Experiments in the Retrieval of Unsegmented Japanese Text at the NTCIR-2 Workshop

Paul McNamee
Johns Hopkins University Applied Physics Laboratory
11100 Johns Hopkins Road, Laurel MD 20723-6099, USA
mcnamee@jhuapl.edu

Abstract

Our work with the Hopkins Automated Information Retriever for Combing Unstructured Text (HAIRCUT) system has made use of overlapping character n-grams in the indexing and retrieval of text. In previous experiments with Western European languages we have shown that longer length n-grams (e.g., n=6) are capable of providing an effective form of alinguistic term normalization. We have wanted to investigate whether these methods could be adapted to processing unsegmented languages such as Japanese.

To that end we participated in the Japanese and English portion of the NTCIR-2 evaluation. This paper describes results in monolingual Japanese and English retrieval and in cross-language retrieval using each language as a source language for the other.

We found that 6-grams performed comparably with English words and that 2-grams and 3-grams perform equally well in Japanese text. A combination of runs using each tokenization method resulted in only a marginal improvement over runs using a single approach. These two trends were consistent regardless of query length or source language.

Keywords: Japanese text processing, n-grams, information retrieval

1 Introduction

The Hopkins Automated Information Retriever for Combing Unstructured Text (HAIRCUT) is a research retrieval system developed at the Johns Hopkins University Applied Physics Lab (APL). One of the areas that we want to investigate with HAIRCUT is the relative merit of different tokenization schemes. Routinely, overlapping character n-grams and simple words are used as indexing terms. The system also supports multi-word phrases and morphological stemming, however, neither technique is utilized here. A desire to flexibly process text in a large number of languages has motivated a reliance on language neutral techniques.

Our experience in other large-scale evaluations [7] [8] has led us to believe that while n-grams and words are comparable in retrieval performance, a combination of both techniques outperforms the use of a single approach. Accordingly, we indexed the text in multiple ways. For English text both 6-grams and unstemmed words were used while with Japanese text, 2- and 3-grams were used instead. In each of the four tasks that we participated in, we submitted four runs, one using all query fields with both tokenization schemes, one using only the <DESCRIPTION> field with both tokenization schemes, and two runs using all query fields for each tokenization variant.

We had no prior experience in Japanese text processing and no ability to read Japanese, factors that complicated our work. It is doubtful that we could have completed the tasks at all without the tremendous reference on CJKV processing by Ken Lunde [6]. Being unfamiliar with linguistic resources for Japanese, we relied on a single commercial machine translation product for our experiments in cross-language retrieval.

2 Background

Effective text retrieval in Asian languages requires attention to unique problems that arise from the unique linguistic nature of each language. The most fundamental questions of determining what elemental units should be used to represent text and how such units should be identified (i.e., tokenization and normalization) remain a central area of research. No clear consensus seems to exist as to whether word-based methods or n-gram based methods are superior. N-gram based methods are common in Chinese, Japanese, and Korean retrieval, but hybrid techniques have sometimes achieved better performance. Advances in segmentation could impact this trend. In all three languages, the mean word length is approximately two characters, thus bigrams are an obvious and popular choice.

Ogawa and Matsuda have studied a variety of n-gram methods for indexing Japanese text. In one experiment using the BMIR-J1 collection [10], they found that indexing with 2-grams was preferable to indexing with either 1-grams or 3-grams, however a combination of multiple n-grams yielded slightly superior results. In later work using the BMIR-J2 collection [11], they investigated ‘character-class’ n-grams, where certain n-grams are ignored, in particular, n-grams containing hiragana were discarded. The performance of the character-class n-grams was slightly higher than the use of simple n-grams, however, the chief motivation for the method appeared to be the resulting reduction in lexicon and index size.

At the NTCIR-1 workshop [4] several groups examined the role of segmentation and the merits of different approaches to tokenization. Chen et al report that "bigram segmentation of kanji and katakana text fragments outperformed dictionary based segmentation by more than 30%" on monolingual retrieval [2]. Ozawa et al found that an adaptive method of segmentation that produces n-grams of various lengths outperforms simple bigrams [12]. Their hypothesis was that bigrams are insufficient in technical language where word length increases.

