Automatic Detection of News Articles of Interest to Regional Communities

Robin M. E. Swezey†, Hiroyuki Sano†, Shun Shiramatsu†, Tadachika Ozono† and Toramatsu Shintani†,

†Dept. of Computer Science and Engineering, Graduate School of Engineering, Nagoya Institute of Technology
Aichi, 466–8555 Japan

Summary
In this paper, we devise an approach for identifying and classifying contents of interest related to geographic communities from news articles streams. We first conduct a short study on related works, and then present our approach, which consists in 1) filtering out contents irrelevant to communities and 2) classifying the remaining relevant news articles. Using a confidence threshold, the filtering and classification tasks can be performed in one pass using the weights learned by the same algorithm. We use Bayesian text classification, and because of important empiric class imbalance in Web-crawled corpora, we test several approaches: Naïve Bayes, Complementary Naïve Bayes, use of {1,2,3}-Grams, and use of oversampling. We find out in our testing experiment on Japanese prefectures that 3-gram CNB with oversampling is the most effective approach in terms of precision, while retaining acceptable training time and testing time.

Key words:
Web Intelligence, Natural Language Processing, Machine Learning, Semantic Web.

1. Introduction

We developed an e-Participation Web platform, O2, for regional communities. The platform aims at supporting citizen e-Participation in ongoing regional debates by gathering and openly publishing news and opinions. Structuring citizens' awareness of regional issues and sharing structured data are two requirements in conducting productive discussions about various issues. O2 consists of three tools: Sophia, SOCIA, and *citispe@k*. Sophia is a mining and intelligent annotation platform that classifies and clusters news articles and tweets. SOCIA is a data set and the ontology of the same name, developed to support debate, and based on Linked Open Data (LOD). The goal of this project is to archive information and discussion about events occurring in regional communities. *citispe@k* is an application to support the discussion of regional issues identified by Sophia, using annotated data stored and SOCIA.

In order to gain better engagement and involvement from citizens, information from the Web (e.g articles, blogs, tweets) needs to be thoroughly classified by region, and then presented to citizens in an understandable way. Using our platform and ontology, news and opinions are structured and linked with regional issues, and the data is openly published on the Web using the OWL-based ontology of SOCIA. Through this process, e-Participative data becomes re-usable and transparent. Transparency is a requirement of Government 2.0 initiatives.

Data mined from the Web is structured in the form of events by region, which are then used as discussion seeds to further build SOCIA. Citizens then create discussion topics out of each seed, e.g a cluster of news related to the same event, and input their opinions by using the system, among other functionalities. The system first collects news articles and microblog posts along with necessary metadata (dates, emission sources, etc). It then classifies this crawled data by region and filters out noise irrelevant to the interest of regional communities or current events.

In this paper, we focus on the automatic filtering and classification of news articles by region. With our method, only one Bayesian classifier needs to be trained, which can be done in a short amount of time, and filtering and classification follow this. However, there are many assumptions made in text classification research, the most problematic one being the assumption of classes being equally balanced. Most of the testing corpora used for learning algorithms, such as Reuters and Newsgroups, are balanced. However, in real-case applications, this is rarely the case.

The rest of this paper is organized as follows. In section 2, we present related works. In section 3, we introduce our system's architecture and approach, and detail the theoretical background. In section 4, we conduct two experiments for classification and filtering to find the best approach to the problem at hand, and discuss the results. We summarize our contributions and conclude the paper in section 5.

2. Related Works

Several challenges have to be met to use the advantages of public corpora. The corpus can be highly imbalanced, as is the case in most Web corpora built by crawling or page scrapping of site contents. In the present paper, this is the

Manuscript received June 5, 2012
Manuscript revised June 20, 2012

case for news corpora once they are labeled according to geography, which are of interest and relevant for local communities. For each class *c* of a possible set of classes *C* where *c* is a region, prefecture (sub-region) or city, some classes contain a much larger number of documents than others. Capital cities and highly populated areas normally get more news than other regions.

There are various ways of classifying text, but in this work we use Bayesian text classification, using Naïve Bayes (NB) [1] and Transformed Weight-normalized Complementary Naive Bayes classification (TWCNB) [2]. Both algorithms are known to perform well on text classification problems, and to have a shorter training time compared to other learning algorithms such as Support Vector Machines, with similar performance in accuracy. Although it is generally assumed that CNB performs better than NB, in previous works we have shown that this is not always the case in experimental frameworks that diverge from known problems, particularly problems affected by high class imbalance [3].

