Language Independent Ranked Retrieval with NeWT

J. Shane Culpepper
RMIT University
Victoria, Australia 3000
shane.culpepper@rmit.edu.au

Michiko Yasukawa
Gunma University
Kiryu, Japan 376-8515
michi@cs.gunma-u.ac.jp

Falk Scholer
RMIT University
Victoria, Australia 3000
falk.scholer@rmit.edu.au

Abstract *In this paper, we present a novel approach to language independent, ranked document retrieval using our new self-index search engine,* NeWT. *To our knowledge, this is the first experimental study of ranked self-indexing for multilingual Information Retrieval tasks. We evaluate the query effectiveness of our indexes using Japanese and English. We explore the impact that linguistic processing, stemming and stopping have on our character-aligned indexes, and present advantages and challenges discovered during our initial evaluation.*

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

1 Introduction

In this paper, we present a new self-index search engine, NeWT, which implements a novel approach to language independent, ranked document retrieval. NeWT is a hybrid search engine derived from an FM-*index* [Ferragina and Manzini, 2000] for fast substring matching, and a *wavelet tree* [Grossi et al., 2003] to support document level ranking. To our knowledge, this is the first comprehensive evaluation of ranked self-indexes using multilingual document collections.

We focus on an unexplored key aspect of supporting traditional ranked document retrieval using self-indexes – bag-of-words queries. Past work on ranked document retrieval in self-indexes has focused primarily on top-k phrase querying using simple TF×IDF similarity metrics [Hon et al., 2009, Culpepper et al., 2010]. In previous self-indexing approaches, the top-k documents are retrieved using raw frequency counts. While top-k retrieval using raw counts is a valuable first step to developing new approaches for efficient and effective ranked document retrieval, these methods have never been properly evaluated for traditional retrieval effectiveness on any reasonably sized document collection.

We extend the work of Culpepper et al. [2010] to support the Okapi BM25 [Robertson et al., 1994] similarity metric along with generic bag-of-words querying capabilities. We also perform the first comprehensive ranking evaluation using three different test collections: the TREC 7 & 8 *ad hoc* collection, the NTCIR-9 GeoTime collection of English and Japanese documents, and the JA-Category subset of the ClueWeb09 collection, as described in the NTCIR-9 INTENT track [Song et al., 2011]. As part of our evaluation, we expose and explore an interesting substring problem unique to character-aligned bag-of-words querying approaches.

Our Contributions. We present the first self-index capable of supporting language independent bag-of-words queries in a top-k ranked document framework. We provide the first comparative evaluation of self-indexes for top-k ranked retrieval against state-of-the-art inverted indexes on the TREC and NTCIR data collections.

We also explore the viability of constructing a monolithic multilingual index capable of delaying linguistic and morphological processing until *query time*. This is an important first step in building a new class of indexes for ranked text retrieval that make no a priori assumptions about the use of the text at indexing time, enabling decisions about complex linguistic processing to be deferred until query time.

2 Background

2.1 Inverted Indexes

For bag-of-words information retrieval tasks, indexing and querying is usually accomplished us-

Proceedings of the 16th Australasian Document Computing Symposium, Canberra, Australia, December 2, 2011.

ing an inverted index [Zobel and Moffat, 2006]. An inverted index contains a vocabulary of *terms* mapped to a list of tuples containing document and frequency data for the entire text collection. A term can be anything, but for simplicity we consider each term to be a single word in the linguistic sense.

For English text, stemming, stopping, case folding, and removing diacritics are standard steps in text preprocessing for inverted files [Witten et al., 1999, Büttcher et al., 2010, Croft et al., 2010], and is always done at indexing time. However, this preprocessing requires specific domain knowledge to be applied *at indexing time*. A considerable body of literature now exists on the advantages and disadvantages of different preprocessing techniques. For instance, stemming [Lovins, 1968, Porter, 1980, Frakes, 1984, Paice, 1990, Krovetz, 1993] of English text tends to help for large collections [Hull, 1996], but not necessarily in smaller collections [Harman, 1991]. The conventional wisdom on using stop words for English text is also context dependent. For some queries, using stop words can dramatically increase effectiveness, but for others stop words have a negative impact. Retaining stop words is also important for phrase queries [Witten et al., 1999, Manning et al., 2008].