3 Experimental Overview

We participated in the Japanese and English monolingual tasks and on the Japanese-to-English and English-to-Japanese bilingual tasks. Four indices were constructed, two for the Japanese subcollection that used 2- and 3-grams respectively, and two for the English subcollection, one using words and one using overlapping 6-grams. In each of the four tasks we participated in, we submitted four runs (XX denotes one of the four tasks, JJ, JE, EE, or EJ):

APLXX1: The <TITLE>, <DESCRIPTION>, <NARRATIVE>, and <CONCEPT> fields were used. The submitted run is a combination of two constituent runs formed using each tokenization method for the target language. (In English, 6-grams and words were used; in Japanese, 2-grams and 3-grams were used.)

APLXX2: Just like APLXX1 but only the <DESCRIPTION> field was used

APLXX3: Like APLXX1, this run uses all of the query fields, however only a single tokenization method is used, either 6-grams (English) or 2-grams (Japanese).

APLXX4: Just like APLXX3, however, words (English) or 3-grams (Japanese) are used.

The method used in APLXX1 and APLXX2 to combine two runs is to first normalize document scores for each topic and then merge the normalized runs.

	Docs	Type	Distinct Terms	Index size (MB)
English	262,058	Words	614,510	202
		6-grams	3,687,005	1,427
Japanese	676,116	2-grams	997,291	2,281
		3-grams	9,161,588	2,904

Table 1. Index statistics for the four indices.

3.1 Index Construction

HAIRCUT is written entirely in Java, a programming language with native support for converting many character encodings to Unicode, however, at the time of the evaluation our system did not use the Java String type internally (though the code has since been changed to do so). Documents in the EUC-JP encoding were processed on the byte level and punctuation was mapped to ISO-8859-1 equivalents. Roman letters were downcased and only the first two of a sequence of digits were preserved (e.g., 1920 became 19##). Only the unsegmented Japanese texts were used and no attempt was made to segment the text. N-grams may span word boundaries (in English the separating space is preserved) but sentence boundaries are noted so that n-grams spanning sentence boundaries are not recorded. Thus n-grams with leading, central, or trailing spaces are formed at word boundaries

When words were used no stemming or stopword removal was performed. As can be seen from Table 1, the use of 6-grams as indexing terms increases both the size of the inverted file (~600% increase) and the dictionary (~500% increase) compared to the corresponding word index.

3.2 Query Processing

Normally HAIRCUT performs rudimentary preprocessing on queries to remove stop structure, *e.g.*, affixes such as "… would be relevant" or "relevant documents should….", however, we did not have a convenient method for identifying and removing stop structure in the Japanese queries. Therefore, stop structure was removed only for English queries. The topics statements were tokenized in the same manner as the documents being retrieved.

In all of our experiments we used a retrieval model motivated by work in statistical language modeling [3] [13]. This approach has also been cast as a simple two-state hidden Markov model that captures both document and collection statistics [9]. After the query is parsed each term is weighted by the query term frequency and an initial retrieval is performed followed by a single round of relevance feedback. The calculation that is performed is:

$$Sim(q,d) = \prod_{t=terms} \left(a \cdot f(t,d) + (1-a) \cdot df(t) \right)^{f(t,q)}$$

Equation 1. A language-inspired similarity metric.

where *f(t,d)* is the frequency of term *t* in document *d* and *df(t)* denotes the document frequency of *t*. a is the probability that a term is generated by a model based on a single document instead of a model based on the language in general.

To perform relevance feedback an initial retrieval is performed to identify the top ranked 1000 documents.

The top 20 documents are used for positive feedback and the bottom 75 documents are used for negative feedback, however, no duplicate or neo-duplicate documents are included in these sets. Then terms for the expanded query are selected based on three factors, a term's initial query term frequency (if any), the (α=3, β=2, γ=2) Rocchio score, and a metric that incorporates an idf component. The top-scoring terms are then used as the revised query. Because we suspect that different tokenization methods may possess different discriminating ability, a different number of expansion terms was used for each method. The parameters that vary for each method are shown in Table 2.

