A Study on Regional Contents Classification using Dataset Structure Refinement for an Intelligent e-Participation Platform

Robin M. E. Swezey

March 5, 2013

Contents

Chapter 1

Introduction: A Platform for Structuring Contents

1.1 Introduction

Sophia was originally developed with the aim of providing an intelligent platform of which the objective is to assess concern and increase public involvement of the Japanese population about social issues.

In the last years, research about topics such as the problem of information overload and overcustomization of data have been trending in research [Greengard, 2011]. We can witness a certain lack of public interest in social issues on the Web, which can be assessed by looking at recurring important trends on social networking sites, in which social issues do not appear. There is also the lack of a targeted solution to this lack of public interest and participation in the political and social life of local communities. eParticipation tools, for example, do not use outside data and are not scalable to the Web [Macintosh et al., 2009, Freschi et al., 2009]. Tools such as readers and social dashboards are merely aggregators with few intelligent functions. However, some research has been done and tools developed to monitor the pulse of social networks and sentiment analysis about certain issues, such as financial sentiment [Bollen et al., 2011]. Still, there are no tools to assess concern about social issues and develop informed debate about social issues that scale to the Web.

Thus, we proposed to build an intelligent platform that enables us to assess concern and increase public involvement from the population.

Challenges to meet such a goal are mainly the scale and diversity of information. Data sources are various and numerous: input data of the system, coming from the outside Web, is mainly fuzzy and unstructured information coming from normal news sites as well as social networking sites, or even public debate logs. Thus, the platform needs to be modular and easily adaptable, and also scalable to the quantity of data to analyze.

In this chapter, we focus mainly on Sophia as a platform for intelligent processing of fuzzy and big data. We believe the platform can have uses that extend further than its original aim. which we consider in this chapter as one case of application.

In the next sections, we state the main contributions and originality of this work.

1.2 Main Contributions

In this work, we will use Bayesian classification to classify documents. Using better algorithms such as [Rennie et al., 2003] is not enough and can actually result in a drop of performance when using real-world datasets for applications such as an intelligent e-Participation platforms. Also, for filtering noise, it turns out that it is possible to train a single Naive Bayes model for classification and then use it for filtering noise.

1. After conducting a survey we propose the meta-algorithm SAHB (Chapter 2) for dataset restructuring which is a novel method of re-sampling for Bayesian text classification based on trees along with clustering of classes. The method shows promising results with low bias on a training fit on the Wikipedia Japanese dataset. When generalized this algorithm can be the better approach for problems with very large numbers of classes that are imbalanced, thanks to its hierarchization, node clustering approach and log-time complexity.

2. We have devised a method to use only one trained Bayesian model to be able to filter out irrelevant contents without having to train one more algorithm (e.g. Support Vector Machines), to filter local against global contents. Our method shows excellent results in precision/recall in Chapter 3.

3. For text classification, increasing the number of grams and using the better algorithm TWCNB (the straightforward approach which we use

as a control experiment) is not enough: dataset structure refinement through adequate re-sampling (Chapter 3) and/or hierarchization (Chapters 2 and 6) is necessary, otherwise there can actually be an accuracy and performance drop when using better methods than simple Naive Bayes and more features which would normally help on a training fit.

4. We contributed a novel e-Participation system in Chapter 4 that utilizes classification, clustering and annotation with an adequate ontology. Through a qualitative comparison we believe the system is best fit to support debate among local citizens in the current context of the Internet, upon improvements.

1.3 Overall Structure

1.3.1 Research Modules

Our research is divided into three modules. In the particular case of application to the broader project from which Sophia originated (debate support), this can be seen on Figure 1.2.

1. Mining and Learning Outside data from the Web must be mined, processed and classified before any attempt at structuring it in an understandable manner can occur. This is where we use and conduct research on various information retrieval and machine learning algorithms.

2. Modeling and Structuring Once data is processed, classified, clustered, it can be structured according to a relevant debate and discussion ontology which answers the needs for a citizen's participation and involvement tool. We call this ontology Socia, and it is used by Sophia.

