” , search results are intermingled with documents for “macintosh,” “aromatherapy,” “onomatopoeia” and so on. To reduce the risk of this problem, we leveraged the Japanese Wikipedia title list to expand the dictionary for MeCab. We defined dictionary entries for search topics that are not included in the IPA dictionary, but are present as Wikipedia titles. All newly added words were treated as nouns in our experimental dictionary. We used both dictionaries in the topic processing of the **Vanilla 1**, **Vanilla 2** and **Diversity 1** runs. Therefore, search topics that are Wikipedia titles are unmodified in the query sets.

Note that our vocabulary building for topic processing was performed independently from collection processing. The number of **ClueWeb09-JA** documents is 67,337,717 and the morphological analysis for the entire document corpus is burdensome. If the search engine utilizes inverted indexes, the morphological analysis must be done on the entire collection each time the vocabulary for topic processing is changed. Such recurrent morphological analysis is impractical, especially for massive collections such as **ClueWeb09-JA**. Since our approach utilizes self-indexes, we do not need a predefined vocabulary for document processing. In our approach, documents are instead indexed without morphological analysis, and the vocabulary for topic processing is applied at query time, independent of the original indexing process.

Search results from **Diversity 1** were merged into a single run per topic in a round robin fashion. Algorithm 1 shows the round robin diversity merge algorithm used in **RMIT-D-J-1** and **RMIT-D-J-3**. First, a total of s senses of each topic are derived from Wikipedia, and ranked

Algorithm 1 Wikipedia Round Robin Merge Algorithm

INPUT: A collection of s ranked document lists, $W_1 \dots W_s$, of length k_{max} .
 OUTPUT: A ranked list of k_{max} documents, A .

```

1: set  $i \leftarrow 1$ 
2: set  $A \leftarrow \{ \}$ 
3: set  $doc \leftarrow \mathbf{First}(W_i)$ 
4: for  $k = 1$  to  $k_{max}$  do
5:   while  $doc \in A$  do
6:     set  $doc \leftarrow \mathbf{Successor}(W_i)$ 
7:   end while
8:   Append( $A, doc$ )
9:   set  $i \leftarrow (i \bmod s) + 1$ 
10: end for
11: return  $A$ 

```

separately to a depth of k_{max} . Next, a single answer set A is generated by a round robin merge of all of the Wikipedia Sense lists $W_1 \dots W_s$. For each list i , the next highest ranked document in list W_i is found and appended to A to a total length k_{max} . This ensures that each sense contributes the same percentage of documents to the final ranked list.

Runs that were submitted to the document ranking task had to be ordered according the priority in which they should be evaluated by the task organisers. Run names with smaller numbers therefore correspond to a higher priority. Based on a preliminary experiment, we defined the following conditions:

1. Put a higher priority on runs with **Diversity 1** (query set) because these were expanded with subtopic senses from Japanese Wikipedia and merged with round robin query diversification. We expected this approach to achieve the best performance across all of our runs.
2. In terms of the two vanilla query sets, put a higher priority on runs with **Vanilla 2** (query set) because particles and affixes in **Vanilla 1** (query set) were likely to be unnecessary in the search process, with non-nouns being expected to reduce ranking effectiveness.
3. In terms of the ranking algorithm, we put a higher priority on runs with BM25 (ranking algorithm) since the TF (ranking algorithm) does not include any IDF renormalisation.

The corresponding relationships between the submitted runs and the query sets are shown in Table 4. Figure 1 gives a diagrammatic summary of how the runs were prepared.

4. EVALUATION AND RESULTS

In this section we present the results of our experiments for the Intent task.