Method	# Top Terms	Alpha
Words	60	0.30
6-grams	400	0.15
2-grams	100	0.23
3-grams	400	0.15

Table 2. Parameters used for different indices.

The experiments were conducted on a 4-node Sun Microsystems Ultra Enterprise 450 server. The workstation had 2.5 GB of physical memory and access to 100 GB of dedicated hard disk space. The HAIRCUT system comprises approximately 25,000 lines of Java code.

4 Monolingual Experiments

Table 3 summarizes the performance of our official monolingual runs using the strict relevance criteria. Comparisons to median and top score are based on the complete set of automatic runs.

Run	Avg-Prec	Recall	# best	# ≥ median
apljj1	0.3597	2446	2	37
apljj2	0.2800	2086	0	21
apljj3	0.3362	2409	2	30
apljj4	0.3399	2351	0	32
aplee1	0.2594	1024	4	28
aplee2	0.1955	820	0	13
aplee3	0.2481	987	4	25
aplee4	0.2316	969	1	26

Table 3. Official monolingual runs.

Figures 1 and 2 display Precision-Recall graphs for the monolingual runs. There is significant symmetry between the two. Not surprisingly, of the four types of runs, the worst-performer is the one using the shortest topic statements. More interesting is the fact that in both tasks, a combination of runs using different indexing terms achieves better recall and average precision than the best run using a single type of term. Finally, we observe that the different indexing methods are comparable and in particular, 2-grams and 3-grams performed equally well for Japanese retrieval. This is interesting since different results have been reported for Chinese and Korean text retrieval[1], [5].

This trend was also observed in the constituent runs that were merged to produce APLJJ2 and APLEE2, so this is not simply a feature obtained on the longer (i.e., easier) topic statements. Figure 3 shows the topic-by-topic variability for the different runs.

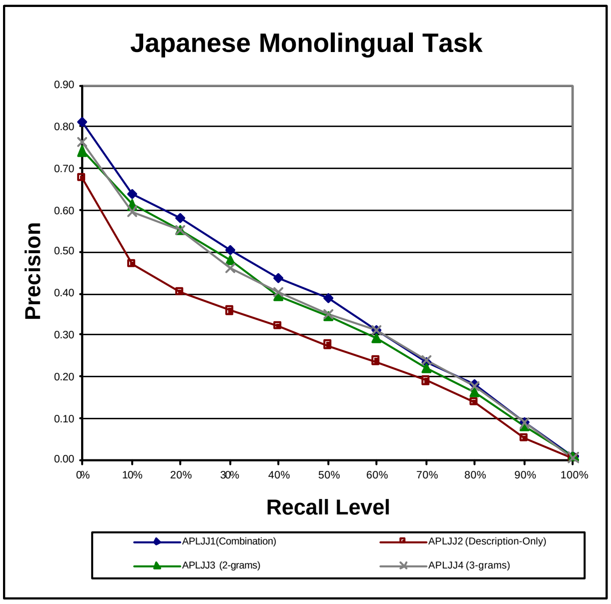


Figure 1. Comparison of Japanese monolingual retrieval performance under four different conditions.

