World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:2, No:6, 2008

Mining News Sites to Create Special Domain News Collections

David B. Bracewell, Fuji Ren, Shingo Kuroiwa

***Abstract*—**We present a method to create special domain collections from news sites. The method only requires a single sample article as a seed. No prior corpus statistics are needed and the method is applicable to multiple languages. We examine various similarity measures and the creation of document collections for English and Japanese. The main contributions are as follows. First, the algorithm can build special domain collections from as little as one sample document. Second, unlike other algorithms it does not require a second "general" corpus to compute statistics. Third, in our testing the algorithm outperformed others in creating collections made up of highly relevant articles.

***Keywords*—**Information Retrieval, News, Special Domain Collections,

I. Introduction

THIS document collections are important for Information Retrieval (IR), Knowledge Engineering and Natural Language Processing (NLP). For IR systems, often times a document collection is the information source for the system. One particular type of collection that is extremely important is special domain collections. Special domain collections contain documents that are specific to a given topic or theme. To be of use, these collections need to be relatively large and contain highly relevant domain specific documents. A small document collection or a collection full of erroneous documents will degrade the performance of any algorithm or system that uses them.

Typically, special domain collections are manually created by combining documents from various sources. However, while this method ensures the collection contains highly relevant documents, it is extremely time consuming. Because of this, creating large collections is a burden. Moreover, since a collection of relevant domain specific documents can provide statistical information, domain specific terms, etc. for the respective domain, it is sometimes desirable to build a collection for the sole purpose of mining this information for later use. Manually building the collection for such a short-lived task is too cumbersome and the cost outweighs the benefit of the acquired information. As such, semi-automatic or automatic methods need to be created for constructing the document collections.

Recently, due to the explosion of information available, a great deal of research has been done on utilizing the web for varying IR tasks. Specifically, the web is often treated as a large source of information for Question & Answering Systems [1] and [2] mining for bilingual corpora [3], using web-based statistics for NLP [4], etc. With the wealth of knowledge available and the success of researchers on other tasks, it appears that the web is a viable source for building domain specific document collections.

The major problem with using the Internet, however, is that the credibility of the source and the quality of writing varies from site to site. One source of information that can be generally considered credible and have a high writing quality is news sites. News articles are a good source of information and the information within can be considered trustworthy. In addition, news covers many topics and domains of information.

This paper presents an algorithm for semi-automatically creating special domain collections from news articles. Given at least one sample article, it is capable of creating moderate size collections that contain articles highly related to each other and relevant to the sample. It makes use of a keyword extraction algorithm that can extract keywords from a single document without a document collection or corpus statistics. Using the extracted keywords, a directed search can be performed over various news sites to find relevant articles. A similarity measure is then used to determine which of the articles are in the domain and which are not.

This paper will continue as follows, in section II related work is discussed. In section III an overview of the proposed method is given. In section IV, the keyword extraction algorithm is examined. Next, in section V, the article gathering module is examined. Then, in section VI, computing the similarity between two articles is discussed. In section VII, the results of the experiments on article gathering and article similarity are shown. Finally, in section VIII, future work is discussed and concluding remarks made.

Manuscript received January 18, 2007. (Write the date on which you submitted your paper for review.) This research has been partially supported by the Ministry of Education, Culture, Sports, Science and Technology of Japan under Grant-in-Aid for Scientific Research (B), 14380166, 17300065, Exploratory Research, 17656128.

David B. Bracewell is with the University of Tokushima, Tokushima, Japan (e-mail: davidb@is.tokushima-u.ac.jp).

Fuji ren is with the University of Tokushima, Tokushima, Japan and Beijing University of Posts and Telecommunications, Beijing, China (e-mail: ren@is.tokushima-u.ac.jp).

Shingo Kuroiwa is with the University of Tokushima, Tokushima, Japan (e-mail: kuroiwa@is.tokushima-u.ac.jp).

II. Related Work

The web is already seen as a good source of information for creating corpora, as seen by the birth of workshops such as ACL's "Web as Corpus." Researchers are also more and more seeing the Web as a way of building ad hoc corpora, such as [5], [6], and [3]. The use of the web for creating corpora is not limited to just computer science research. Research such as [6] and [7] have produced systems for creating bilingual lexicons that can aid humans in the translation of documents. However, there has been little work done to date on semi-automatically or automatically creating special domain collections (corpora) from the Web. In the following paragraphs we will look at some research that has focused on special domain collections or could be used for special domain collections.