4.1 Evaluation Metrics

The official effectiveness performance measures for the intent subtopic mining and document ranking tasks are: I-rec, which measures the proportion of intents covered by the documents in the search results list; D-nDCG, which uses a global gain to measure how relevant each document is to an intent, weighted by the importance of each intent;

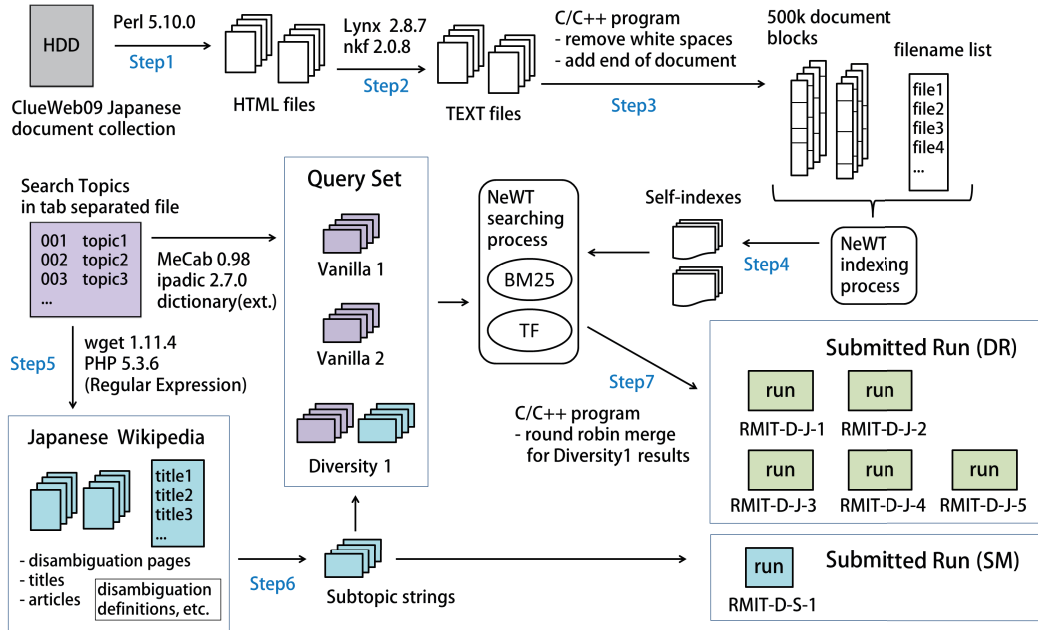


Figure 1: RMIT runs at a glance.

Cutoff	I-rec@10	D-nDCG@10	D#-nDCG@10
10	0.0876	0.0973	0.0925
30	0.0876	0.0624	0.0750

Table 5: Effectiveness results based on the I-rec, D-nDCG and D#nDCG measures for the subtopic mining run RMIT-S-J-1.

and, D#-nDCG, which is a linear combination of I-rec and D-nDCG [17]. D#-nDCG was chosen as the primary evaluation measure by the task organisers.

4.2 Subtopic Mining

For the subtopic mining task, we submitted a single run, RMIT-S-J-1. The effectiveness results are shown in Table 5. Our run for the subtopic mining task did not identify many subtopic strings. Consequently, our intent recall was low. Figure 2 shows the number of identified subtopics strings and intents in our runs, together with the number of officially judged intents in the task.

4.3 Document Ranking

Five runs were submitted by our team for the intent document ranking task, as detailed in Table 3. The run RMIT-D-J-4 used the Okapi BM25 similarity function and the original topic queries, segmented using MeCab (Vanilla 1). We therefore consider this run as a *baseline*, against which the effectiveness of our other runs is compared.