In CJKV languages, segmenting text into terms is not always possible since words are not space delimited. Instead of mapping a single word in a language to a term, an alternative approach is to construct inverted indexes using character n-grams. This approach has a long history in Information Retrieval, particularly for language independent indexing [Burnett et al., 1979, Willett, 1979, de Heer, 1982, Hollink et al., 2004]. The general idea of n-grams is to index overlapping substrings of fixed length, and support bag-of-words querying using the same merging or intersection techniques as inverted files. One advantage of this approach is that language-specific parsing can be avoided at index time, and multiple languages can be supported simultaneously. However, the effectiveness of this approach is strongly dependent on the underlying language of the document [McNamee and Mayfield, 2004, McNamee et al., 2008]. In particular, the retrieval effectiveness in English text using n-grams is mixed, and often worse than term-based approaches, limiting widespread adoption [Büttcher et al., 2010].

English and many European languages can be described as *space-delimited languages*, wherein words are separated by white space. In English, a term in an inverted index is simply a word, which is a sequence of characters preceded and followed by whitespace [Dale et al., 2000]. However, east Asian languages such as Chinese, Japanese, Korean, and Vietnamese (CJKV languages), and agglutinated European languages such as Finnish or Turkish can be classified as *unsegmented languages* wherein words are not neatly delimited by spaces.

In Japanese, text is mainly composed of Kanji (Chinese characters) with Hiragana syllables for inflectional endings and function words [Manning et al., 2008]. Because words in Japanese are written in succession, with no indication of word boundaries, the word segmentation process is generally performed on Japanese documents by using morphological analyzers such as ChaSen[1] and MeCab.[2] 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 [Yasukawa and Yokoo, 2010]. 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. In newspaper articles, unique proper nouns are commonplace. In patent documents, uncommon compound words are widely used to describe a developed technology. When wrongly segmented morphemes are stored as index terms, search results are often poor.

A considerable body of research comparing the trade-offs when applying n-gram or term-based methods preprocessing to Chinese exists (see, for instance, Kwok [1997], Nie et al. [2000], Luk and Kwok [2002]). Generally, n-gram based methods increase recall, while word-based methods improve precision [Luk and Kwok, 2002]. However, recent work has focused primarily on improving word segmentation using morphological analysis, since character and n-gram indexes can impose a significant space overhead.

2.2 Self-Indexes

Inverted indexes are a robust solution for search problems dealing with readily-parsable natural language text [Zobel and Moffat, 2006]. However, forcing unsegmented languages to rely on morphological analysis is not always the best option. Recently, *self-indexing* algorithms have emerged as a viable alternative to inverted indexes [Navarro and Mäkinen, 2007]. Self-indexing algorithms are capable of supporting character level matching similar to an n-gram index, but with the same bounded efficiency as a suffix array. Significant performance gains using these new data structures for basic pattern matching are possible, but efficiently supporting ranked queries on massive data sets using these novel

[1] http://chasen-legacy.sourceforge.jp/
[2] http://mecab.sourceforge.net/

data structures is still relatively unexplored [Hon et al., 2010]. Self-indexes use space close to what can be obtained by the best possible data compression algorithms. Previous work on self-indexes has focused predominantly on classical pattern matching problems. Self-indexes are extraordinarily efficient for simple pattern matching operations, but are capable of much more.

In this paper, we investigate the problem of using self-indexing algorithms to solve the **ranked document search** problem. The document collection $\mathcal{T}$ is a contiguous string drawn from an alphabet Σ^*, where $\sigma = |\Sigma^*|$. Each document in $\mathcal{T}$ is separated by a unique end of document symbol. In practice, Σ^* can be characters (UTF8 or ASCII), bytes, integers, or even terms.