One approach that has been created and greatly used over the years is focused or restricted crawling [8]. These approaches try to crawl only sites that contain information about a certain topic. They require an initial set of seed pages in which the focused crawler can start crawling from. These methods rely on pages having links to other pages that are of a similar topic. The main disadvantage is in the need for many initial seed pages where as the algorithm presented in this research requires only one sample article. Moreover, we believe that news articles present special problems for these types of approaches. The main reason is that many of the links that are on a page with a news article have no relation to the article or the topic of the article. In an informal investigation performed by us on various English language news sites, we found that over 70% of the links on each page had no relation to the article.

One way to create special domain corpora would be to treat the problem as a search problem. In this case the standard vector space model using TFIDF and cosine similarity with a threshold could be used to find documents similar to a user given sample. With the addition of more advanced algorithms like the pseudo-relevance method presented by [9] it would have the possibility to create good special domain corpora. However, there are some disadvantages with these type of methods. The first is that they would require a second corpus in which to calculate IDF values. The second is that when using a second corpus as a surrogate corpus for IDF calculation the performance of TFIDF is decreased. In addition the size and domain of the surrogate corpus will impact the performance as well.

To show some of this performance decrease we present an example from some previous testing we have done. Table 1 shows the performance of surrogate corpora on keyword search for a 1,000 article special domain collection on sports. Documents had keywords extracted using TFIDF and then the collection was searched using the top 10 keywords as a query for each document. The table shows the percentage of time it returned the correct document. The results when using the sports collection as the IDF corpus are 49.9%. The results show poor performance for all surrogate corpora, except Google, and in some cases TF only was better than using a surrogate.

TABLE I
TFIDF with a Surrogate Corpus

Measure	Search Accuracy
Google	52.8%
Wired (5,000 documents)	47.7%
NONE	45.7%
Yahoo (1,000 documents)	44.4%
Reuters (1,000 documents)	42.5%

Fairon introduced the Corporator system which created special domain corpora by mining RSS feeds [10]. The benefit is that the system retrieves a set of articles that have already been deemed related. The disadvantage, though, is that even with the growth of RSS it could still be problematic to find feeds that are "compatible" with the user desired topic.

Baroni and Bernardini introduced BootCat, which is a set of tools to build ad-hoc corpora and term lists [6]. It is capable of creating single and multiple-word lists as well as corpora. Through their experimentation they found the system was able to create word lists that could aid humans in translation. The main problem is that the system works on unrestricted text. The credibility and quality of the text is unknown when dealing with unrestricted text. In addition it requires a second "general" corpus to be used for mining words.

Google News (http://news.google.com) has an option on some of its articles to list similar news articles. However, the underlying algorithm is not open. Also, this option does not seem to be available for all articles.

III. Overview of Proposed Algorithm

The proposed algorithm makes use of the searching capabilities of news sites and news aggregators, like Google News. Queries are created by using keywords that are extracted from sample articles. The results of the search engine are then examined to determine which of the articles are truly relevant.

An overview can be seen in figure 1. The algorithm is broken down into three main modules: keyword extraction, article gathering and article similarity. Keyword extraction finds the important words and phrases in the article that can describe the article and its content. The keywords are then used for searching news sites and news aggregator sites. The articles from the resulting HTML pages are extracted and their similarity to the current sample article is calculated. Those articles with a high enough similarity are added as relevant articles and are also added to the article queue so that they can become a new sample article.

The following sections will look at each of the three main parts in detail. There are few contributions that the proposed algorithm gives over previous methods. First, the only data that is needed to create a collection is at least one sample article. If a broader topic is desired then multiple sample articles can be given. Second, the credibility and quality of writing of the created collection should be high. Finally, it is easily expanded into other languages.