As shown in our experiments in Section 4, neither algorithm can be used straightforwardly in the case of geographical classification using a Japanese news articles corpus. The bias created by class imbalance being too high, it needs either to be supervised by a meta-algorithm (subject of another work [4]), or other approaches to tackle class imbalance. Such meta-algorithms for improving classifiers already exist, like AdaBoost. However, if the number of classes is to become relatively large when compared to classical examples of multi-class Bayesian classification problems (20NewsGroups, Reuteurs, etc), AdaBoost improvement [5] can lead to relatively longer training time over several iterations. Oversampling and undersampling [3] also are known methods for alleviating class imbalance problems, although they can also affect the prior class calculation and other variables in Bayesian classification learning.

Also, many methods described do not consider using n-gram features instead of terms and words. Unigram models do not take into account the interdependence of term features. This leads to the bag-of-words model, and turns out to generate a multinomial distribution over words independently. While methods such as linear discriminant analysis [6] and latent Dirichlet allocation [7] alleviate this problem using dimensionality reduction instead of / on top of the use of TF*IDF features, n-gram consists in extracting combinations of features from the text, and is thus closer to linear basis expansion and dimensionality augmentation.

In this work, along with an approach that calls for training of only one learning algorithm to filter and classify news articles according to geography, we propose the use of oversampling combined with use of n-gram features in order to improve precision while retaining acceptable training and testing times. This makes the method efficient and easy to re-use when new information is added to the datasets, which occurs permanently with news articles.

3. Proposed Approach

2.1 Overview

The aim of the original project from which this research stems is to create a citizen involvement platform. The purpose of the platform is to assess the population's concern about social issues as well as increase public involvement from the Web. To this end, structuring data is a mandatory step in order to comprehend and make the information easily understandable. Data such as news articles must be filtered, classified and then clustered. This paper describes the filtering and classification tasks. Clustering is the subject of another paper [8].

A stream of data such as news articles requires its regions of interest to be identified and clustered adequately (Fig. 1). Here, the term *region of interest* (ROI) denotes in the general sense a selected subset of samples within a dataset identified for a particular purpose, not to be mistaken with the *geographical region* against which we wish to classify the utterances. Since the ROI at hand are actual contents, we simplify by calling them Contents Of Interest (COI). As can be seen in Figure 1, the classification process happens at first in a chain of operations whose aim is to structure the data that is mined from various streams, such as the Twitter streaming API in the case of microposts, or news site RSS feeds in the case of news articles. Contents from the feeds is mined and then processed by the classification module having a model trained by a learning algorithm, for which we evaluate several training approaches in Section 4.

2.2 Pre-processing

We first pre-process the text by converting each document to a morpheme string. Although it is possible to use arrays of morpheme-type objects, we convert the text to a string of space-separated Japanese morphemes for two reasons: First, to avoid object overhead, and for easier compatibility with other software. Second and more importantly, to conserve the order of sentences so that 2-gram and 3-gram features can be extracted easily. Each document undergoes the following steps:

1. Decomposition into morphemes using MeCab, a morphological analysis package.
2. Filtering: elimination of stop-words, acceptance of content-words only, stemming.

3. Conversion to a morpheme string for storage, learning and testing.

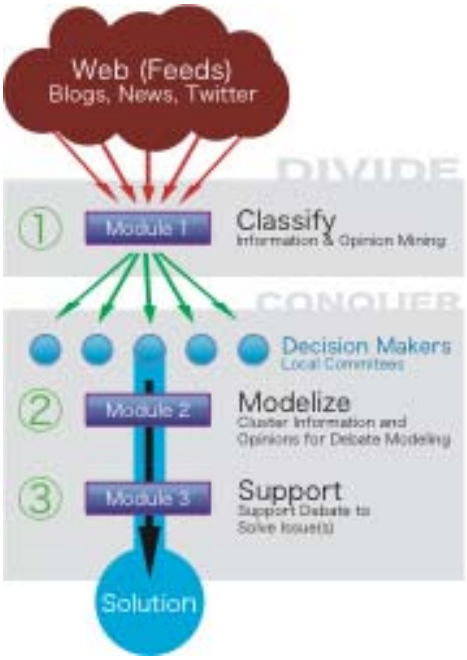

Fig. 2 Outline of the System

2.3 Classification by Region

After mining, we perform classification of news articles and tweets by geography (against the 47 prefectures/sub-regions of Japan). To this end, we use Naive Bayes text classification with TF*IDF(t, d, D) = tf(t, d) × idf(t, D), a.k.a the TF*IDF metric, defined by:

$$TF(t,d) = \vec{t} \cdot \vec{d} \text{ (\#t appearing in d)}$$

$$IDF(t,D) = \log \frac{|D|}{1+|\{d \in D : tf(t,d) > 0\}|}$$

where let t be a N-gram term, d a document, and D a document set in a corpus.