Definition 1 *A* **ranked document search** *takes a query $q \in \Sigma^*$, an integer $0 < k \leq d$, and a text $\mathcal{T} \in \Sigma^*$ that is partitioned into d documents $\{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_d\}$, and returns the top-$k$ documents ordered by a similarity measure $\hat{S}(q, d_i)$.*

In order to solve the **document search problem**, unadorned self-indexing algorithms are not sufficient. The index also requires a *document array* – a mapping between every suffix in $\mathcal{T}$ to a corresponding document identifier [Mithukrishnan, 2002, Välimäki and Mäkinen, 2007, Culpepper et al., 2010]. By representing the document array with a wavelet tree, all occurrences of a substring $\mathcal{P}$ in each distinct document, and across the whole collection, can be counted in $\mathcal{O}(\log u)$ time, where $u \leq min(\sigma, d)$ using $\mathcal{O}(nH_k(\mathcal{T}) + n \log d) + o(n \log \sigma) + o(n \log d)$ bits of space. Here, $H_k(\mathcal{T})$ represents the kth order empirical entropy of the compressed text. The document and collection occurrence counts can then used to calculate TF×IDF based $\hat{S}(q, d_i)$ metrics at query time. A comprehensive discussion of prior work on applications of self-indexes to the **ranked document search** problem is beyond the scope of this paper. In this work, we use an enhanced version of the *greedy* top-k approach described in Culpepper et al. [2010].

2.3 Ranking in Self-Indexes

Effective and efficient document ranking in a self-index is an interesting open problem. For succinctness, we use "term" to represent any word, substring, or phrase as self-indexing methods can support any character sequence. All prior published work on ranked self-indexes uses a trivial TF×IDF ranking metric whereby the absolute term, substring, or phrase frequency is calculated, along with the number of distinct documents in which the term, substring, or phrase appears. Ranked self-indexes are capable of searching for a phrase, a bag of words, or a bag of substrings with equal ease.

For our experiments, our $\hat{S}(q, d_i)$ ranking function is a variant of BM25. To our knowledge this is the first ranked document self-index using BM25. We have began experimenting with other ranking models such as Dirichlet language model ranking, but BM25 is simple and effective for bag-of-words queries. Our $\hat{S}(q, d_i)$ ranking function has the following formulation:

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

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

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 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 *independent and identically distributed* assumptions, as topics for future work.

3 Experimental Framework

3.1 Text Collections

TREC 7 and 8 *Ad Hoc*. We conduct our English language experiments using the TREC 7 and 8 *ad hoc* collections. These use around 2 GB of newswire data from the *Financial Times*, *Federal Register*, *LA Times*, and *Foreign Broadcast Information Service*, consisting of around 528,000 documents in total [Voorhees and Harman, 1999]. A total of 100 test topics and associated relevance judgments are available for these collections (topic numbers 351–400 for TREC 7, and 401–450 for TREC 8). In our experiments, we use the *title* field of each topic; this gives a short query consisting of 2 to 3 keywords, on average, and is representative of the kinds of queries that users enter into popular web search engines.

NTCIR 2011 GeoTime. The NTCIR GeoTime task supports the investigation of search based on geographic and temporal constraints. Each information need therefore includes both time and space aspects. The GeoTime 2011 task used two document collections: English and Japanese. 25 information needs were made available in both

languages, enabling the investigation of retrieval for the same queries in different languages, as well as the potential for cross-lingual retrieval [Gey et al., 2011]. The English collection consists of newswire data from the *New York Times*, *Korea Times*, *Mainichi*, and *Xinhua*, together covering a time period from 1998 to 2005, and amounting to around 800,000 documents. The Japanese collection consists of newswire data from *Mainichi*, also covering a time period from 1998 to 2005, and including around 800,000 documents.

