Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 225 (2009) 51–65

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Category Classification and Topic Discovery of Japanese and English News Articles

David B. Bracewell[1,2]

Department of Information Science and Intelligent Systems
The University of Tokushima
Tokushima, Japan

Jiajun Yan[3]

Department of Information Science and Intelligent Systems
The University of Tokushima
Tokushima, Japan

Fuji Ren[4]

Department of Information Science and Intelligent Systems
The University of Tokushima
Tokushima, Japan
School of Information Engineering
Beijing University of Posts and Telecommunications
Beijing 100876, China

Shingo Kuroiwa[5]

Department of Information Science and Intelligent Systems
The University of Tokushima
Tokushima, Japan

Abstract

This paper presents algorithms for topic analysis of news articles. Topic analysis entails category classification and topic discovery and classification. Dealing with news has special requirements that standard classification approaches typically cannot handle. The algorithms proposed in this paper are able to do online training for both category and topic classification as well as discover new topics as they arise. Both algorithms are based on a keyword extraction algorithm that is applicable to any language that has basic morphological analysis tools. As such, both the category classification and topic discovery and classification algorithms can be easily used by multiple languages. Through experimentation the algorithms are shown to have high precision and recall in tests on English and Japanese.

Keywords: Category Classification, Topic Discovery, Topic Classification, Information Retrieval, News Domain

doi:10.1016/j.entcs.2008.12.066

1 Introduction

We define topic analysis for news as identifying not only the topic, but also the category of a news article. For news, categories are high level groupings that allow for easier navigation of articles. Newspapers and Internet news sites are broken down by category. For example, a newspaper will have a sports page, business page, etc. We define topics to be the main themes of news articles. Topics are also a part of newspapers and Internet news sites.

The combination of topics and categories create a hierarchical structure that allows for drill down navigation. Figure 1 gives an example of such a hierarchy taken from Yahoo! News(http://news.yahoo.com) in December of 2005. For example, an article about the “World Baseball Classic” can belong to a topic on “baseball” and to the category “sports.” There is a one-to-many mapping between topics and categories, meaning that one topic can belong to many categories. For example, a topic about hurricanes could be linked to multiple categories, such as “Science and Nature,” “Health,” and “Business.”

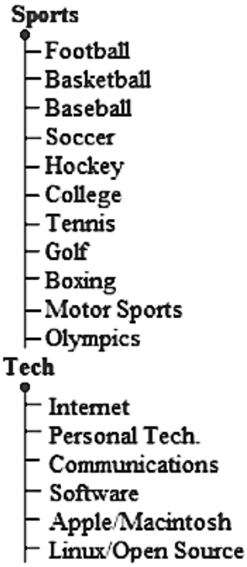

Fig. 1. Example of a Category-Topic Hierarchy

Both category and topic classification can be seen as text classification problems. However, news introduces new requirements that present difficulties for standard classification algorithms. Dealing with news is different than dealing with a document collection. New documents keep appearing that must be processed. These

[1] 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.

[2] Email:davidbis.tokushima-u.ac.jp

[3] Email:{yanjj,ren,kuroiwa}@is.tokushima-u.ac.jp

[4] Email:ren@is.tokushima-u.ac.jp

[5] Email:kuroiwa@is.tokushima-u.ac.jp

new documents can have never-before-seen information. As such, news requires dynamic online classification and discovery. Moreover, because discovery is possible, classification must be able to be done using sparse training data. These three requirements, online classification, discovery, and classification with sparse training data, pose problems for standard techniques.

This paper presents algorithms for category classification and topic discovery and classification that are shown to be highly effective. They meet all three of the added requirements of news. In addition, they are easily applied to any language with basic morphological analysis tools.

This paper will continue as follows. First, in section 2 background information and related work will be examined. Then, in section 3 the algorithm for category classification will be given. In section 4 the algorithm for topic discovery and topic classification will be shown. Next, in section 5 experimental results are given. Finally, in section 6 concluding remarks are made and future work discussed.