Naive Bayes itself assumes that words are drawn independently from a multinomial distribution and that, given a class label *c* from the set of class labels *C* (in this work, the regions of Japan), the probability of a class label given a document is:

$$P(c|d) = \frac{P(d|c) \times P(c)}{P(d)}$$

```
Input: Imbalanced Training Data
Output: Oversampled Balanced Training Data

procedure Oversample(ITD)
begin
  max ← 0
    for each class label c do
      if |training data (c)| > max then
        max ← |training data (c)|
      endif
    enddo
  for each class label c do
    i ← 0
    while(|training data (c)| < max) do
      add sample i from training data(c) to training data(c)
      i ← i + 1 % |training data (c)|
    enddo
  enddo
end.
```

Fig. 1 Oversampling Algorithm

A variant of Naïve Bayes text classification is the Transformed Weight-normalized Complementary Naive

Bayes algorithm [2], where the score for classifying a document *d* into a class *C* is calculated as:

$$\hat{w}(c|d) = \log P(c|d) - \sum_t tf(t,d) \log \frac{1+\sum_{k=1}^{|C|} tf(t,c_k)}{N+\sum_{k=1}^{|C|}\sum_{x=1}^{N} tf(t_x,c_k)}$$

where *N* denotes the size of the vocabulary. Further improvements of the classification algorithm are the subject of another work.

2.4 Filtering

To decide whether or not contents should be filtered out, our approach is to use a confidence threshold to determine the decision boundary for out-of-domain data, where the classifier's confidence score γ is determined by:

$$\gamma(d) = \hat{w}(c_1|d) - \hat{w}(c_2|d)$$

with:

$$c_1 = \arg\max_{c \in C} \hat{w}(c|d),\ c_2 = \arg\max_{c \in C \setminus \{c_1\}} \hat{w}(c|d)$$

where c_1 and c_2 are respectively the first and second class labels where the test document *d* weighs the most, the first two classes for which the classifier is most confident. A good threshold is one that gives better precision to the classifier, normally at the expense of recall.