NTCIR 2011 INTENT in Japanese. One of the subtasks of the NTCIR 2011 INTENT task was Document Ranking in Japanese [Song et al., 2011]. The Japanese collection in the task consists of the Japanese segment of the ClueWeb09 collection[3], which entails 67.3 million documents. A total of 100 test topics and associated relevance judgments are available for the collection (topic numbers 0101–0200 for NTCIR 2011 INTENT).

3.2 Query Processing

In order to test the effectiveness of bag-of-words queries, the same query processing method was used for each collection. We use the Indri Search Engine[4] as the state-of-the-art baseline for our experiments. When stemming is required, we use Krovetz stemming in order to remain consistent with the Indri baseline [Krovetz, 1993]. English queries for NEWT and Indri were initially treated as a bag-of-words with no stemming or stopword removal. Each word was treated as a single term. The Japanese queries were initially partitioned into terms using ChaSen/MeCab with the standard IPA dictionary[5]. Our preliminary testing led to the discovery of an unexpected issue, which we will refer to as the *substring problem*.

The substring problem. The preliminary evaluation of simple Japanese and English queries in the test collections resulted in unusually poor effectiveness. Further investigation revealed a problem with both query sets, which is rather obvious in hindsight. Short query terms in English or Japanese are also substrings of other terms. These false matches degrade the discriminatory power of certain terms, and the problem becomes more pronounced with longer queries. The longer the query, the more likely at least one of the terms is a substring of another common term in the collection. For example, the acronym "ana" (All Nippon Airways) found in the GeoTime-034 query is also a substring of the word "analytic", and, worse, it is an overlapping substring in words like "banana / bananna". This problem can be somewhat mitigated using whitespace padding, as we will see shortly.

[3]http://lemurproject.org/clueweb09/
[4]http://sourceforge.net/projects/lemur/
[5]http://sourceforge.jp/projects/ipadic/

However, there is no white space in the Japanese collection used in NEWT, so word boundaries are particularly problematic in the Japanese collection. Consider the following example scenario. The Japanese word "ナス" meaning "eggplant" is the same as the middle of the word "バナナスムージー" meaning "banana-smoothie". Consequently, search results with NEWT are intermingled with documents that are not relevant, but were coincident with the short words in queries. This problem is further exacerbated by the fact that short substrings in Japanese queries can traverse word boundaries since there are no spaces in the index.

Conversely, substring matches in Japanese texts can also have advantages over morphological parsing. For example, suppose that document X contains the Japanese word "バナナスムージー" meaning "banana-smoothie," and document Y contains the Japanese word "ストロベリースムージー" meaning "strawberry-smoothie". Because the Japanese Katakana loan word "スムージー" meaning "smoothie" is not included in the standard dictionary for morphological analyzers, both compound words would not be segmented and become terms in an inverted index. Under this circumstance, the query term "スムージー" meaning "smoothie" finds neither document X nor document Y, because the query term is not an exact match to the terms in inverted indexes. On the other hand, in NEWT, the short query substring will correctly match any documents that contain "smoothie," even as part of a compound word. Moreover, a document for "スムージー の 材料" meaning "ingredients for smoothie" is also found in self-indexes without any analysis or parsing of the sentence being performed. However, it is not clear if these substring matches overcome the rank pollution imposed by cross-term or intra-term matching of very short substrings.

English query processing. In order to circumvent the substring problem in the English query set, we experiment with term padding, whereby each term in the query string uses additional whitespace:

1. *prefix* - add a single whitespace at the beginning of each query term.
2. *suffix* - add a single whitespace at the end of each query term.
3. *space* - add a single whitespace at the beginning and end of each query term.

We investigate each of these query expansions in conjunction with a mixture of the four different indexing strategies: stemming, nostemming, stopping, and nostopping. Each of these alternatives is discussed further in Section 4. These modifications significantly improve the effectiveness of

NEWT on the TREC collection, but the results were still lackluster for the GeoTime collection. The problem with the GeoTime query set seems to be that each topic is phrased as a question, instead of a set of keywords. For example, the GeoTime-026 topic is "Where and when did the space shuttle Columbia disaster take place?". Since many of the words in the topic description are not relevant to the query, we generated a manual set of keyword queries using only a subset of important words from each topic. So, GeoTime-026 was reduced to "space,shuttle,columbia,disaster". These keyword queries are then run with Indri and NEWT and compared explicitly.

