RMIT and Gunma University at NTCIR-9 GeoTime Task

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

*michi@cs.gunma-u.ac.jp †{shane.culpepper, falk.scholer, matthias.petri}@rmit.edu.au

RMIT UNIVERSITY

群馬大学 GUNMA UNIVERSITY

Introduction

Self-indexing algorithms have interesting theoretical and practical performance on basic pattern matching operations[1], but ranked search capabilities on large datasets is still open. We investigate the problem of using self-indexing algorithms to solve the **ranked document search** problem.

Collection Processing

We participated in the English - English and Japanese - Japanese subtasks. We selected the Indri search engine as a baseline to test our new class of indexing algorithms.

English documents for Indri: Each document was converted to lowercase and written in TREC SGML format. We then indexed the collection using Krovetz stemming and stopword removal.

English documents for NeWT: Each document was converted to lowercase. All non-alphanumeric characters and spaces were left unchanged, with no stemming. The preprocessed English documents were then merged into a single monolithic index.

Japanese documents for Indri: Each document was extracted and converted to UTF8 character codes. Then, word segmentation of documents was performed with ChaSen. The word segmented documents were converted into TREC SGML format. Japanese morphemes in documents were tokenised into terms within Indri.

Japanese documents for NeWT: Each document was converted to UTF8 character codes. Next, all whitespace was removed from each document. The Japanese documents for NeWT were not word segmented. The preprocessed Japanese documents were then merged into the English index and all Japanese and English queries were run against the same index.

Topic Processing

English queries for Indri: All terms from the DESCRIPTION field were used with Krovetz stemming and stopword removal.

English queries for NeWT: All terms from the DESCRIPTION field were used with no stemming or stopword removal. To avoid substring matching, proper nouns were treated as a phrase (e.g. "steve fosset" instead of "steve, fosset"), and acronyms had spaces added before and after the term. (e.g. "␣ana␣")

Japanese queries for Indri: Queries were composed of nouns and substantive non-nouns extracted from the DESCRIPTION field.

Japanese queries for NeWT: The same queries for Indri were used with NeWT, but were not effective because substring matching polluted the ranking results. To avoid substring matching, we performed an n-character suffix (n-suffix) expansion.

Table 1: Regular Expression Examples.

Regular Expression	n-suffixes
('peter piper', 'p.{1}', 'g')	{pe}, {pi}, {pe}
('peter piper', 'p.{2}', 'g')	{pet}, {pip}
('peter piper', 'p.{3}', 'g')	{pete}, {pipe}
('peter piper', 'p.{4}', 'g')	{peter}, {piper}
('peter piper', 'p.{8}', 'g')	{peter pip}