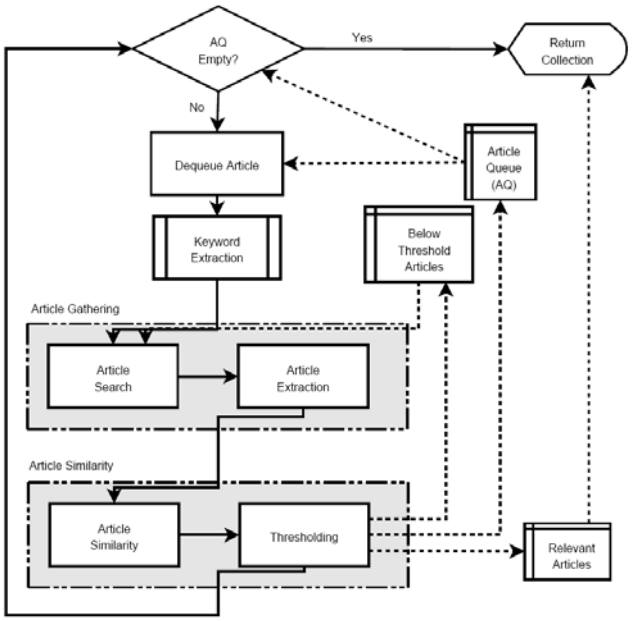

Fig. 1 Overview of proposed method

IV. Keyword Extraction

The keyword extraction algorithm used in the algorithm was proposed by Bracewell et al. [11]. It is able to extract keywords from a single document without requiring a document collection or corpus statistics. The keywords are restricted to being noun phrases, because noun phrases carry the most information in describing the article. In this section we will briefly describe the algorithm and modifications that we made to make the algorithm better suited for computing article-article similarity.

A. Overview

The algorithm uses linguistic information in the form of noun phrases to extract keywords. Phrases can make better keywords than single words as they can keep compound words together. For example "White House" would be a better keyword than having "White" and "House" separately. The original algorithm was broken down into three modules listed below. However, we do not make use of NP Clustering and Scoring as we found it to be detrimental in calculating article-article similarity.

1. Morphological Analysis
2. Noun Phrase (NP) Extraction and Scoring
3. Noun Phrase (NP) Clustering and Scoring

The Morphological Analysis module takes care of word segmentation, word stemming and part-of-speech tagging. The NP Extraction & Scoring module uses a simple NP grammar to extract noun phrases and then scores them based on their frequency and the frequencies of the words in the NP.

Bracewell et al. defined the following advantages of this algorithm over others. First, it works on a single document and does not require a collection or corpora to compute statistics from or to use as training data. Second, it out performs various other methods in both human judged and task related evaluation in multiple languages.

B. Morphological Analysis

Morphological analysis is the identification of word stems and, optionally, syntactic categories (Parts-of-Speech). Most languages used in research have these tools available. For English, Porter's stemmer [12] and Brill's tagger [13] were used. For Japanese, Chasen [14] was used.

C. Noun Phrase Extraction and Scoring

The algorithm extracts noun phrases using a very simple NP grammar which justs looks for adjectives and nouns. After the NPs are extracted any stop words appearing in them are removed. The noun phrases are then given a score to determine their weight within in the article.

The *NPScore* is based on the frequency of the noun phrase in the article and the words making up the noun phrase in the article, see equation 1. In the equation, NP = {w1,w2, … ,wn}, i.e. an NP is a set of words, f_{NP} is the frequency of the NP within in the document, and tf_{wi} is the term frequency of word i.

$$NPScore(NP) = \log\left(n + \frac{\sum_{i=1}^{n} tf_{wi}}{n} + f_{NP} \right) \quad (1)$$

V. Article Gathering

The Article Gathering module searches for possibly relevant news articles and extracts them from their HTML pages. The module is made up of two parts: article search and article extraction. Each of these will be described in more detail below.

A. Article Search

Google News and a list of 11 sites (6 English and 5 Japanese), listed below, were used for searching. For each sample article five queries are created. Each query is made up of three keywords randomly chosen from the top 15 scoring keywords for the sample article. Using these queries, the top 10 results from each site are downloaded.

- Yahoo! News English
- CNN English
- BBC News English
- International Herald Tribune English
- Mainichi Shimbun English
- Yomiuri Shimbun English
- Yahoo! News Japanese
- Mainichi Shimbun Japanese
- Yomiuri Shimbun Japanese
- Livedoor News Japanese
- Asahi Shimbun Japanese

After the first sample article is processed, subsequent article searches also include the news articles that were previously

deemed non-relevant. This is done for two reasons. The first is to lower the possibility of missing relevant articles. The second is to act like a cache and help to avoid downloading an article more than once.

Currently, a simple method is employed to not download a relevant article more than once. This method simply checks the document's URL and the article's title in the list of already downloaded articles. If either one has been downloaded, then it will not be downloaded again.

B. Article Extraction

Rule-based article extraction is performed on the downloaded articles. Certain sites have hand crafted rules in the form of regular expressions assigned that allow for very precise extraction. However, since this poses a possible bottleneck for the addition of new news sites, there is also a default rule set that was created. After examining many news sites we found a certain pattern that described the prototypical article. The pattern is shown in figure 2 and is made up of a title then the article body and finally a footer.