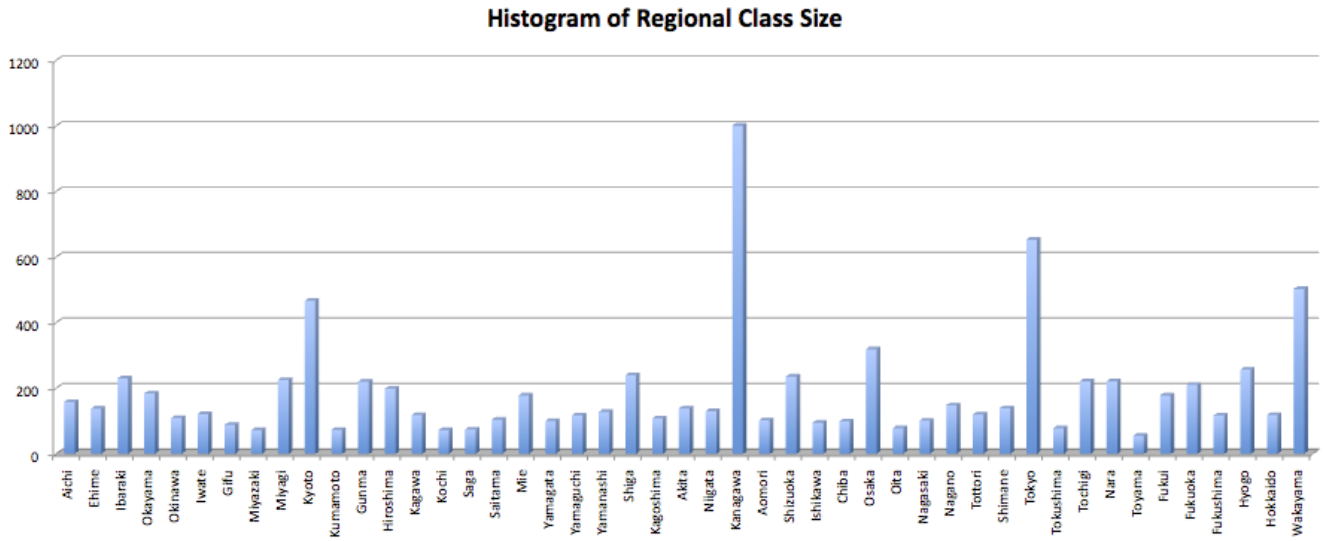

Fig. 3 Histogram of Class Size by Regional Class Label

2.5 Oversampling Process

The method chosen for re-sampling the training data in order to counter the effect of class imbalance is random oversampling. Assuming that the samples are in a random order, the oversampling process is as described in Fig. 2.

4. Experimental Results

4.1 Experimental Setup

Experiments. We conducted two experiments. The *classification* experiment was conducted for finding the ideal algorithm and settings depending on training and testing time as well as precision. The *threshold* experiment consists in finding an ideal threshold for real-world application of the system, by varying confidence value (as described in Sec. 2.2), so that precision increases, normally at the expense of recall.

Corpora. We gathered a corpus of 8,811 news articles related to Japanese prefectures crawled from Yahoo! Japan News in the period between June 13 and July 12, 2011. We call it the *regional* corpus. Another corpus was built with 1,133 news articles that do not relate to any regions, prefectures or geographical communities. We call it the *noise* corpus. The noise corpus is used in the threshold experiment, along with the regional corpus, to find the ideal confidence threshold of the classifier. Fig. 3 gives a histogram of class imbalance in the regional corpus. When the training data undergoes oversampling, the amount of documents obtained is 47,047 given that the maximum class size in the present distribution is 10,001.

4.2 Classification Experiment

Method. We test two Bayesian text classification algorithms, Naïve Bayes (abbreviated as NB) and Transformed Weight-normalized Complementary Naïve Bayes (shortened as TWCNB or CNB). The metric used is the TF*IDF metric as described in Sec. 3. As features, we test 1-gram, 2-gram and 3-gram terms. The number of models trained is six, since CNB-trained weights can be used for NB testing as well. We tested NB and CNB over each model. Two tests were conducted. The first is a closed test, which is known to give low bias in contrast to higher variance since it is more prone to overfit its own test data and thus normally gives better training accuracy at the expense of higher variance [6]. In the closed test, the original training set constitutes the testing set in all cases. When no oversampling is conducted, the testing set is the training set. When there is oversampling, the testing set is the original training set (not oversampled). The second test is a 10-fold test, which normally has lower accuracy than the closed one, but is better to assess variance, since in a real-case application all the training data is used and variance cannot be assessed without testing unknown samples. In this research, since the first test was sufficient to understand the need for oversampling, the second test (10-fold) was conducted with oversampled training sets only. Training and testing data were separated into 10 different folds and the training data of each fold was oversampled, then the learning algorithm was trained and tested on the test data of the fold. In each case, the training

Class Resampling		None			Random Oversampling		
N-Grams		1	2	3	1	2	3
Model Training Time		5'1"	5'1"	5'1"	5'10"	9'54"	137'30"
CNB	Precision	25.15%	8.05%	8.40%	78.03%	99.03%	99.40%
	Testing Time	1'2"	2'48"	5'40"	0'57"	2'50"	9'44"
NB	Precision	11.90%	7.68%	8.03%	25.42%	70.97%	84.40%
	Testing Time	0'42"	2'06"	4'16"	0'45"	2'8"	7'25"

Tab. 1 Efficiency of classification algorithms with respect to oversampling and gramization (closed test)

time is the time needed to train one of the six models. The testing time is the time necessary to test all samples. Precision over the training set is given by the formula below:

$$P = \sum_{c \in C} \left(\frac{TP}{TP + FP} \right)_c = \frac{\sum_{c \in C} TP_c}{|D_{training}|}$$

where *TP, FP* are defined respectively as the cardinals of the *True Positives, False Positives* sets of documents obtained when testing all samples of the training set against one class *C* of the domain of class labels *D*. The classifier's precision can also be defined as the ratio of the number of correctly classified instances over the cardinal of the testing set which we express as |D|.