Tables 6 and 7 show the results of our runs for the three evaluation metrics with cutoffs at rank positions 10 and 30, respectively. For both cutoff levels, the best performance was achieved by run RMIT-D-J-3 which used the Diversity 1 queries with TF weighting, and a round-

Run	I-rec@10	D-nDCG@10	D#-nDCG@10
RMIT-D-J-4	0.6489	0.3301	0.4895
RMIT-D-J-3	0.6723	0.3664	0.5193
RMIT-D-J-1	0.6356	0.3540	0.4948
RMIT-D-J-2	0.6306†	0.3283	0.4795
RMIT-D-J-5	0.5639†	0.2989	0.4314†

Table 6: Effectiveness results based on the I-rec, D-nDCG and D#nDCG measures at cutoff 10. Run RMIT-D-J-4, which uses the Okapi BM25 similarity measure with the original queries and no diversification of the ranked results list is treated as a baseline; † and ‡ indicate statistical significance relative to the baseline at the 0.05 and 0.001 levels, respectively, based on a paired *t*-test.

robin re-ranking approach for diversification of the results list, leading to an absolute improvement in D#-nDCG of 0.0298 and 0.0004 for cutoff levels 10 and 30, respectively. However, these improvements are not statistically significant relative to the RMIT-D-J-4 baseline using Okapi BM25 and no diversification.

Adding round-robin diversification to the BM25 similarity measure (run RMIT-D-J-1) led to a fractional improvement over the non-diversified baseline at cutoff level 10 (an absolute increase in D#-nDCG of 0.0053), and a fractional decrease in performance relative to the baseline at cutoff level 30 (an absolute decrease in D#-nDCG of 0.013); the impact of this diversification approach on the Okapi BM25 measure was not statistically significant in either case.

The additional processing of query terms by dropping non-nouns (Vanilla 2 queries in run RMIT-D-J-2) led to a statistically non-significant decrease in Okapi BM25

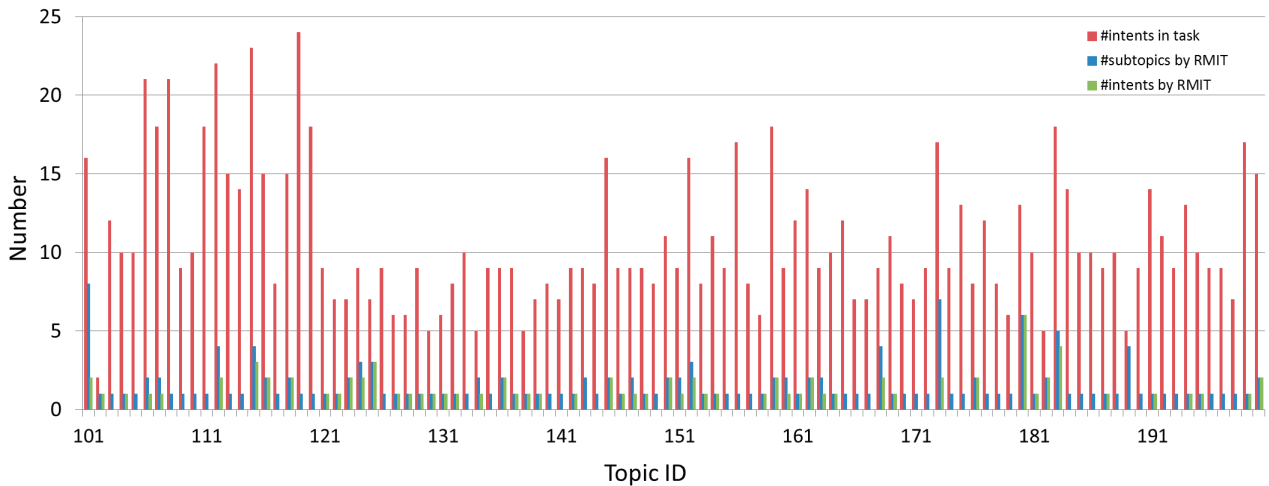


Figure 2: Number of subtopic strings and intents.