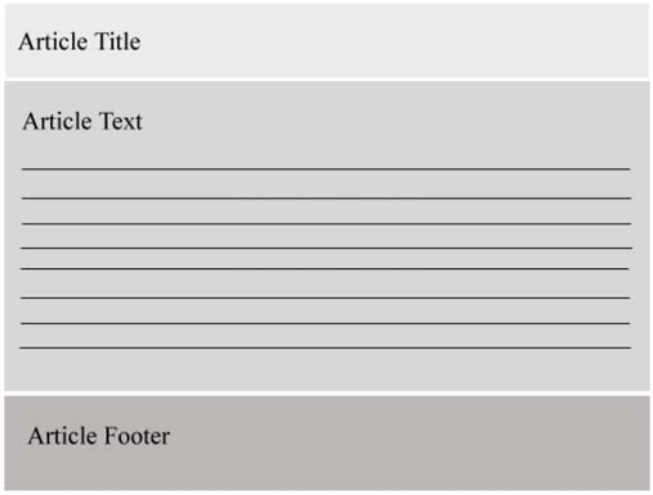

Fig. 2 prototypical news article layout

The extraction process is done in three parts: title extraction, text extraction and text cleansing. An overview can be seen in figure 3. Title extraction starts by assigning the web page's title element as the article's title. The title is then searched for line by line in the HTML body. If the title is found then the process moves on to find the footer. If the title is not found then starting at the first line in the HTML body, each line is examined by using it as a regular expression against the web page's title element. If the regular expression matches then that line is assigned as the article's title.

The next step is to find the article footer. This starts by examining each line that comes after the article's title. Each line is checked to see if it contains one of the defined footer elements. Some of the currently used footer patterns are Copyright, ©, (C), Email this, Related Stories, etc. If no footer pattern is found then the bottom of the HTML body is used.

After the footer is found the article can be extracted. The article is taken as the text that falls between the title and the footer. The article is then cleansed. The cleansing process removes multiple white space and non-important multiple punctuation marks. In informal testing, the rule based extraction method only failed to extract three articles out of 5,000.

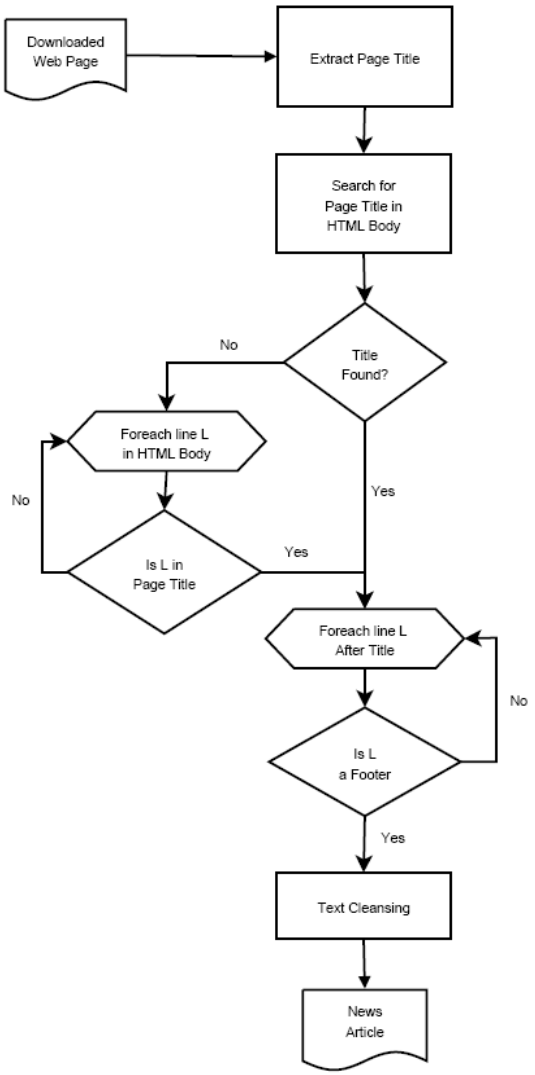

Fig. 3 overview of rule based extraction

VI. Article Similarity

This section presents how to take the extracted articles and determine if they are relevant or not to the special domain. First, how to calculate the similarity between two articles is discussed. Finally, article thresholding is examined.

A. Article Similarity

The Article Similarity module takes the extracted articles and determines if they are similar to the sample article or not. Article similarity is computed using a keyword based similarity measure. The similarity measure used can be any

that uses keywords. Seven different similarity measures were experimented with as well as varying thresholds. Here, the seven measures will be explained and in the experimentation section results on using the different measures with varying thresholds will be shown.

Because the keyword extraction algorithm that is being used is new and has not been previously used for article-article similarity measures, a variety of similarity coefficients and measures were tested. In all seven different measures were tested and are listed below. The first five measures are standard in the information retrieval community, but the last two were created for the current research.