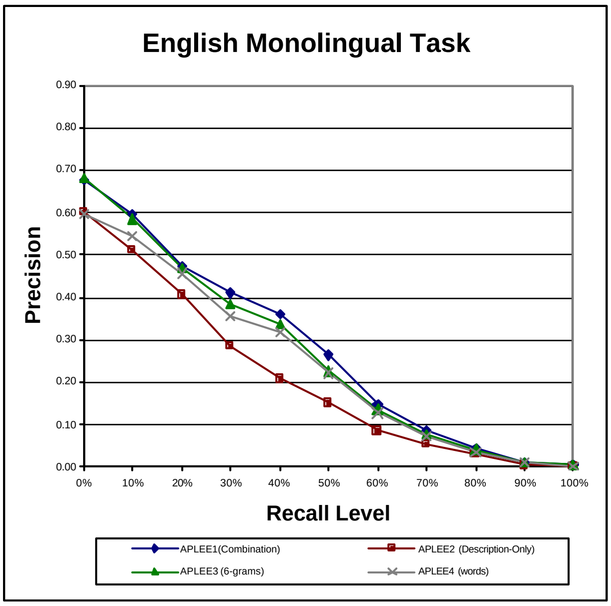
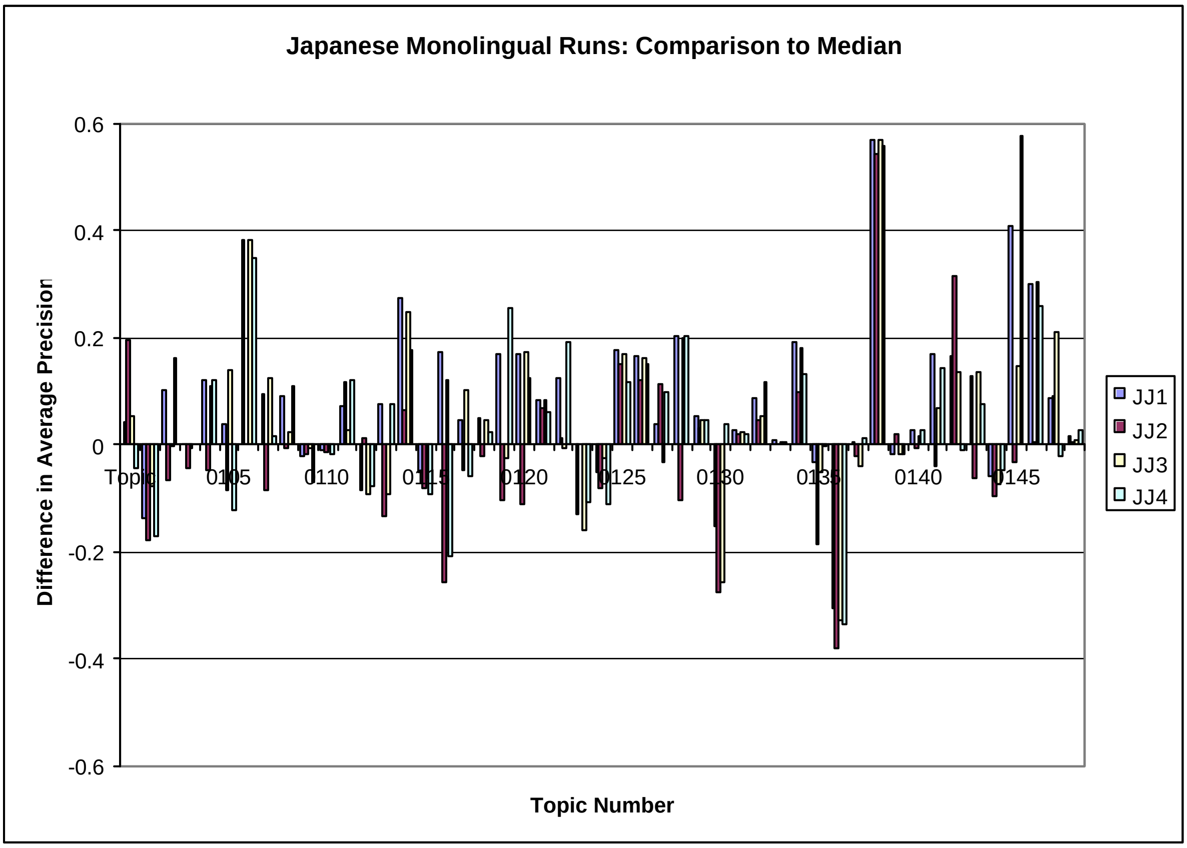


Figure 2. Comparison of English monolingual retrieval performance under four different conditions.