Japanese query processing. To examine the substring problem in the Japanese query set, we evaluate the impact of morphological analysis on effectiveness, and then test four possibilities:

1. ChaSen/MeCab word segmentation on both documents and queries.
2. ChaSen/MeCab word segmentation on documents and no word segmentation on queries.
3. ChaSen/MeCab word segmentation on queries and no word segmentation in documents.
4. No word segmentation on documents or queries.

For the GeoTime collection, bag-of-words queries from the topic description and a manual set of keyword queries were prepared as described above for the English queries. For the JA-Category subset of the ClueWeb09 collection, we used the official Japanese topics in the NTCIR-9 INTENT task. Since those topics were selected from the user query log of a commercial search engine, no manual keyword selection was necessary.

4 Results

4.1 English Newswire

Text preprocessing at indexing time. A variety of text preprocessing approaches are commonly applied to raw natural language text with the aim of improving subsequent retrieval from a collection. We investigate the impact of applying stopping and stemming, at indexing time, for both the the inverted index (Indri) and self-index (NEWT) approaches. The results are shown in Table 1, with performance measured using precision at 10 documents retrieved, mean average precision, and *normalized discounted cummulative gain* (nDCG) for the TREC 7 and 8 newswire collections.

In our experiments, the best text preprocessing options for Indri are to apply stemming but not stopping; for NEWT, the best performance is achieved by applying both stemming and stopping. These correspond to rows 3 and 8 in Table 1. The difference between the results is not statistically significant (paired t-test, $p > 0.1$).

Query processing. A key potential advantage presented by self-indexing approaches is the opportunity for flexible query processing at retrieval time, without the requirement of collection preprocessing at indexing time. As explained in Section 3.2, by default NEWT will match *any* substring that corresponds to a query term. While this can be useful, it may also lead to unexpected outcomes for short query terms. We therefore compare the default NEWT method (*plain*) with three other query processing techniques: *prefix*, *suffix*, and *space* (see Section 3.2 for a full description). The results for the TREC 7 and 8 collections are shown in Table 2. Note that, as the aim is to avoid the need to commit to any particular type of preprocessing at indexing time, all results are based on the original collection (no stemming or stopping).

The Indri and NEWT *plain* runs are of roughly similar effectiveness. Adding *prefix* or *suffix* query processing increases performance slightly compared to *plain*. However, *space* query processing leads to a substantial boost, with statistically significant improvements for all three effectiveness measure for the TREC 7 data (paired t-test, $p < 0.05$), and a significant improvement for nDCG on the TREC 8 data.

4.2 GeoTime

English GeoTime. We compare the inverted index and self-index approaches on a more complex family of queries – with geographic and temporal constraints – through experiments on the NTCIR 2011 GeoTime English task. The results are shown in Table 3, as measured by mean average precision, mean reciprocal rank, and normalised discounted cumulative gain at cutoff level 10 as well as over the full ranked list.

The performance of Indri and NEWT is comparable over the *full* GeoTime 2011 queries when queries are processed using the *space* approach. Specifically, there are no statistically significant differences when using the *suffix* or *space* query processing approaches, while using the *plain* or *prefix* query processing approached leads to significantly worse performance on the nDCG measures (paired t-test, $p < 0.05$).