Run	I-rec@30	D-nDCG@30	D#-nDCG@30
RMIT-D-J-4	0.8012	0.3617	0.5814
RMIT-D-J-3	0.7836	0.3800	0.5818
RMIT-D-J-2	0.7977	0.3575	0.5776
RMIT-D-J-1	0.7752	0.3617	0.5684
RMIT-D-J-5	0.6759 [†]	0.3118 [†]	0.4938 [‡]

Table 7: Effectiveness results based on the I-rec, D-nDCG and D#nDCG measures at cutoff 30. Run RMIT-D-J-4, which uses the Okapi BM25 similarity measure with the original queries and no diversification of the ranked results list, is treated as a baseline; [†] and [‡] indicate statistical significance relative to the baseline at the 0.05 and 0.001 levels, respectively, based on a paired *t*-test.

performance at both cutoff levels.

Finally, combining the *Vanilla 2* queries with a TF weighting scheme (run RMIT-D-J-5) led to the lowest performance for both cutoff levels, with a statistically significant decrease relative to the baseline run.

5. CONCLUSIONS

In this report, we have presented results for our new experimental ranked self-index *NeWT*, using the *ClueWeb09-JA* collection. Our experiments investigated the impact of diversifying topics using the Japanese Wikipedia, and using a round-robin approach for re-ranking search result lists for individual topic aspects. These approaches did not have a significant impact on search performance as measured by D#-nDCG, relative to an already strong BM25 baseline. We suspect that this is due to the relatively small number of intents that were identified, compared to the total number of officially identified intents (see Figure 2).

While overall effectiveness for document ranking in our first attempt with the *INTENT* topics is suboptimal, we have shown that ranked self-indexes are a viable alternative to classical inverted indexing approaches for Gigabyte scale Japanese document collections. Furthermore, our method

does not require any domain knowledge about the underlying text being indexed, allowing all domain and language decisions to be deferred until query time. In future work, we will investigate new approaches to improving system effectiveness using query expansion and alternative ranking algorithms within our self-indexing framework.

6. ACKNOWLEDGMENTS

The second author was supported by the Australian Research Council.

7. REFERENCES

- [1] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International Conference on Research and Development in Information Retrieval (SIGIR 1998)*, pages 335–336, New York, NY, USA, August 1998. ACM Press.
- [2] C. L. A. Clarke, N. Craswell, and I. Soboroff. Overview of the TREC 2009 web track. In E. M. Voorhees and L. P. Buckland, editors, *Proceedings of the 18th Text REtrieval Conference (TREC 2009)*. NIST Special Publication 500 – 278, November 2009. See <http://trec.nist.gov>.
- [3] C. L. A. Clarke, N. Craswell, I. Soboroff, and G. V. Cormack. Overview of the TREC 2010 web track. In E. M. Voorhees and L. P. Buckland, editors, *Proceedings of the 19th Text REtrieval Conference (TREC 2010)*. NIST Special Publication 500 – 279, November 2010. See <http://trec.nist.gov>.
- [4] J. S. Culpepper, G. Navarro, S. J. Puglisi, and A. Turpin. Top-*k* ranked document search in general text databases. In M. de Berg and U. Meyer, editors, *Proceedings of the 18th Annual European Symposium on Algorithms (ESA 2010), Part II*, volume 6347 of *LNCS*, pages 194–205. Springer, 2010.
- [5] J. S. Culpepper, M. Yasukawa, and F. Scholer. Language independent ranked retrieval with *NeWT*.