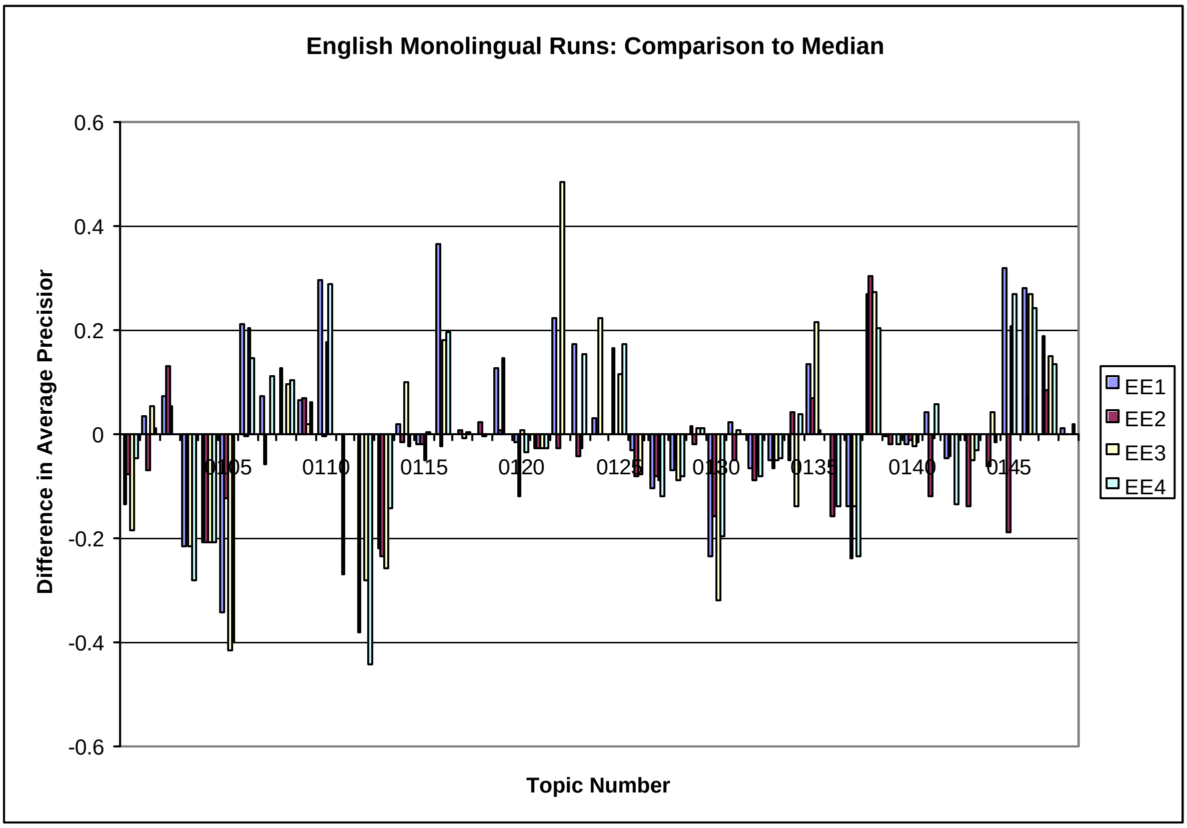
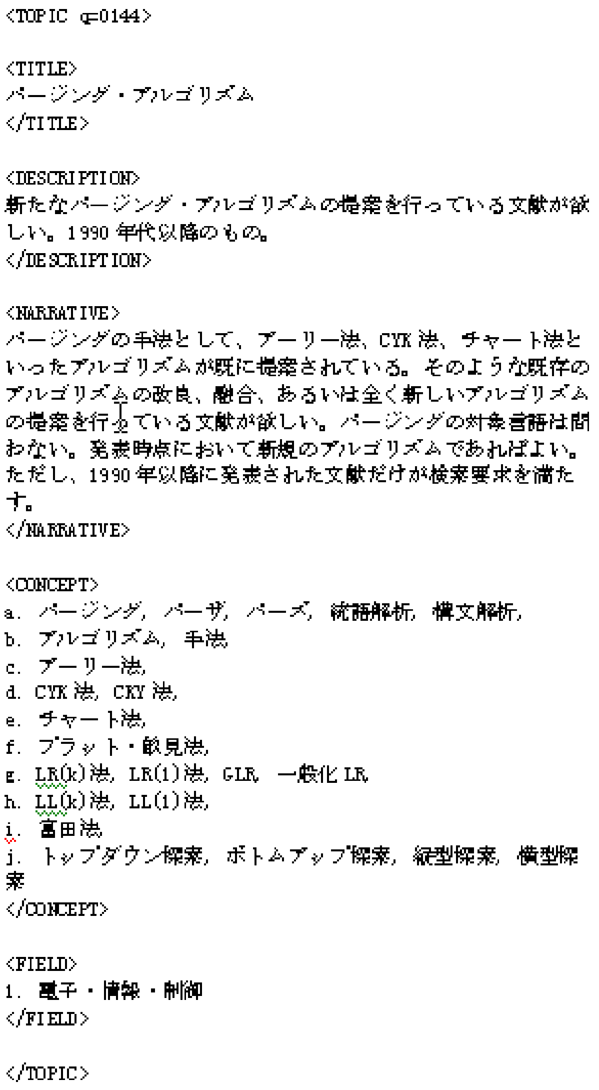