- Dice's Coefficient (Dice) [15]
- Jaccard's Coefficient (Jaccard) [15]
- Cosine Coefficient (Cosine) [15]
- Overlap Coefficient (Overlap) [15]
- Cosine Similarity (VectorCosine) [16]
- Percentage Score Similarity (PS)
- Partial Cosine Similarity (PartialCosine)

The first four measures are based on the number of keywords shared and do not use any scoring information, see equations 2, 3, 4, and 5 (K_i is used to represent the set of keywords for an article and $\vec{K_i}$ a keyword vector for the article). The Percentage Score, equation 6, and Cosine Similarity, equation 7, use the score returned from the keyword extraction algorithm.

$$Dice(a_q, a_i) = \frac{2 * \left|K_q \cap K_i\right|}{\left|K_q\right| + \left|K_i\right|} \tag{2}$$

$$Jaccard(a_q, a_i) = \frac{\left|K_q \cap K_i\right|}{\left|K_q \cup K_i\right|} \tag{3}$$

$$Cosine(a_q, a_i) = \frac{\left|K_q \cap K_i\right|}{\sqrt{\left|K_q\right|} \times \sqrt{\left|K_i\right|}} \tag{4}$$

$$Overlap(a_q, a_i) = \frac{\left|K_q \cap K_i\right|}{\min\left(\left|K_q\right|, \left|K_i\right|\right)} \tag{5}$$

$$PS(a_q, a_i) = \frac{\vec{K}_q \bullet \vec{K}_i}{\sum_{j=1}^{|K_q|} Score(k_{qj}) + \sum_{j=1}^{|K_i|} Score(k_{ij})} \tag{6}$$

$$VectorCosine(a_q, a_i) = \frac{\vec{K}_q \bullet \vec{K}_i}{\left|\vec{K}_q\right| + \left|\vec{K}_i\right|} \tag{7}$$

The partial cosine measure is an extended version of the cosine similarity. Because the keywords are noun phrases, it is less likely for two articles to contain the same keyword. Because of this, the partial cosine measure includes a percentage of the keyword scores from two keywords that partially match, i.e. share an n-gram in common. For example, "natural language" and "natural language processing" partially match and a percentage of their score would be added. The partial matching takes place after the initial dot product for the cosine similarity has been calculated and excludes keywords that have been fully matched. Figure 4 shows pseudo code for calculating the partial cosine. The partial match can be determined by any calculation, but currently dice's coefficient is used.

$dProduct \leftarrow \vec{K_q} \cdot \vec{K_i}$
$K'_q \leftarrow \forall k_j \in K_q - (K_q \cap K_i)$
$K'_i \leftarrow \forall k_j \in K_i - (K_q \cap K_i)$
foreach k_j in K'_q
{
    $maxPartialMatch \leftarrow$ maximum Partial Match for k_j
    $k_m \leftarrow k \in K_i$ with maximum Partial Match
    $dProduct += maxPartialMatch * k_j * k_m$
    Remove k_m from K_i
}
$PartialCosine \leftarrow \frac{dProduct}{|\vec{K_q}| \times |\vec{K_i}|}$

Fig. 4 partial cosine algorithm

B. Article Thresholding

After the similarity measure is calculated it is compared to a threshold. If the score is over the threshold then it is added to the article collection. The threshold depends on the similarity measure used and can determine the size and noisiness of the created collection.

VII. Experimentation

This section covers three experiments that were performed. The first was a test on various similarity measures and thresholds in an attempt to determine the best combination for the algorithm. The second experiment looks at the quality of the collections that are created. The final experiment estimates the possible size of collections created using this method.

A. Similarity Measure

There are many similarity measures and similarity coefficients. Seven algorithms were compared to find which one works the best with the keyword algorithm and with English and Japanese.

For each language eight special domain collections were manually created. Each collection contained 51 articles for a total article collection of 408 articles per language. One article from each of the special domain collections was randomly chosen and extracted from the total article collection to be the sample (query) article for that set. Each of the similarity measures where then computed between each of the sample articles and the entire article set. The thresholds were then used to determine which articles were deemed relevant to the sample article and which were not.