2 Background

This section will introduce some background information about category classification and topic classification. It will also take a look at some of the related work. First, category classification will be examined and then topic discovery and classification.

2.1 Category Classification

Category classification, for news, is a multi-label text classification problem. The goal is to assign one or more categories to a news article. A standard technique in multi-label text classification is to use a set of binary classifiers. For each category, a classifier is used to give a “yes” or “no” answer on if the category should be assigned to a text. Some of the standard algorithms for text classification that are used for binary classifiers, include Naive Bayesian Classifiers [6] and Support Vector Machines [9]. Some other approaches to multi-label classification include boosting [7] and mixture models trained by the em algorithm [5].

A category classification algorithm for news, besides having the desired high precision and recall should also be easily updated. This is because as the world changes so does the news and information about new technology, events, etc. will need to be added to the classifier. For example, in 1980 we would have had a category called “Technology,” now in 2006 we have such things as the iPod and plasma televisions, but the training data used in 1980 would not be able to cover these new technologies. By easily updatable, we mean that updating the classifier requires a simple non-exhaustive retraining or no retraining at all. Moreover, because of the amount of news that is available it is possible that retaining the training data could be a problem. As such, an algorithm that does not need the previously used training data when retraining is a plus.

The previous methods, typically, require both positive and negative examples for training data. The initial set of training data requires that each document is assigned all positive labels. Support Vector Machines offer state-of-the-art performance, however they are slow to train and updating the training data is not really a viable option. Naive Bayesian Classifiers can give good performance as well, but depending on the features used they can require for previous training data to be kept.

2.2 Topic Discovery and Classification

In recent years research has been done on automatically discovering topics and groups in an existing document collection. Wang et al introduced a method for discovering groups and topics from the relations in text [11]. Their group-topic model was designed to aid social network analysis. [8] presents work on an unsupervised topic discovery using a document clustering technique.

The above mentioned algorithms are not suitable for news though. The reason is that news is not a static data collection. It is an online stream of information that does not stop flowing. Therefore, an algorithm for news must also be able to handle things in an online manner. This idea is an obvious one and researchers have come together to take part in the Topic Detection and Tracking [1] project by NIST to examine this project.

There is however, one fundamental difference between the proposed topic classifier and those done within TDT and it is the definition of a topic. TDT defines a topic as the main event of the article. We define a topic as the main theme of the article, which is not limited to the main event. For example, for TDT an article about “North Korean Nuclear Talks” and one about “Iranian Nuclear Talks” would be two independent events and thus two different topics, however for us if the content of the articles are similar we would like to treat the two as one topic called “Nuclear talks.” Perhaps the difference is small enough that algorithms designed for TDT could easily be changed to deal with our definition.

3 Category Classification

Category classification deals with assigning one or more category labels to a news article. The categories are very broad groupings and as such, a set of primitive categories can be decided upon. Because of this, the first step in designing a category classification algorithm is to determine the primitive categories. Because we hope to use this algorithm in a cross-lingual information retrieval environment, we create categories that can span many countries and cultures. We analyzed news sites from many countries and found that the while the names were different that categories most news site shared were ”World,” ”National,” ”Sports,” and ”Business.” However, because we do not want the system to be tied to one country, “World” and “National” are not good choices. Instead we broke them down into smaller categories. The set of categories used in the proposed classifier are shown below. This list was created after examining news sites from many countries.

- Business

- Politics
- Crime and Misfortune
- Health
- Sports
- Entertainment
- Technology
- Science and Nature

In addition to the categories listed below, the world regions, as defined by the United Nations (http://www.un.org/depts/dhl/maplib/worldregions.htm) are also used. The list of regions can be seen below. However, classification of world regions is done using a simple dictionary lookup instead of the category classification algorithm. The world regions are used, because in a multinational environment it is not possible to define a national category.