Figure 3. Topic-by-topic performance of official monolingual runs. The ‘all-field’ runs outperform the ‘description-only’ runs (JJ2/EE2). Also, when the single tokenization methods (JJ3/2-grams vs. JJ4/3-grams, and EE3/6-grams vs. EE4/words) are compared, they are found to achieve similar performance on a given topic.

```
<TOPIC q=0144>

<TITLE>
Parsing algorithm
</TITLE>

<DESCRIPTION>
Papers published in 1990's in which a new
parsing algorithm is proposed
</DESCRIPTION>

<NARRATIVE>
Many parsing algorithms such as the Earley
method, the CYK algorithm, and the chart
algorithm have been proposed. I want papers
that modify these existing algorithms, or
propose a completely original algorithm. The
language parsed by the algorithm is not
specified. Papers whose algorithm was original
at the time of publishing satisfy this
retrieval request. I want papers published
since 1990.
</NARRATIVE>

<CONCEPT>
a. parsing, parser, parse, syntactic analysis,
b. algorithm, method,
c. Earley method,
d. CYK algorithm, CKY algorithm,
e. chart algorithm,
f. Pratt-Unemi's algorithm,
g. LR(k) algorithm, LR(1) algorithm, GLR,
generalized LR,
h. LL(k) algorithm, LL(1) algorithm,
i. Tomita's algorithm,
j. top-down search, bottom-up search, depth-
first search, breadth-first search
</CONCEPT>

<FIELD>
1. Electricity, information and control
</FIELD>

</TOPIC>
```



```
<TOPIC q=0144>

<TITLE>
パージング・アルゴリズム
</TITLE>

<DESCRIPTION>
新たなパージング・アルゴリズムの提案を行っている文献が欲
しい。1990 年代以降のもの。
</DESCRIPTION>

<NARRATIVE>
パージングの手法として、アーリー法、CYK 法、チャート法と
いったアルゴリズムが既に提案されている。そのような既存の
アルゴリズムの改良、融合、あるいは全く新しいアルゴリズム
の提案を行っている文献が欲しい。パージングの対象言語は問
わない。発表時点において新規のアルゴリズムであればよい。
ただし、1990 年以降に発表された文献だけが検索要求を満た
す。
</NARRATIVE>

<CONCEPT>
a. パージング, パーザ, パーズ, 統語解析, 構文解析,
b. アルゴリズム, 手法
c. アーリー法,
d. CYK 法, CKY 法,
e. チャート法,
f. プラット・畝見法,
g. LR(k)法, LR(1)法, GLR, 一般化 LR,
h. LL(k)法, LL(1)法,
i. 富田法
j. トップダウン探索, ボトムアップ探索, 縦型探索, 横型探
索
</CONCEPT>

<FIELD>
1. 電子・情報・制御
</FIELD>

</TOPIC>
```

Figure 4. A sample topic statement.