As the queries for the GeoTime task are very verbose and include many instruction words, we also carried out experiments with these queries when they are reduced to keywords only (see Section 3.2). Removing the instruction words leads to substantially higher performance for both the inverted index and self-index approaches

System	Preprocessing	TREC 7 P@10	TREC 7 MAP	TREC 7 nDCG	TREC 8 P@10	TREC 8 MAP	TREC 8 nDCG
Indri	nostem, nostop	0.3760	0.1570	0.3885	0.4280	0.1938	0.4398
Indri	nostem, stop	0.3760	0.1570	0.3885	0.4220	0.1917	0.4367
Indri	stem, nostop	0.4000	0.1717	0.4185	0.4520	0.2256	0.4948
Indri	stem, stop	0.3960	0.1685	0.4153	0.4420	0.2230	0.4917
NEWT	nostem, nostop	0.3740	0.1711	0.4178	0.4080	0.1881	0.4463
NEWT	nostem, stop	0.2837	0.1252	0.3445	0.2729	0.1226	0.3405
NEWT	stem, nostop	0.3700	0.1466	0.3897	0.3980	0.2000	0.4569
NEWT	stem, stop	0.4040	0.1736	0.4275	0.4340	0.2122	0.4715

Table 1: Effectiveness results for Indri and NEWT, *with preprocessing of text* at indexing time, measured by precision at 10 documents retrieved, mean average precision, and normalised discounted cumulative gain. Preprocessing variants are combinations of stemming and stopping.

System	TREC 7 P@10	TREC 7 MAP	TREC 7 nDCG	TREC 8 P@10	TREC 8 MAP	TREC 8 nDCG
Indri	0.3760	0.1570	0.3885	0.4280	0.1938	0.4398
NEWT *plain*	0.3740	0.1711	0.4178	0.4080	0.1881	0.4463
NEWT *prefix*	0.4040	0.1773	0.4222	0.4400	0.1974	0.4614∗
NEWT *suffix*	0.3920	0.1653	0.4083	0.4120	0.1954	0.4462
NEWT *space*	0.4300∗	0.1763∗∗	0.4227∗∗	0.4400	0.2093	0.4702∗

Table 2: Effectiveness results for *query processing* approaches with NEWT for the TREC 7 and TREC 8 newswire documents. No preprocessing of the collection was carried out at indexing time. NEWT query processing variants are *plain*, *space*, *prefix*, and *suffix*, as described in Section 3.2. ∗ and ∗∗ indicate statistical significance relative to the Indri run at the 0.05 and 0.01 levels respectively, based on a paired t-test.

compared to using the full queries. The relative performance between Indri and NEWT on these *keyword* queries is very similar, with no statistically significant differences between them.

Japanese GeoTime. We also investigate the impact of word segmentation on documents and queries in Japanese. For documents in the the inverted index (Indri), word segmentation was applied using the Japanese morphological analyzer ChaSen. For documents in the self-index (NEWT), no word segmentation was applied. The results are shown in Table 4, with performance measured by precision at 10 documents retrieved, mean average precision, and normalised discounted cumulative gain, for the the NTCIR 2011 GeoTime Japanese collections.

As expected, the best method for the inverted index approach is to segment both documents and queries, while for the self-indexing approach good performance is achieved with no segmentation being applied. The difference in performance between these two approaches is not statistically significant (paired t-test, $p > 0.1$).

Unsurprisingly, mixing segmentation for only one of either documents or queries leads to a drop in performance for both indexing methods, as the probability of mismatches is sharply increased.

4.3 Japanese Web Pages

We conducted experiments using the NTCIR 2011 INTENT collection. For documents in the the inverted index (Indri), word segmentation is applied using MeCab. For documents in the self-index (NEWT), no word segmentation is applied. Separate query sets were then tested with and without segmentation using MeCab. The results are shown in Table 5, with performance measured by P@10, MAP, and nDCG for the the NTCIR 2011 INTENT Japanese collections.

The best performance for the inverted index approach (Indri) is when both queries and documents are segmented; for the self-index approach (NEWT) performance is best when no segmentation is carried out at either stage. The difference between these two runs is not statistically significant for any of the measures, except for nDCG@1000 (paired t-test, $p < 0.01$).

As for the GeoTime Japanese results, mixing segmentation approaches between documents and queries harms both indexing approaches.