- Africa
- Asia
- Europe
- Latin America
- North America
- Oceania

3.1 Algorithm Overview

The proposed algorithm builds a category model to describe a category. The category model is made up of a category name, total number of documents counter and a list of associated keywords. Each entry in the keyword list stemmed keyword, the shortest non-stemmed version of the keyword and the number of training documents it appeared in. The stemmed version of the keyword is what is used internally by the algorithm while the non-stemmed version of the keyword is used externally to show to the user. The keywords are extracted using the keyword extraction algorithm proposed by Bracewell et al. [2] and can extract high quality keywords from a single document without a document collection or corpus statistics. Moreover, it is able to work on any language that has basic morphological analysis tools. The algorithm extracts noun phrases instead of unigrams to use as keywords. It uses in-document statistical information about the noun and the individual words to weigh the extracted keywords. It was found that this approached had some advantages over using surrogate corpora when there was no existing document collection to use.

A classifier is trained for each category. Each classifier can be trained independently of each other, which allows for easy updating of category information. The classifiers are not binary, meaning they do not give a “yes” or “no” answer. Instead they give an estimate of the likelihood that the article is in the category. The likelihoods from all the categories are used to determine which of the categories should be assigned to the article.

3.1.1 Training

To train a classifier on a category, a set of training articles is needed. An automatic method for acquiring these training sets has been created, which involves creating special domain corpora. From these articles keywords are extracted using the keyword extraction algorithm previously mentioned. The keywords and the number of training articles they appeared in are recorded. This is the only training information needed by the classifier. Only positive examples for a category, i.e only documents that belong to the category, are required as training data. Updating the classifier is as simple as updating a few integer counters.

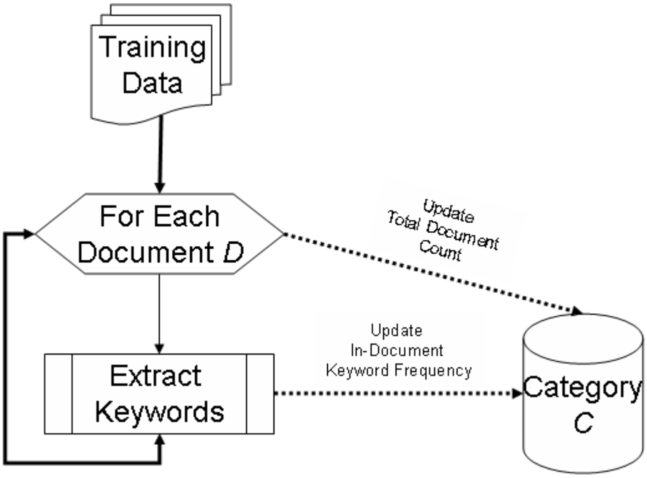
Training Data
For Each Document *D*
Update Total Document Count
Extract Keywords
Update In-Document Keyword Frequency
Category *C*

Fig. 2. Overview of Training

Figure 2 shows an overview of the training process. Each time a new article is added as training data, the "total number of documents" count is updated for the category. This count, as its name implies, tells how many training documents have been seen for this category.

Keywords are then extracted from the article. Each of the keywords are looked for in the category's keyword set. If the keyword is found then the keyword set is updated, by incrementing the keywords "In-Document" count. If the keyword is not found then the keyword is added to the category's keyword set with an initial "In-Document" count of 1. The stemmed form of the keywords are used to for matching and keeping the keyword vectors small. The shortest non-stemmed version of the keyword is also stored to use if the keywords need to be displayed to end users. After an article has been used for training data it will not be needed again and can be discarded.

Creating a category model in this fashion allows for the model to be easily updated. The probability of the st keyword given a category can be easily calculated using the keyword's "In-Document" count and the "total number of documents" count. Moreover, it allows for users to easily correct and update misclassified categories.

3.2 Classification

Classification involves four steps. First, keywords are extracted from the given article. Next, the likelihood that the article is in each category is calculated. Then, a dynamic threshold is created. Finally, categories are assigned to the article. An overview of the process can be seen in figure 3.