Precision is more important than recall in creating a special domain collection, because the erroneous articles will hurt the collection more than missing articles. Because of this, the F0.25-measure, shown in equation 8, was used to examine the results of the similarity measures and the thresholds. The F0.25-measure is a variation of the standard F-measure that weights precision four times more than recall. Figure 5 shows results for English and figure 6 for Japanese. The graphs plot the f0.25-measure versus the threshold.

$$F_{0.25} - Measure = \frac{1.25 * Precision * Recall}{(0.25 * Precision) + Recall} \quad (8)$$

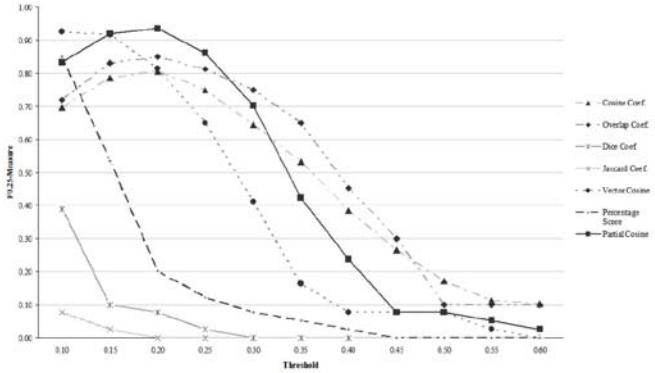

Fig. 5 similarity results for English

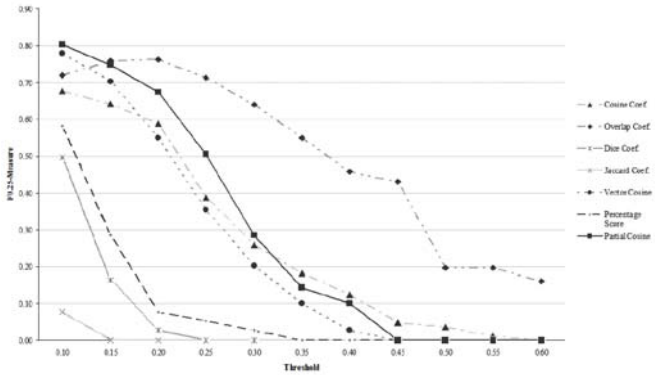

Fig. 6 similarity results for Japanese

The goal of this experiment was to determine if a single similarity measure and threshold could be chosen that can work well form both languages. Because the quality of the collection is heavily dependent on the similarity measure this testing was needed. For English, the partial cosine similarity with a threshold of 0.20 had the highest F0.25-measure of 0.94. It had a recall of 0.61 and a precision of 0.99. For Japanese, the partial cosine similarity measure also had the highest F0.25-measure of 0.8, but at a threshold of 0.1. It had a recall of 0.33 and precision of 0.94.

In order to determine if these measures and thresholds should be kept we examined the combinations that had 100% precision. Table 2 shows the threshold and highest recall value for each measure that achieved a precision of 100%. As can be seen in the table to achieve the extra 1% of precision for English will cost 43% of recall and for an extra 6% of precision for Japanese will cost 22% recall. Because of the great variation in recall, we chose to stick with the F0.25-measure results and choose the partial cosine similarity measure to be used in the algorithm.

TABLE II
COMBINATIONS WITH 100% PRECISION

English		
Measure	Threshold	Recall
Partial Cosine	0.30	18%
Overlap	0.35	15%
Vector Cosine	0.25	15%
Percentage Score	0.15	10%
Cosine	0.45	7%
Dice	0.10	6%
Jaccard	0.10	1%
Japanese		
Measure	Threshold	Recall
Cosine	0.25	11%
Percentage Score	0.10	11%
Partial Cosine	0.25	9%
Dice	0.10	8%
Vector Cosine	0.25	5%
Jaccard	0.10	1%

While we were able to find a single similarity measure to work on both languages, we were not able to come to a consensus on the threshold. Therefore, we have left both the measure and threshold to be tunable. This will allow collections to be built that have noise or for small collections that have no error to be built. With this said, the partial cosine similarity measure with a threshold of 0.20 for English and 0.10 for Japanese were used for the rest of the evaluations.

B. Text Classification Evaluation

To test the quality of the created collections we used text classification. The text classification algorithm presented by Bracewell et al. [17] was used. It requires only positive examples for training data and achieves high precision and recall.

For comparison purposes, a baseline collection method and TFIDF based collection method were also used. The baseline method simply returns the top 20 results from each of the news site's search engines. The query words were manually created. The TFIDF based algorithm used the same algorithm presented here, but changed the keyword extraction algorithm to TFIDF keyword extraction. To calculate the IDF values, a 10,000 document corpus containing news articles across all categories and topics was created. For computing the similarity, the standard cosine similarity measure was used with a threshold of 0.3.