5 Bilingual Experiments

Table 4 summarizes the performance of our official bilingual runs using the strict relevance criteria. Comparisons to median and top score are based on the complete set of automatic runs.

Run	Avg-Prec	Recall	# best	# ≥ median	% mono
aplje1	0.1388	633	0	19	53.51%
aplje2	0.0814	455	0	9	41.63%
aplje3	0.1279	622	0	17	51.55%
aplje4	0.1095	566	1	14	47.28%
aplej1	0.1414	1572	1	25	39.31%
aplej2	0.0945	1227	1	18	33.75%
aplej3	0.1331	1520	1	24	39.59%
aplej4	0.1336	1308	1	26	39.31%

Table 4. Official cross-language runs.

For our cross-language experiments we used the Systran™ machine translation product (which supports both English to Japanese and Japanese to English conversions) to translate topic statements. The performance measures for our cross-language runs are appreciably below the corresponding monolingual runs. To understand why our cross-language results fell below our expectations we examined some individual topic translations. We observed that the translations involving katakana are frequently incorrect. For example, words from Topic 144 such as *parsing*, *Earley*, and *algorithm* were translated to the phonetically similar ‘purging’, ‘early’, and ‘a rhythm’.

This can also explain why we found worse relative bilingual performance when working from English queries to Japanese documents rather than from Japanese queries to English documents (see Table 4 and Figure 6). Since the number of English words is smaller than the number of comparable katakana

```
<TOPIC q=0144>
<TITLE>
Purge algorithm

<DESCRIPTION>
The of new purge algorithm we want the
literature which is 1990 thing.

<NARRATIVE>
As technique of the purge the algorithm such
as Early method, CYK method and chart method
already improvement, fusion or the of the of
that kind of previous  A rhythm of B new
algorithm we want the literature which is.
If it should have been a new algorithm B at
the time of the of the purge in the. However,
1990 just the literature which the is done
satisfies retrieval request after N.

<CONCEPT>
A.Purge per the, per, ¥ syntax analysis,
B.Algorithm, technique,
C.Early method,
D.CYK method and CKY method,
E.Chart method,
F.Pratt & ridge seeing method,
G.LR (K) law, LR (1) law, GLR, one R
H.LL (K) law, LL (1) law,
I.Wealth C method,
J.Top Dow Jones T cord, bottom-up T cord,
vertical die T cord and horizontal T cord

<FIELD>
1.D child & information & control
```

Figure 5. Translation of Topic 144 from Japanese. Underlined words are phonetic errors.

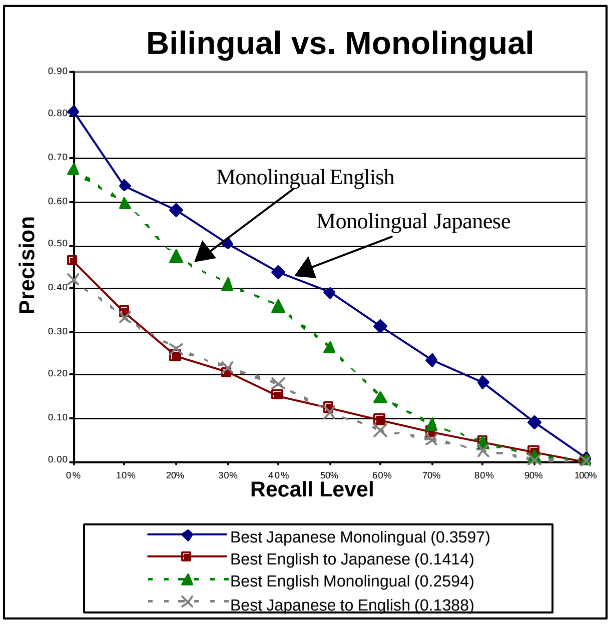


Figure 6. Bilingual performance compared to monolingual baselines.

transliterations, it is more difficult to pick the most precise katakana when translating an unknown word, than it is to pick the right word from katakana. The use of better bilingual dictionaries or context-dependent translation could reduce the number of this type of error.