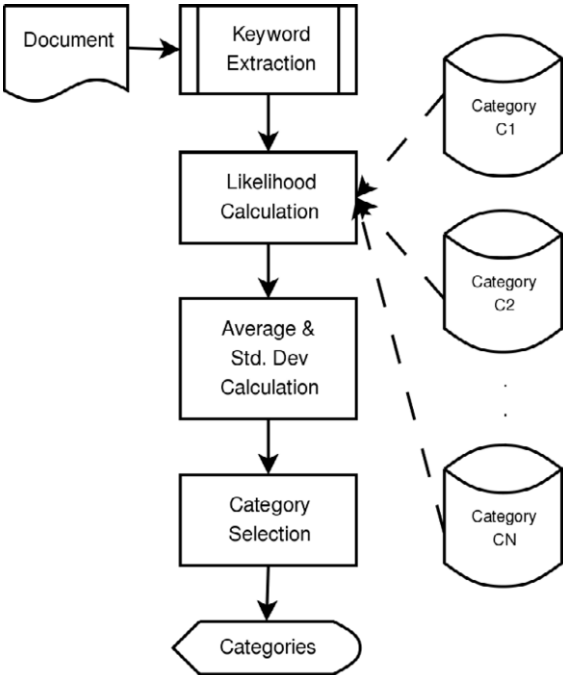

Fig. 3. Overview of Category Classification

Keyword extraction is done using the same algorithm that was used for training. The keywords are used to describe the document. With this description, a likelihood can be calculated for each category. The likelihood of a category given an article is defined in equation 1, which is the same as calculating entropy. In the equation, c_j is a category, A is the given article defined by a set of keywords and $P(k_i|c_j)$ is calculated using the "In-Document" and the "total number of documents" count.

$$Likelihood(c_j|A = \{k_1, k_2, \cdots, k_n\}) = -\sum_{i=1}^{n} P(k_i|c_j)\log\left(P(k_i|c_j)\right) \tag{1}$$

After all the likelihoods have been calculated a dynamic threshold is created, shown in equation 2, where L is the list of all likelihoods and l_i is the likelihood of category i. The mean and standard deviation of the likelihoods are used to decide the dynamic threshold. The categories that have a likelihood greater than the mean plus one standard deviation are assigned to the article. The assumption is that these