Each algorithm was used to generate eight English language collections covering Iran, Iraq, North Korea, Israel, GM-Nissan-Renault talks, home prices, HP scandal, and inflation. The size of each collection was 100 documents. The size was limited because of the difficulties we had in creating large collections with the TFIDF based algorithm. From each of the collections 50 articles were chosen as training data and the

TABLE III
CLASSIFICATION RESULTS

Baseline			
	Recall	Precision	F-Measure
Micro	59.3%	60.6%	59.9%
Macro	59.3%	67.7%	58.3%
TFIDF			
	Recall	Precision	F-Measure
Micro	50.8%	52.1%	51.4%
Macro	50.8%	55.4%	47.5%
Proposed			
	Recall	Precision	F-Measure
Micro	90.5%	91.0%	90.7%
Macro	90.5%	92.3%	90.9%

rest of the articles as testing data. Table 3 shows the micro and macro averaged recall, precision and F-measure for classification.

As can be seen from the table the proposed algorithm greatly outperformed the TFIDF and baseline algorithms. Despite the fact the threshold was set high, which caused creating even a small 100 article collection difficult, the TFIDF based algorithm had very poor results. On average it required more than 15 documents to create a 100 article collection. The simple user initiated search created better results than the TFIDF based algorithm. This could be possibly due to better keywords chosen for the query. The proposed algorithm, however, had very good results and only required one document to easily create a 100 document collection.

C. Article Collection Size

Judging the possible collection size is difficult. Depending on the number of sample articles and the time allotted to build the collection the collection size can vary greatly. The average articles per hour is also heavily dependent on the speed of the Internet connection, the article content and the similarity measure used. Articles that contain more current and well known events will have more articles available and should, therefore, have a larger articles per hour. The partial cosine similarity measure requires costly calculations that make it slower than a simpler measure like dice or even the standard cosine similarity.

As such some simple decisions were made to gauge the algorithm. A single sample article was given to the system and the system was allowed to run for thirty minutes. Each language had five different runs using five different sample articles. Table 4 shows the average articles per hour for English and Japanese.

With the rate achieved in testing, we can estimate that for English a 7,440 article collection could be created in one day and for Japanese a 5,328 article collection could be created in one day. However, these are only estimates and the actual articles per hour and collection size will vary depending on the sample article. Additionally, the system is currently unoptimized and only uses a single thread. A more optimized multi-threaded version should help to increase the articles per hour greatly.

TABLE IV
COLLECTION SIZE RESULTS

Language	Articles Per Hour
English	310
Japanese	222

VIII. CONCLUSION AND FUTURE WORK

An algorithm for creating special domain collections by mining news sites was presented. The main contributions of the algorithm are as follows. First, the algorithm can build special domain collections from as little as one sample document. Second, unlike other algorithms it does not require a second "general" corpus to compute statistics. Third, in our testing the algorithm outperformed others in creating collections made up of highly relevant articles.

Using a keyword extraction algorithm that only requires a single document and is applicable to multiple languages, the proposed method extracts keywords from a user supplied sample article and searches news sites for possibly relevant articles. The documents are then downloaded and their articles are extracted. Each of these articles has keywords extracted and compared to the sample article for similarity. Those articles that are found similar are kept and added to the collection.

The algorithm was tested on both English and Japanese to see its ability to work across multiple languages. Through testing we found that there is no one similarity measure and threshold that can work equally well on both English and Japanese. As such, both of these are left as tunable options so that a user can determine the size and amount of noise in the collection.

To test the quality of the created collections a text classifier that requires only positive examples was used. The proposed algorithm was compared to a baseline and a TFIDF based algorithm. Each algorithm created 8 small collections for the English language. Half of the articles in each collection were used to train the classifier and the other half to test the classifier. The classification results from training and testing with the proposed algorithm were greatly higher than that of the baseline and TFIDF algorithms.

In the future, we want to use the algorithm to automatically create training data to be used in category classification. We also want to use the ad-hoc corpora to build word lists, create bilingual lexicons and be the source for answers in a Question & Answering system. We also hope to extend the algorithm to create comparable corpora.

REFERENCES

[1] Dragomir Radev, Weiguo Fan, Hong Qi, Harris Wu, and Amardeep Grewal, "Probabilistic question answering on the web", in WWW '02: Proceedings of the 11th international conference on World Wide Web, New York, NY, USA, 2002, pp. 408–419, ACM Press.