Results. Results are presented in Table 1. Color gradation is used to show where the algorithm performs more or less desirably. Overall, it can be seen that NB has inferior precision when compared to CNB. The impact of oversampling and the effect of class imbalance can be witnessed in a further degradation of performance when {2,3}-gram features are used without re-sampling. Performance with the use of n-gram in both cases is likely inversely correlated. Whereas training and testing times remain on a similar order of magnitude over all, training with 3-grams takes more than 2 hours to train, given the amount of 3-gram weights to calculate and the oversampled class sizes. For each of the 1-gram, 2-gram and 3-gram trainings, the resulting vocabulary sizes were respectively 62 275, 647 755, and 1 975 202 features/words. By looking at the results, we observe that the framework in which we obtain the best precision as well as reasonable training and testing times is by training a CNB algorithm with oversampling of the classes and 3-gram.

4.3 Filtering Experiment

Method. We first run a control experiment limited to the testing set, in which no threshold is set, in order to measure the range of confidence of the classifier. For each

Class Resampling		Random Oversampling		
N-Grams		1	2	3
Model Training Time		4'6"	8'10"	15'24"
CNB	Precision	43.00 %	81.21 %	85.95 %
	Testing Time	0'8"	0'25"	0'55"
NB	Precision	7.52%	20.70 %	29.46 %
	Testing Time	0'7"	0'25"	0'54"

test sample that undergoes classification, a confidence score is measured by recoding the value of the formula given in Sec. 2.3. At the end of the control experiment, the minimum and maximum confidence scores are retrieved as respectively γ_{min} and γ_{max}. $[\gamma_{min}, \gamma_{max}]$ constitutes the range of confidence of the control experiment. We then conduct *n* experiment iterations where the threshold at iteration *i* is set at the confidence value given by:

$$\gamma_i = \frac{i}{\tau}$$

with:

$$\tau = \eta \frac{|\gamma_{max} - \gamma_{min}|}{n}$$

where η is a shrinking parameter adjusted over several sets of iterations to obtain a meaningful continuous set of points. Measures given in the results were made with η set to 0.03. The precision *P* and recall *R* of the system are given by the formulae below.

$$P = \frac{\sum_{c \in C} TP_c}{\sum_{c \in C} (TP + FP)_c + FP_C}$$

$$R = \frac{\sum_{c \in C} TP_c}{\sum_{c \in C} (TP + FN)_c + FN_C}$$

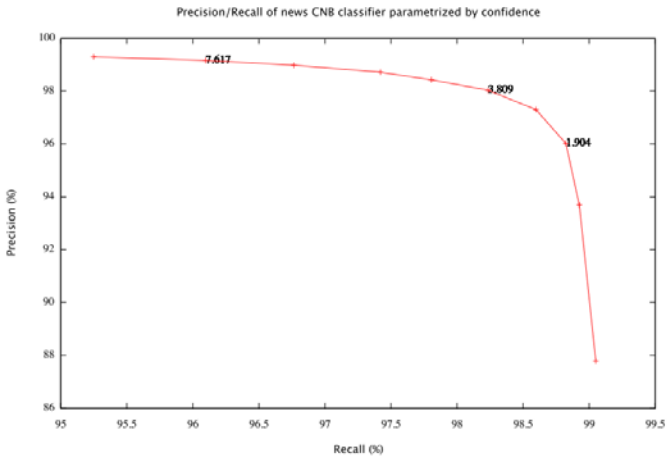

Fig. 4 Precision/Recall Parametrized by Confidence Threshold

where FP_C and FN_C are the False Positives and False Negatives of the overall domain of classes, respectively samples of the noise corpus that have been accepted by the classifier and samples of the regional corpus that have been dropped by the classifier, with respect to his threshold value. In this case, the sets counted by each FP_c and FN_c do not overlap with the sets of FP_C and FN_C. This is because in the two-step filter/accept process of the test, accepted noise and rejected in-domain data are included in the sets counted by FP_D and FN_D but since they are not accepted as classified data, the system does not count them in FP_C and FN_C. Fig. shows the evolution of precision and recall according to γ. The F_1 score F_1 is used to assess the best precision/recall combination and decide for the optimal confidence threshold. F_1 is given by the following:

$$F_1 = 2\frac{PR}{P+R}$$

Results. Figure 4 shows precision P in function of recall R according to the confidence threshold parameter γ. Measures including the are given on Table 3. It is shown that the classifier can retain a reasonable recall at 95.24% while at the maximum precision of 99.29%, which is desirable performance since we opt for precision at the expense of recall. Thus as a threshold for the real-world application we choose to use a precision-optimal value of about 8.56.