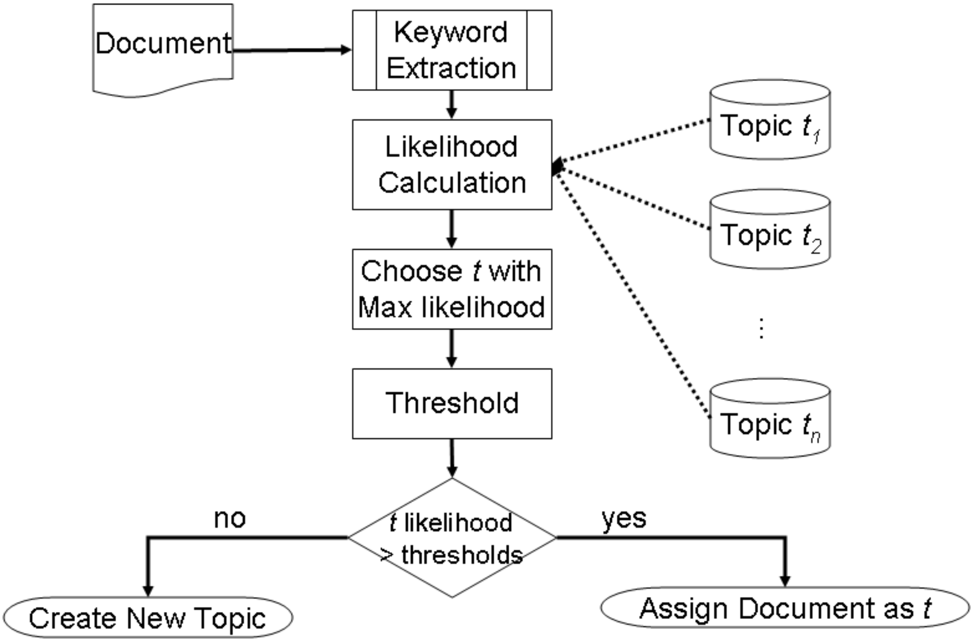

Fig. 4. Overview of Topic Discovery and Classification

categories stand out among the group and are the best choices for the article.

$$Threshold = \frac{\sum_1^{|L|} l_i}{|L|} + \sqrt{\frac{\sum \left(l_i - \frac{\sum_1^{|L|} l_i}{|L|} \right)^2}{|L|}} \tag{2}$$

4 Topic Discovery and Classification

News topics, unlike categories, are created daily as news happens. Therefore, it is more difficult if not impossible to assign an initial set of topics that can cover all articles in the foreseeable future. This means that not only is topic classification needed, but also topic discovery (also called new topic detection or new topic creation).

4.1 Algorithm Overview

An overview of the topic discovery and classification algorithm can be see in figure 4. Unlike most of the algorithms used in TDT, this algorithm requires no corpus for statistics or training data. The algorithm, first tries to classify a given article as previously seen topic. It uses a type of one-pass clustering to determine the topic of an article. Classification is done by finding the most similar topic to the article. But, because new topics arise everyday we need some mechanism to determine if the conditionally assigned topic is really a good choice. This is the job of topic discovery. The algorithm for each will described in the next in sections.

4.1.1 Topic Classification

Like the category classification algorithm, the topic classification algorithm is also based on the keyword extraction algorithm described in [2]. It calculates a similarity between each known topic and the given article using the keywords. Then it assigns the most similar article as the conditionally assigned topic.

In a similar manner as category classification, a topic is described in terms of a keyword vector. The values of the vector are the number of articles in which the keyword appeared. When the topic is used for similarity measures the values of the keyword vector are converted to a normalized frequency using add one smoothing.

Keywords are extracted from the given article and create a keyword vector where the values are the keyword scores. In order to compare the article's keyword vector and the topic's keyword vector, the two are transformed into the same vector space. This is a simple process of adding a slot for a non-existent keyword and assigning it a value of 0, figure 5 shows an example of an article and topic's keyword vectors would be transformed when their dimensions do not match.

Topic:	war	iraq	US	UK		war	iraq	US	UK	violence
	2	5	4	1	⇒	2	5	4	1	0
Article:	war	iraq	violence			war	iraq	US	UK	violence
	1	3	1		⇒	1	3	0	0	1

Fig. 5. Vector Transformation Example

After the topic and article are in the same vector space the similarity between the two can be computed. To do this the standard cosine similarity [10] is used. The cosine similarity is shown in equation 3 and t_i represents one of the topics and A the given article. Many researchers found the cosine similarity to be highly valueable in the TDT task. The topic with the highest cosine similarity is then chosen as the conditionally assigned topic.

$$CosSim(t_i, A) = \frac{t_i \bullet A}{|t_i|\,|A|} \tag{3}$$

4.1.2 Topic Discovery

Topic discovery determines if the conditionally assigned topic should remain or if a new topic should be created. This is done through dynamic thresholding, see figure 6. The first threshold compares the cosine similarity of the conditionally classified topic (t_c) and the article (A) to the cosine similarity of the article and a hypothetical topic as calculated by $NewTSim$ in equation 4. $NewTSim$ uses the information from the conditionally classified topic and the article to try and determine the cosine similarity between the article and hypothetical topic that is somewhat similar to it. The second threshold is useful when enough topics have been discovered, which in this case was determined experimentally to be 10. It checks that the cosine similarity of the conditionally classified topic is much greater than the cosine similarity of the other known topics.