5 Conclusions

In this paper, we present results for our new experimental ranked self-index NEWT. We show that NEWT is capable of supporting bag-of-words queries

System	Query type	MAP	RR	nDCG@10	nDCG
Indri	full	0.1931	0.4922	0.2782	0.3635
NEWT *plain*	full	0.1519	0.2481	0.1687∗	0.2430∗∗
NEWT *prefix*	full	0.1591	0.3411	0.1937∗	0.2622∗
NEWT *suffix*	full	0.1666	0.4110	0.2043	0.2919
NEWT *space*	full	0.1847	0.4742	0.2625	0.3348
Indri	keyword	0.2846	0.6950	0.4159	0.5344
NEWT *plain*	keyword	0.2541	0.6448	0.3572	0.4817
NEWT *prefix*	keyword	0.2689	0.6568	0.3892	0.5029
NEWT *suffix*	keyword	0.2737	0.6953	0.4105	0.5175
NEWT *space*	keyword	0.2822	0.7131	0.4133	0.5235

Table 3: Effectiveness results for Indri and NEWT, with *no text preprocessing*, for the GeoTime'11 collection for English topics. NEWT query processing variants are *plain*, *space*, *prefix*, and *suffix*, as described in Section 3.2. ∗ and ∗∗ indicate statistical significance relative to the Indri run at the 0.05 and 0.01 levels respectively, based on a paired t-test.

System	Query type	Document	Query	MAP	RR	nDCG@10	nDCG
Indri	full	segmented	segmented	0.3229	0.6056	0.3731	0.5144
NEWT	full	no segmentation	segmented	0.0447	0.1256	0.0656	0.1093
Indri	keyword	segmented	no segmented	0.2568	0.5163	0.3181	0.4261
NEWT	keyword	no segmentation	no segmentation	0.3535	0.6869	0.3851	0.5633

Table 4: Effectiveness results for Indri and NEWT, for the Japanese GeoTime'11 collection.

with effectiveness similar to traditional inverted indexing methods, and present the first evaluation of top-k ranked retrieval in large document collections using self-indexes. We then explore potential solutions to an unexpected substring problem inherent to all character-based indexing algorithms.

This work is an important first step in constructing an entirely new class of indexing algorithms and search engines that are capable of delaying linguistic processing until query time. By minimizing processing at index time, multiple text formats and languages can be supported simultaneously. Our indexes implicitly retain the original substring ordering, so new ranking methods that exploit term proximity or linguistic syntax can also be devised in future work.

6 Acknowledgments

The first author was supported by the Australian Research Council.

References

J. E. Burnett, D. Cooper, M. F. Lynch, P. Willett, and M. Wycherley. Document retrieval experiments using indexing vocabularies of varying size. I. Variety generation symbols assigned to the fronts of index terms. *Journal of Documentation*, 35(3):197–206, September 1979.

S. Büttcher, C. L. A. Clarke, and G. V. Cormack. *Information Retrieval: Implementing and evaluating search engines.* MIT Press, Cambridge, Massachusetts, 2010.

W. B. Croft, D. Metzler, and T. Strohman. *Search Engines: Information Retrieval in practice.* Addison Wesley, Boston, 2010.

J. S. Culpepper, G. Navarro, S. J. Puglisi, and A. Turpin. Top-k ranked document search in general text databases. In *ESA, Part II*, volume 6347 of *LNCS*, pages 194–205. Springer, 2010.

R. Dale, H. Moisl, and H. Somers. *Handbook of Natural Language Processing.* CRC Press, New York, NY, 2000.

T. de Heer. The application of the concept of homeosemy to natural language information retrieval. *Information Processing & Management*, 18(5):229–236, 1982.

P. Ferragina and G. Manzini. Opportunistic data structures with applications. In *FOCS*, pages 390–398. IEEE, November 2000.

W. B. Frakes. Term conflation for information retrieval. In *SIGIR*, pages 383–389, August 1984.

F. Gey, R. Larson, J. Machado, and M. Yoshioka. NTCIR9-GeoTime overview - evaluating geographic and temporal search: Round 2. In *Proceedings of NTCIR-9 Workshop Meeting (in printing)*, December 2011.