[2] Dmitri Roussinov and Jose Robles, "Learning patterns to answer open domain questions on the web", in SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, 2004, pp. 500–501, ACM Press.
[3] P. Resnik and N. A. Smith, "The web as a parallel corpus", Computational Linguistics, vol. 29, pp. 349–380, 2003.
[4] Mirella Lapata and Frank Keller, "Web-based models for natural language processing", ACM Trans. Speech Lang. Process., vol. 2, no. 1, pp. 1–31, 2005.
[5] William H. Fletcher, "Facilitating the compilation and dissemination of ad-hoc web corpora", in Papers from the Fifth International Conference on Teaching and Language Corpora, 2004.
[6] M. Baroni and S. Bernardini, "Bootcat: Bootstrapping corpora and terms from the web", in Proceedings of LREC 2004, 2004.
[7] Sara Castagnoli, Using the Web as a Source of LSP Corpora in the Terminology Classroom, chapter 6, pp. 159–172, GEDIT, 2006.
[8] Soumen Chakrabarti, Martin van den Berg, and Byron Dom, "Focused crawling: a new approach to topic-specific Web resource discovery", Computer Networks (Amsterdam, Netherlands: 1999), vol. 31, no. 11–16, pp. 1623–1640, 1999.
[9] G. Salton and C. Buckley, "Improving retrieval performance by relevance feedback", Journal of the American Society for Information Science, vol. 41, pp. 288–297, 1990.
[10] Cdrick Fairon, "Corporator: A tool for creating rss-based specialized corpora", in Proceedings of the 2nd International Workshop on Web as Corpus, 2006.
[11] David B. Bracewell, Fuji Ren, and Shingo Kuroiwa, "Multilingual single document keyword extraction for information retrieval", in Proceedings of the 2005 IEEE International Conference on Natural Language Processing and Knowledge Engineering, Wuhan, China, November 2005.
[12] M.F. Porter, "An algorithm for suffix stripping", Program, vol. 14, pp. 130–137, 1980.
[13] E. Brill, "A simple rule-based part-of-speech tagger", in Proceedings of 3rd Applied Natural Language Processing, 1992, pp. 152–155.
[14] Yuji Matsumoto, Akira Kitauchi, Tatsuo Yamashita, Yoshitaka Hirano, Hiroshi Matsuda, Kazuma Takaoka, and Masayuki Asahara, "Morphological analysis system chasen version 2.2.9 manual.", Tech. Rep., Nara Institute of Science and Technology, 2002.
[15] R. C. J. van Rijsbergen, Information Retrieval: Second Edition, Butterworth-Heinemann, 1979.
[16] Gerald Salton, Automatic Text Processing, Addison-Wesley Publishing Company, 1998.
[17] David B. Bracewell, Fuji Ren, and Shingo Kuroiwa, "Category classification and topic discovery of news articles", in Proceedings of Information-MFCSIT 2006, 2006, pp. 345–348.

David B. Bracewell received the BS and MS degree in Computer Science from the University of Central Florida, Orlando, in 2002 and 2004 respectively, and is currently working toward the Ph.D. degree at The University of Tokushima, Japan. His current research interests include Information Retrieval, Natural Language Processing, Evolutionary Computation and Affective Computing. He has worked on research funded by NASA and is currently working on a Japanese-English crosslingual information retrieval system for news articles. He is a member of IEEE, IPSJ and the International Association of Engineers. He is on the review board for the International Journal of Computational Intelligence Research and has been a reviewer and session chair for several international conferences.

Fuji Ren received the Ph. D. degree in 1991 from Faculty of Engineering, Hokkaido University, Sapporo, Japan. He worked at CSK, Japan, where he was a chief researcher of NLP. From 1994 to 2000, he was an Associate Professor in the Faculty of Information Sciences, Hiroshima City University. From 2001 he joined the Faculty of Engineering, the University of Tokushima, Japan, as a Professor. He was a visiting professor at CRL, New Mexico State University, USA, from 1996-1997. His current research interests include Natural Language Processing, Machine Translation, Artificial Intelligence, Language Understanding and Communication, Multi-Lingual Multi-Function Multi-Media Intelligent System, Super-Function Methodology, Sensitive Information Processing and Affective Computing. He is a senior member of IEEE, a member of NLP, AAMT, IPSJ, Area Editor-in-Chief of INFORMATION, Associate Editor of Asian Information-Science-Life, Editor of IJITDM.

Shingo Kuroiwa received the B.E., M.E. and D.E. degrees in electro-communications from The University of Electro Communications, Tokyo, Japan, in 1986, 1988, and 2000, respectively. From 1988 to 2001 he had been a resercher at the KDD R & D Laboratories. Since 2001, he has been with the Faculty of Engineering, Tokushima University, Tokushima, Japan, where he is currently an Associate Professor. His current research interests include speech recognition, speaker recognition, natural language processing, and information retrieval. He is a member of the Information Processing Society, the Institute of Electronics, Information and Communication Engineers, the Japanese Society for Artificial Intelligence, and the Acoustical Society of Japan.