- Technical Report RMIT-TR-11-03, RMIT University, October 2011.
- [6] P. Ferragina and G. Manzini. Opportunistic data structures with applications. In *Proceedings of the 41st IEEE Annual Symposium on Foundations of Computer Science (FOCS 2000)*, pages 390–398. IEEE Computer Society Press, November 2000.
- [7] R. Grossi, A. Gupta, and J. S. Vitter. Higher-order entropy-compressed text indexes. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2003)*, pages 841–850, January 2003.
- [8] W. K. Hon, R. Shah, and J. S. Vitter. Compression, indexing, and retrieval for massive string data. In A. Amir and L. Parida, editors, *Proceedings of the 21st Annual Symposium on Combinatorial Pattern Matching (CPM 2010)*, volume 6129 of *LNCS*, pages 260–274. Springer, June 2010.
- [9] J. Kazama and K. Torisawa. Exploiting Wikipedia as external knowledge for named entity recognition. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)*, pages 698–707. ACL, June 2007.
- [10] K. L. Kwok. Comparing representations in Chinese information retrieval. In *Proceedings of the 20th Annual International Conference on Research and Development in Information Retrieval (SIGIR 1997)*, pages 34–41. ACM Press, August 1997.
- [11] R. W. P. Luk and K. L. Kwok. A comparison of Chinese document indexing strategies and retrieval models. *ACM Transactions on Asian Language Information Processing*, 1(3):225–268, 2002.
- [12] R. Mihalcea. Using Wikipedia for automatic word sense disambiguation. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2007)*, pages 196–203. The Association for Computational Linguistics, April 2007.
- [13] S. Mithukrishnan. Efficient algorithms for document retrieval problems. In D. Eppstein, editor, *Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2002)*, pages 657–666, January 2002.
- [14] G. Navarro and V. Mäkinen. Compressed full-text indexes. *ACM Computing Surveys*, 39(1):2–1 – 2–61, 2007.
- [15] J.-Y. Nie, K. Gao, J. Zhang, and M. Zhou. On the use of words and n -grams for Chinese information retrieval. In *Proceedings of the 5th Annual International Workshop on Information Retrieval with Asian Languages (IRAL 2000)*, pages 141–148. ACM Press, 2000.
- [16] F. Radlinski and S. Dumais. Improving personalized web search using result diversification. In *Proceedings of the 29th Annual International Conference on Research and Development in Information Retrieval (SIGIR 2006)*, pages 691–692, New York, NY, USA, 2006. ACM Press.
- [17] T. Sakai, N. Craswell, R. Song, S. Robertson, Z. Dou, and C. Y. Lin. Simple evaluation metrics for diversified search results. In *Proceedings of the 3rd Annual International Workshop on Evaluation of Information Access (EVIA 2010)*, pages 42–50. National Institute of Informatics, 2010.
- [18] R. Song, M. Zhang, T. Sakai, M. Kato, Y. Liu, M. Sugimoto, Q. Wang, and N. Orii. Overview of the NTCIR-9 INTENT task. In *Proceedings of NTCIR-9 Workshop Meeting*, December 2011.
- [19] N. Välimäki and V. Mäkinen. Space-efficient algorithms for document retrieval. In B. Ma and K. Zhang, editors, *Proceedings of the 18th Annual Symposium on Combinatorial Pattern Matching (CPM 2007)*, volume 4580 of *LNCS*, pages 205–215. Springer, July 2007.
- [20] E. M. Voorhees. Query expansion using lexical-semantic relations. In W. B. Croft and C. J. van Rijsbergen, editors, *Proceedings of the 17th Annual International Conference on Research and Development in Information Retrieval (SIGIR 1994)*, pages 61–69. ACM Press, July 1994.
- [21] J. R. Wen, J. Y. Nie, and H. J. Zhang. Query clustering using user logs. *ACM Transactions on Information Systems (TOIS)*, 20:59–81, January 2002.
- [22] M. Yasukawa and H. Yokoo. Composition and decomposition of Japanese katakana and kanji morphemes for decision rule induction from patent documents. In *Proceedings of the 15th Australasian Document Computing Symposium (ADCS 2010)*, pages 28–35, December 2010.
- [23] Z. Zhang and O. Nasraoui. Mining search engine query logs for query recommendation. In *Proceedings of the 15th international conference on World Wide Web (WWW 2006)*, pages 1039–1040, New York, NY, USA, May 2006. ACM Press.
- [24] J. Zobel and A. Moffat. Inverted files for text search engines. *ACM Computing Surveys*, 38(2):6–1 – 6–56, 2006.