R. Grossi, A. Gupta, and J. S. Vitter. Higher-order entropy-compressed text indexes. In *SODA*, pages 841–850, January 2003.

System	Document	Query	MAP	RR	nDCG@10	nDCG
Indri	segmented	segmented	0.3413	0.9255	0.3553	0.5298
Indri	segmented	no segmentation	0.1550	0.3924	0.1509	0.2169
NEWT	no segmentation	segmented	0.3044	0.8251	0.2629	0.4433
NEWT	no segmentation	no segmentation	0.3651	0.9675	0.3182	0.4638

Table 5: Effectiveness results for Indri and NEWT, for the Japanese NTCIR 2011 INTENT collection.

D. Harman. How effective is suffixing? *Journal of the American Society for Information Science*, 42(1): 7–15, 1991.

V. Hollink, J. Kamps, C. Monz, and M. de Rijke. Monolingual document retieval for European languages. *Information Retrieval*, 7(1-2):33–52, 2004.

W.-K. Hon, R. Shah, and J. S. Vitter. Space-efficient framework for top-k string retrieval problems. In *FOCS*, pages 713–722. IEEE, October 2009.

W.-K. Hon, R. Shah, and J. S. Vitter. Compression, indexing, and retrieval for massive string data. In *CPM*, volume 6129 of *LNCS*, pages 260–274. Springer, June 2010.

D. A. Hull. Stemming algorithms: A case study for detailed evaluation. *Journal of the American Society for Information Science*, 47(1):70–84, 1996.

R. Krovetz. Vierwing morphology as an inference process. In *SIGIR*, pages 191–202, June 1993.

K. L. Kwok. Comparing representations in Chinese information retrieval. In *SIGIR 1997*, pages 34–41. ACM Press, August 1997.

J. B. Lovins. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11(1–2):22–31, 1968.

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.

C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK, 2008.

P. McNamee and J. Mayfield. Character n-gram tokenization for European language text retrieval. *Information Retrieval*, 7(1–2):73–97, 2004.

P. McNamee, C. Nicholas, and J. Mayfield. Don't have a stemmer? Be un+concern+ed. In *SIGIR*, pages 813–814. ACM Press, July 2008.

S. Mithukrishnan. Efficient algorithms for document retrieval problems. In *SODA*, pages 657–666, January 2002.

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

J.-Y. Nie, K. Gao, J. Zhang, and M. Zhou. On the use of words and n-grams for Chinese information retrieval. In *IRAL 2000*, pages 141–148. ACM Press, 2000.

C. D. Paice. Another stemmer. *ACM SIGIR Forum*, 24(3):56–61, 1990.

M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, July 1980.

S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In *TREC-3*, 1994.

R. Song, M. Zhang, T. Sakai, M. Kato, Y. Liu, M. Sugimoto, Q. Wang, and N. Orii. Overview of the NTCIR-9 INTENT tasks. In *Proceedings of NTCIR-9 Workshop Meeting (in printing)*, December 2011.

N. Välimäki and V. Mäkinen. Space-efficient algorithms for document retrieval. In *CPM*, volume 4580 of *LNCS*, pages 205–215. Springer, July 2007.

Ellen M. Voorhees and Donna K. Harman. Overview of the Eighth Text REtrieval Conference (TREC-8). In *TREC-8*, pages 1–24, 1999.

P. Willett. Document retrieval experiments using indexing vocabularies of varying size. II. Hashing, truncation, digram and trigram encoding of index terms. *Journal of Documentation*, 35(4):296–305, December 1979.

I. H. Witten, A. Moffat, and T. C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann, San Francisco, second edition, 1999.

M. Yasukawa and H. Yokoo. Composition and decomposition of Japanese katakana and kanji morphemes for decision rule induction from patent documents. In *ADCS*, pages 28–35, December 2010.

J. Zobel and A. Moffat. Inverted files for text search engines. *ACM Computing Surveys*, 38(2):6–1 – 6–56, 2006.