$$NewTSim(t_c, A) = \frac{(0.05 \times |t_c|) \times (Mean(A) - StdDev.(A)) \times Mean(t_c)}{(|A| \times (Mean(A))^2) \times (|t_c| \times (Mean(t_c))^2)} \tag{4}$$

(i) $CosSim(t_c, A) > 0.1 \ \wedge \ CosSim(t_c, A) > NewTSim(t_c, A)$

(ii) $NumTopics > 10 \ \wedge \ CosSim(t_c, A) > (2 \times StdDev(AllTopicSims) + Mean(AllTopicSims))$

Fig. 6. Topic Discovery Thresholds

If both of the thresholds are met then the conditionally classified topic becomes officially assigned to the article. Otherwise, a new topic is created and the article is the first source of training data. Training is done in the same way as category classification.

The algorithm is very simple, but meets the strict needs of topic classification for news articles. The $NewTSim$ and other thresholds were determined through extensive experimentation. The results shown in the next section come from documents that were not used to determine the thresholds.

5 Experimentation

This section shows experimental results for Japanese and English on the proposed algorithms. First, the results for category classification will be given. Then, the results for topic discovery and classification will be given.

5.1 Category Classification

Each category, for both English and Japanese, had a classifier trained with 1,000 articles. For testing, both English and Japanese each had 800 articles extracted from a variety of online news sites. The category used by the news sites was used to determine the category assigned to the articles. For example, if the article was under sports on the site it would be sports for our categories. Table 1 show results for English and table 2 shows results for Japanese.

	Recall	Precision	F-Measure
Micro Averaged	97.21%	90.19%	92.86%
Macro Averaged	96.46%	97.99%	97.22%

Table 1
Category Classification Results for English

	Recall	Precision	F-Measure
Micro Averaged	94.5%	97.4%	95.9%
Macro Averaged	94.5%	97.6%	95.8%

Table 2
Category Classification Results for Japanese

The results show high recall and precision for both Japanese and English. The Japanese results were slightly better than those of English. This could be the result of the keyword extraction algorithm working more effectively on Japanese. While not directly comparable, the results are similar to those of other researchers, such as [3], were able to achieve with support vector machines on other corpora. This algorithm, though, has the advantage of being able to be easily updated.

5.2 *Topic Discovery and Classification*

A number of tests were performed for topic discovery and classification. First, since the topic classifier must work well even with sparse training data we compared it to other classifiers when trained using sparse data. Second, we performed tests on two different English corpora (Reuters [4] and one created by us using various online news sources). Finally, we experimented with Japanese.

5.2.1 *Sparse Training Data*

The first experiment was training with sparse data in an offline environment. As new topics are found, in the online environment, the initial training samples are small. Even with sparse training data the classifier must be able to accurately determine the topic of the news article. For comparison purposes, a Naive Bayesian Classifier (NBC), Decision Tree (DT) classifier, and Maximum Entropy (ME) classifier were used.

Each of the standard classification algorithms used all the keywords extracted from the training articles as features. The feature vector was made up of the keyword scores. For a fair comparison the proposed algorithm did not use online learning to improve its results.

Tables 3 and 4 show the macro and micro averaged recall, precision, and f-measure respectively. What can be seen from the tables is that the proposed method has a better results for just about every training size.

State-of-the-art classifiers, like Maximum Entropy, are not capable of accurately classifying when there is only sparse training data. This is seen in the results. The Naive Bayesian and Decision Tree classifiers are able to perform much better. The proposed method does achieve better results for the most part. Plus, there is no obvious way of doing online training for the Naive Bayesian and Decision Tree classifiers.