6 Conclusions

We were encouraged by our first attempt at Japanese language retrieval using the HAIRCUT system. Though unfamiliar with the Japanese language we have shown that uninformed methods of segmentation and tokenization can be effective, though there is clearly room for improvement. The individual performance of unstemmed words and character 6-grams was comparable in monolingual English retrieval. And more surprisingly, 2- and 3-grams were found to be comparable in Japanese. A marginal improvement was found by merging ranked document lists obtained with different indexing methods.

Our experiments in bilingual Japanese/English retrieval relied on a single machine translation product, a fact that may explain relatively low performance compared to a monolingual baseline. The NTCIR-2 workshop has provided the opportunity to examine unique and critical problems in Japanese text indexing and retrieval. In the future we hope to revisit these experiments using a larger suite of resources to perform query translation.

References

[1] A. Chen, J. He, L. Xu, F. C. Gey, and J. Meggs, ‘Chinese Text Retrieval Without Using a Dictionary’. In the *Proceedings of the 20th International Conference on Research and Development in Information Retrieval (SIGIR-97)*, pp. 42-49, July 1997.

[2] A. Chen, F. C. Gey, K. Kishida, H. Jiang, and Q. Liang, ‘Comparing Multiple methods for Japanese and Japanese-English text retrieval’. At the *First NTCIR Workshop on Research in Text Retrieval and Term Recognition (NTCIR-1)*, 1999.

[3] D. Hiemstra and A. de Vries, ‘Relating the new language models of information retrieval to the traditional retrieval models.’ CTIT Technical Report TR-CTIT-00-09, May 2000.

[4] N. Kando, K. Kuriyama, and T. Nozue, ‘NACSIS Test Collection Workshop (NTCIR-1)’. In the *Proceedings of the 22nd International Conference on Research and Development in Information Retrieval (SIGIR-99)*, August 1999.

[5] J. H. Lee, H. Y. Cho, and H. R. Park, ‘n-Gram-based indexing for Korean text retrieval’. In *Information Processing & Management*, 35(1), pp. 427-441, 1999.

[6] K. Lunde, *CJKV Information Processing,* O'Reilly & Associates, January 1999.

[7] J. Mayfield, P. McNamee, and C. Piatko, 'The JHU/APL HAIRCUT System at TREC-8.' In E. M. Voorhees and D. K. Harman, eds., *Proceedings of the Eighth Text REtrieval Conference (TREC-8),* 2000.

[8] P. McNamee, J. Mayfield, and C. Piatko, 'A Language-Independent Approach to European Text Retrieval.' Draft version in the *Working Notes of the CLEF-2000 Workshop,* Lisbon, Portugal, September 2000.

[9] D. R. H. Miller, T. Leek, and R. M. Schwartz, 'A Hidden Markov Model Information Retrieval System.' In the *Proceedings of the 22nd International Conference on Research and Development in Information Retrieval (SIGIR-99),* pp. 214-221, August 1999.

[10] Y. Ogawa and T. Matsuda, 'Overlapping statistical word indexing: A new indexing method for Japanese text'. In the *Proceedings of the 20th International Conference on Research and Development in Information Retrieval (SIGIR-97*), pp. 226-234, July 1997.

[11] Y. Ogawa and T. Matsuda, 'Overlapping statistical segmentation for effective indexing of Japanese text'. In *Information Processing & Management*, 35(1), pp. 463-480, 1999.

[12] T. Ozawa, M. Yamamoto, K. Umemura, and K. W. Church, 'Japanese word segmentation using similarity measure for IR'. At the *First NTCIR Workshop on Research in Text Retrieval and Term Recognition (NTCIR-1),* 1999.

[13] J. Ponte and W. B. Croft, 'A Language Modeling Approach to Information Retrieval.' In the *Proceedings of the 21st International Conference on Research and Development in Information Retrieval (SIGIR-98),* pp. 275-281, August 1998.