F_1	P	R	γ	FP_C	FN_C
93.08	87.79	99.04	2.64×10^{-4}	1133	0
96.23	93.68	98.92	0.95	521	27
97.40	96.02	98.82	1.90	301	43
97.94	97.29	98.59	2.85	191	73
98.13	98.03	98.23	3.80	125	107
98.11	98.42	97.80	4.76	92	148
98.06	98.71	97.42	5.71	73	189
97.86	98.98	96.76	6.66	51	249
97.60	99.15	96.09	7.61	38	311
97.22	99.29	95.24	8.56	31	391

Tab. 3 F_1 score, Precision, Recall, Threshold, Domain-FP, Domain-FN

4.4 Discussion

The decrease/stagnation of performance in precision when training with {2, 3}-grams can be attributed to the fact that augmenting the number of features results in a sparser feature space, which is an impediment to better accuracy when there is high class imbalance [3]. Thus, we find that the way to adequately exploit n-gram features to get better precision is through conjugate use of oversampling and basis expansion (use of n-grams). It then becomes possible to train a classifier in a relatively short amount of time, which is of acceptable efficiency for classification as well as for filtering, as shown in our experiments.

A side effect of oversampling to be discussed is the effect on the class priors. Namely, when the dataset is balanced, the class labels become equiprobable when the probabilistic learning algorithm is trained. However, when we carried out the 10-fold cross-validation, the drop in precision is not so big as to consider that equal class priors in this case of Bayesian classification are an impediment to precision for the class labels. However, this is not always the case. Depending on the training set, oversampling may not always be the method of choice.

5. Application

We have built a system for citizens to make direct use of the classifier developed in this work. Citispe@k is a debate support system based on SOCIA, implemented as a Web application, usable on Web browsers. Recall that SOCIA is the dataset for which the system presented in this work annotates contents (such as news articles) after they are filtered and classified. One such annotation is done by region. For mobility and reach, Citispe@k supports Web browsers running on smart phones and tablets. The origin of the word citispe@k is that citizens speak about social issues and current events of the regions they live in. Users can discuss or sort out regional issues by referencing news articles, tweets or other relevant resources on the Web using citispe@k. By creating discussion topics or inputting opinions on the system, those topics or opinions are also stored as Linked Open Data in SOCIA, adding more to Linked Open Data naturally.

Fig.5 shows a screenshot of citispe@k. The screenshot has lists of events or related information. Events recently updated are listed on the left side of the screenshot. First the system shows all events, but users can limit the list to show only events related to their region. When users select an event from a region, only classified articles belonging to this region using the present research are clustered with the event, which consists of a group of co-occurring keywords delimited by a given time window. Details about Citispe@k are given in another work [9].

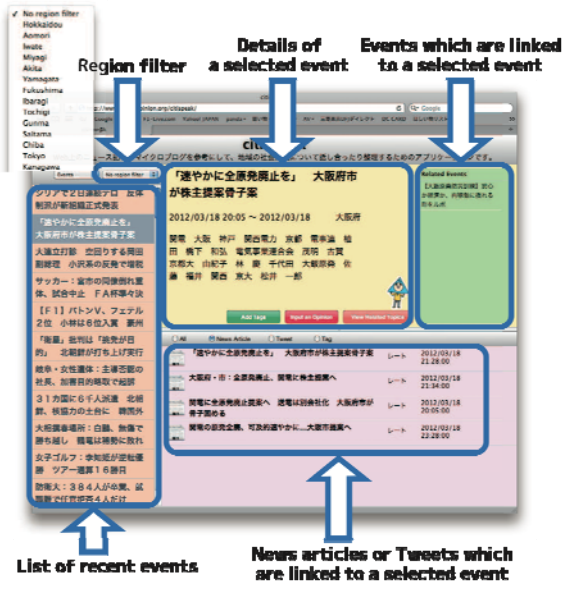

Fig. 5 Screenshot of citispe@k

6. Conclusion

We have introduced an approach for identifying and classifying contents of interest related to geographic communities from news articles streams. One of the main challenges when building a real-world classifier is to address the problem of class imbalance in ad-hoc text corpora, and the problem of filtering out-of-domain data. After conducting a study on related works, we presented our approach, which consists in 1) filtering out contents irrelevant to communities and 2) classifying the remaining relevant news articles. Using a confidence threshold, the filtering and classification tasks can be performed in one pass using the weights learned by the same algorithm. We used Bayesian text classification and tested several approaches: Naïve Bayes, Complementary Naïve Bayes, use of {1,2,3}-Grams, and use of oversampling to tackle class imbalance. In our experiments, we found that using 3-gram CNB in conjunction with oversampling is the most effective approach in terms of precision, while retaining acceptable training time and testing time. We then devised an approach for filtering out extraneous contents using a confidence threshold, which shows real-world-ready performance when tested for precision and recall with a noise corpus. We then finished with the presentation of a working prototype that utilizes the classification system developed in this research.