5.2.2 *English Results*

The first English test used a 1,000 article subset of the Reuters corpus [4]. This subset was made up of 11 topics. Starting with no known topics the news articles were fed into the system in random order. For evaluation, we used four measures: recall, precision, F-measure, and fragmentation factor. The recall, precision, and F-measure are used to evaluate the ability to classify. In this case we were only interested in how well the articles grouped together as topics were really in topic. Because of this we combined the false alarms or created topics that only contain

	Naive Bayesian			Decision Tree		
Size	Recall	Precision	F-Measure	Recall	Precision	F-Measure
10	40.6%	64.9%	44.0%	45.7%	61.8%	49.1%
20	51.7%	67.4%	53.0%	56.6%	62.3%	57.3%
30	54.1%	68.3%	53.5%	53.9%	61.0%	54.2%
40	47.4%	67.4%	48.5%	56.1%	60.8%	56.0%
50	54.3%	69.3%	55.2%	60.2%	60.3%	57.2%
	Maximum Entropy			Proposed		
Size	Recall	Precision	F-Measure	Recall	Precision	F-Measure
10	10.4%	9.8%	8.1%	57.7%	66.4%	56.4%
20	14.4%	11.0%	11.5%	62.2%	68.2%	58.6%
30	13.3%	7.4%	7.1%	60.9%	66.0%	65.2%
40	15.0%	14.4%	12.3%	61.7%	69.1%	67.1%
50	15.3%	14.8%	12.1%	63.4%	68.6%	65.9%

Table 3
Macro Averaged Results with Sparse Training Data

articles in the same larger topic when computing these measures. The next measure is the fragmentation factor, which tells on average how many topics were found per real topic. For example, our original set of topics may have had ”baseball,” but the algorithm could have found ”2005 World Series” and ”baseball” as two distinct topics. In this case the real topic of baseball has been fragmented into two topics. The lower the fragmentation factor the less the number of false alarms.

Table 5 shows the micro and macro averaged results over 10 different runs. The fragmentation factor for the set was 14. The classifier was able to achieve adequate results for classification, but the fragmentation factor was too high. The Reuters corpus, though, is an extremely difficult corpus. It also does not represent the type of articles that will mostly be used in everyday news.

The second test used 500 randomly extracted articles from various online English news sites, including Yahoo! News, The Washington Post, BBC and CNN. While the sites are predominantly from the U.S., we do not think this would make much a difference. This test shows results for articles that are more likely to be encountered in a real world system. The article set had topics manually assigned and resulted in 13 different topics. The experimentation was started with no known topics.

Table 6 shows the results averaged over 10 runs. As can be seen the recall and precision are much higher than that of the Reuters corpus. Moreover, the fragmentation rate fell to only 5. Since, these news articles are the type that are targeted for our system, we were happy with the results.

	Naive Bayesian			Decision Tree		
Size	Recall	Precision	F-Measure	Recall	Precision	F-Measure
10	45.8%	45.8%	45.8%	49.2%	49.2%	49.2%
20	50.3%	50.3%	50.3%	49.2%	49.2%	49.2%
30	54.1%	54.1%	54.1%	56.8%	56.8%	56.8%
40	39.0%	39.0%	39.0%	56.3%	56.3%	56.3%
50	46.5%	46.5%	46.5%	54.6%	54.6%	54.6%
	Maximum Entropy			Proposed		
Size	Recall	Precision	F-Measure	Recall	Precision	F-Measure
10	10.2%	10.2%	10.2%	54.6%	54.6%	54.6%
20	17.0%	17.0%	17.0%	57.5%	57.5%	57.5%
30	11.0%	11.0%	11.0%	51.2%	51.2%	51.2%
40	13.9%	13.9%	13.9%	55.8%	57.3%	56.5%
50	14.1%	14.1%	14.1%	55.9%	58.0%	57.0%

Table 4
Micro Averaged Results with Sparse Training Data

	Recall	Precision	F-Measure
Micro Averaged	77.6%	77.6%	77.6%
Macro Averaged	75.6%	80.9%	76.2%

Table 5
Topic Discovery and Classification Results for Reuters

	Recall	Precision	F-Measure
Micro Averaged	96.0%	96.0%	96.0%
Macro Averaged	93.74%	96.05%	94.67%

Table 6
Topic Discovery and Classification Results for Non-Reuters Data

5.2.3 Japanese Results

The final test used 1,000 randomly extracted articles from various online Japanese news sites, including Mainichi Shimbun, Asahi Shimbum and Yomiuri Shimbun. The article set had topics manually assigned and resulted in 10 different topics. The experimentation was started with no known topics.