RMIT Runs

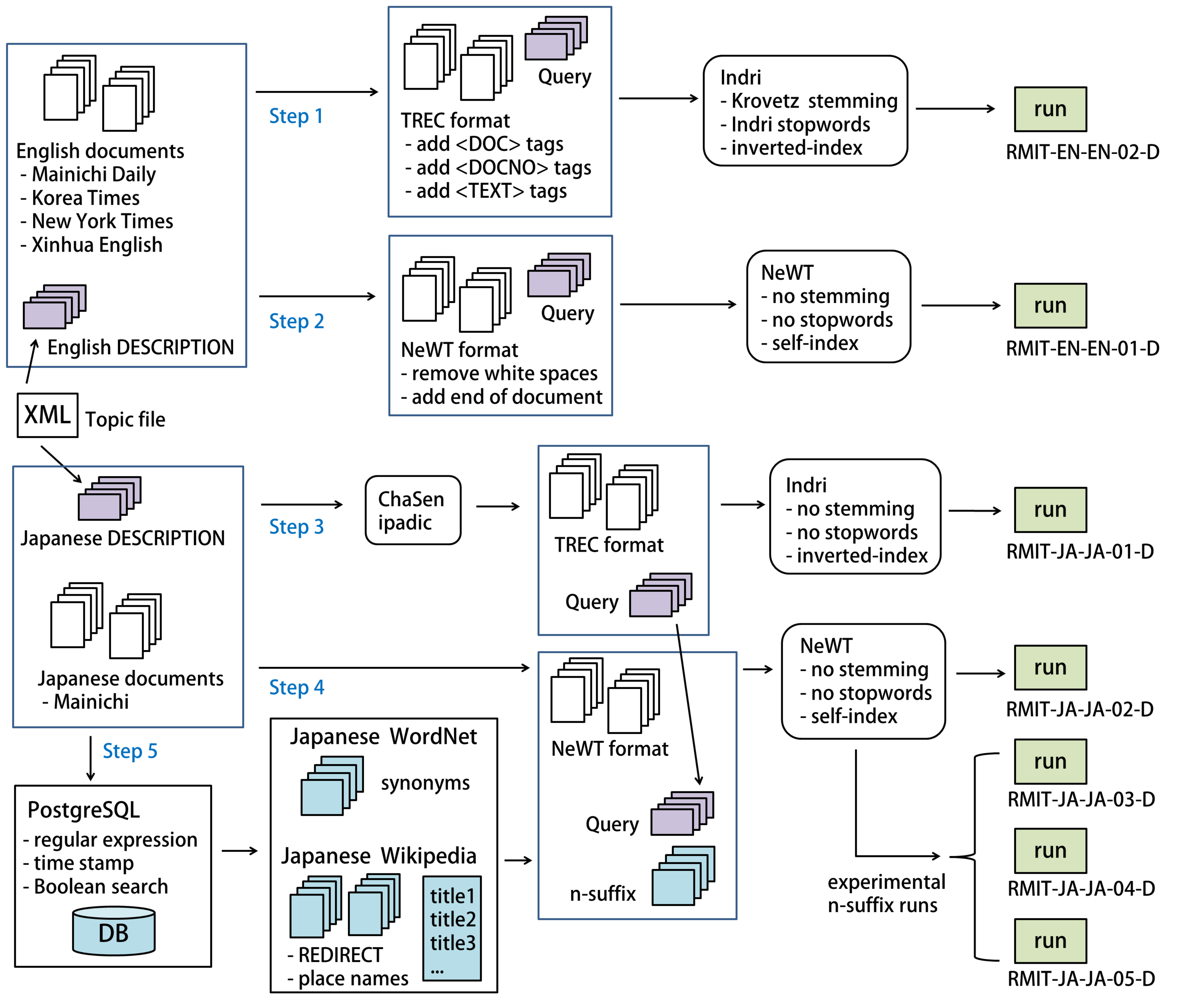

Table 2: Effectiveness Results.

Run	System	Ranking	Preprocess	Expansion	MAP	Q	nDCG@10	@100
EN-01	NeWT	BM25	None	None	0.2477	0.2524	0.4282	0.3691
EN-02	Indri	Dirichlet LM	Krovetz	None	0.2830	0.3057	0.3531	0.3763
JA-01	Indri	Dirichlet LM	ChaSen	None	0.3779	0.4119	0.4769	0.5109
JA-02	NeWT	BM25	None	None	0.3084†	0.3239†	0.3510†	0.3936‡
JA-03	NeWT	BM25	None	2-suffixes	0.3282	0.3349	0.4768	0.4653
JA-04	NeWT	BM25	None	3-suffixes	0.3671	0.3714	0.5230	0.5211
JA-05	NeWT	BM25	None	4-suffixes	0.3376	0.3398	0.4988	0.4841

Ranked Self-Indexing

Two bag-of-words ranking functions were implemented in our experimental search engine, NeWT. NeWT is an enhanced version of the *greedy* top-k approach described by Culpepper et al. [2]. 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.

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

We also implemented a simple BM25 variant as follows:

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

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

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 f_t. Finding the most efficient trade-off is a topic of future work.

Evaluation

In Table 2, † and ‡ indicate statistical significance relative to the baseline run at the 0.05 and 0.001 levels respectively, based on a paired t-test.

English Runs: Compared to the baseline run, the NeWT run is more effective for the highest ranking documents (nDCG@10), but overall effectiveness degrades as the total number of documents retrieved increases (MAP or nDCG@100). Overall, there is no statistically significant difference between the runs.

Japanese Runs: The NeWT run performed worse than the baseline. The runs with 3- and 4-suffix query expansion were more effective than the baseline towards the top of the result list (nDCG@10), but the differences were not statistically significant.

References

[1] G. Navarro and V. Mäkinen. Compressed full-text indexes. *ACM Computing Surveys*, 39(1):2-1 – 2-61, 2007.

[2] 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.