Acknowledgments

This work was supported and promoted by the Strategic Information and Communications R&D Promotion Programme (SCOPE) [12], Ministry of Internal Affairs and Communications, Japan.
Apache Mahout[3] was used to train the algorithms.

References

[1] Harry Zhang. The optimality of naive bayes. In Valerie Barr and Zdravko Markov, editors, Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference (FLAIRS 2004). AAAI Press, 2004.
[2] Jason D. M. Rennie, Jaime Teevan, and David R. Karger. Tackling the Poor Assumptions of Naive Bayes Text Classifiers. In Proceedings of the Twentieth International Conference on Machine Learning, pages 616–623, 2003.
[3] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. Intell. Data Anal., 6(5):429–449, October 2002.
[4] R. Swezey, S. Shiramatsu, T. Ozono, and T. Shintani. An Improve- ment for Na¨ive Bayes Text Classification Applied to Online Imbalanced Crowdsourced Corpora. In Proceedings of the Twenty Fifith Interna- tional Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (IEA-AIE), 2012.
[5] Yoav Freund and Robert E. Schapire. A short introduction to boosting. In In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, pages 1401–1406. Morgan Kaufmann, 1999.
[6] T. Hastie, R. Tibshirani, and J. H. Friedman. The Elements of Statistical Learning. Springer, corrected edition, July 2003.
[7] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. Journal of Machine Learning Research, 3:993–1022, March 2003.
[8] N. Hirata, H. Sano, R. Swezey, S. Shiramatsu, T. Ozono, and T. Shintani. A web agent based on exploratory event mining in social media. In In Proceedings of 3rd IIAI International Conference on e-Services and Knowledge Management (IIAI ESKM 2012), 2012.

[1] http://www.soumu.go.jp/main_sosiki/joho_tsusin/scope/index.html
[2] http://www.soumu.go.jp/soutsu/tokai/tool/kohosiryo/hodo/22/08/0809-1.htm
[3] http://mahout.apache.org/

[9] R. Swezey, H. Sano, N. Hirata, S. Shiramatsu, T. Ozono, and T. Shin- tani. An e-participation support system for regional communities based on linked open data, classification and clustering. In IEEE CS Press, edi- tor, Proceedings of the 11th IEEE International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC 2012), 2012.

Robin M. E. Swezey is currently a PhD student at Nagoya Institute of Technology, Japan. He received his Engineering and MS degree in computer science from the French School of Electronics and Computer Science (EFREI) in 2009. Previously, from 2008 to 2009, he did research on Multi-Agent Systems at Nagoya Institute of Technology as a special research student. His focus is on Data Mining, Machine Learning, Natural Language Processing and the Semantic Web applied to real-world problems.

Hiroyuki Sano is currently a PhD student at Nagoya Institute of Technology, Japan. He received his BS in computer science from Nagoya Institute of Technology, Japan, in 2008 and his MS in 2010. His focus is on intelligent Web technologies.

Shun Shiramatsu received his PhD in information science from Kyoto University, Japan, in 2009, and his MS from Tokyo University of Science in 2003. He is currently (2012) an Assistant Professor of Computer Science. His research interests include discussion support and conversation modeling.

Tadachika Ozono received his MS and PhD in computer science from Nagoya Institute of Technology of Nagoya City, Japan, in 1998 and 2000, respectively, and is currently an Associate Professor of Computer Science there. His main research interest (2012) is Web intelligence.

Toramatsu Shintani received his MS in industrial engineering and his PhD in computer science from Tokyo University of Science in 1982 and 1993, respectively. He was a research staff member in Fujitsu Limited from 1982 to 1993. He is currently (2012) a Professor of Computer Science at Nagoya Institute of Technology of Nagoya City, Japan. His current research interests include decision support systems and Web intelligence.