Table 7 shows the results averaged over 10 runs. The recall, precision and F-

	Recall	Precision	F-Measure
Micro Averaged	91.0%	91.0%	91.0%
Macro Averaged	90.04%	92.08%	90.61%

Table 7
Topic Discovery and Classification Results for Japanese

measure for Japanese were all very high. The fragmentation factor, though, was also high at 11.3. The results were a little worse than those of the English non-Reuters test. This is possibly due to the Japanese use of Chinese characters (kanji). These characters help to disambiguate words, but can also make naive word matching difficult.

6 Conclusion and Future Work

This paper presented algorithms for category classification and topic discovery and classification of news articles. The news domain presents challenges that other domains do not. Dealing with online news demands online classification, topic discovery and classification with sparse training data.

The algorithms presented in this paper were based on a keyword extraction algorithm that is capable of dealing with multiple languages and does not requrie a document collection or corpus statistics. Because of this, the presented algorithms were also able to work with multiple languages, in this case Japanese and English. The results show that, while there is room for improvement, these simple algorithms can achieve good results.

The category classification algorithm can train its classifiers independent of each other and is easily updated. The topic discovery and classification algorithm is unsupervised and learns in an online manner. In the future, we hope to test the algorithms on even larger corpora. We also hope to add named entity recognition to the topic classifier, in hopes that it will help. In addition we will look at ways of improving the algorithm so that the fragmentation is much more acceptable.

References

[1] Allan, J., J. Carbonell, G. Doddington, J. Yamron and Y. Yang, *Topic detection and tracking pilot study: Final report*, in: *Proceedings DARPA Broadcast News Transcription and Understanding Workshop*, 1998, pp. 194–218.

[2] Bracewell, D. B., F. Ren and S. Kuroiwa, *Multilingual single document keyword extraction for information retrieval*, in: *Proceedings of the IEEE International Conference on Natural Language Processingand Knowledge Engineering*, Wuhan, China, 2005.

[3] Joachims, T., *Text categorization with support vector machines: learning with many relevant features*, in: C. Nédellec and C. Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, 1398 (1998), pp. 137–142.

[4] Lewis, D. D., Y. Yang, T. Rose, and F. Li, *Rcv1: A new benchmark collection for text categorization research*, Journal of Machine Learning Research **5** (2004), pp. 361–397.

[5] McCallum, A., *Multi-label text classification with a mixture model trained by em*, in: *AAAI'99 Workshop on Text Learning*, 1999.

[6] McCallum, A. and K. Nigam, *A comparison of event models for naive bayes text classification*, in: *AAAI/ICML-98 Workshop on Learning for Text Categorization*, 1998.

[7] Schapire, R. E. and Y. Singer, *Boostexter: A system for multiclass multi-label text categorization*, Machine Learning **39** (1998), pp. 135–168.

[8] Schwartz, R., *Unsupervised topic discovery*, in: *Proceedings of Workshop on Language Modeling and Information Retrieval*, 2001.

[9] Tong, S. and D. Koller, *Support vector machine active learning with applications to text classification*, in: P. Langley, editor, *Proceedings of ICML-00, 17th International Conference on Machine Learning* (2000), pp. 999–1006.

[10] van Rijsbergen, R. C. J., "Information Retrieval: Second Edition," Butterworth-Heinemann, 1979.

[11] Wang, X., N. Mohanty and A. McCallum, *Group and topic discovery from relations and text*, in: *The 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining Workshop on Link Discovery: Issues, Approaches and Applications (LinkKDD-05)*, 2005.