3. Support and Comprehension Using data structured in module 2, we can present information about social issues in various forms and on various platforms to different distributions of the population following different dimensions. E.g a trending topics interface where information is clustered along a geographical axis and a short-term temporal axis, or an opinion map where the temporal axis is more long-term (evolution of social topics during

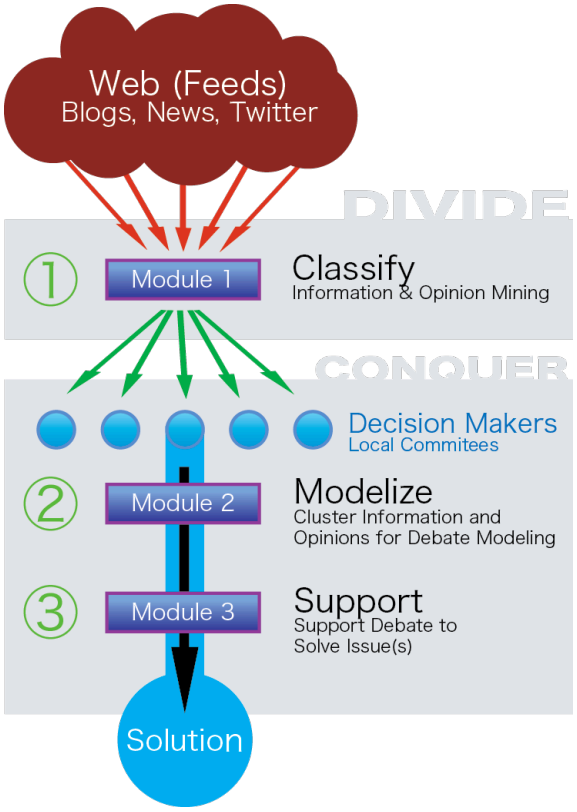

Figure 1.1: The three modules of research chained together.

time). This is where, in the case of application of our broader citizen participation platform project, the final aim is to motivate the end-user (the citizen) to take part in debate and state his opinions. Other applications can of course be derived, such as applications for business intelligence, etc.

1.3.2 Implementation Modules and Sophia Core

As a platform, Sophia is divided into several modules as well. The mining facilities process streaming APIs or crawl sites. The core processing facilities stem, filter and morphemize text and information that was mined through crawlers or streaming APIs. The learning and classification facilities cluster and classify information according to various axises (e.g temporal, regional, semantic). The user interfaces are the end-user tools that actually utilize the

platform directly.

We call Sophia Core the implementation module of the platform which is meant to pre-process (filter, stem, morphemize) all the contents that are mined from the various Web sources. Sophia Core is written in Java and classes can be easily incorporated to represent any kind of input and output data, so long as there is a corresponding implementation of interfaces DataReader <Input>, DataWriter <Output> and DataProcessor <Input,Output>. E.g, to output trend score files from sequences of JSON tweets, one needs only to implement a DataReader <Tweet>, DataWriter <ScoreMap>, DataProcessor <Tweet,ScoreMap>, the latter being a subclass of our TfIdf calculator where calculation is actually performed. Score maps are then outputted that can be directly utilized by the support and comprehension module.

Sophia Core also incorporates numerous interfaces that enables one to tune easily the processing of data, with adequate morphemizing based on MeCab (for Japanese texts), filtering and stemming, custom weighting, etc.

1.4 Mining and Learning

1.4.1 Mining

Mining can be done easily by passing JSON or XML-formatted data from a listener script written in every language (e.g a Perl script to mine the streaming API of Twitter) to Sophia Core and invoking the adequate processing class through the process call. Current applications include mining of sequence and log files, XML files, aggregated Tweets (e.g from the website Topsy) and streaming API tweets, as well as news articles. Processing of input data is multithreaded in a pool and therefore takes advantage of the hardware specifications, should it run only on one machine or on several nodes. Thus, mining in Sophia can be easily scaled according to the amount of input data to preprocess before any learning and classification on the data is to occur.

1.4.2 Learning and Classification

Learning is also scalable, first because of multithreading in the case of simple operations such as score calculation, but because of its facilities to convert

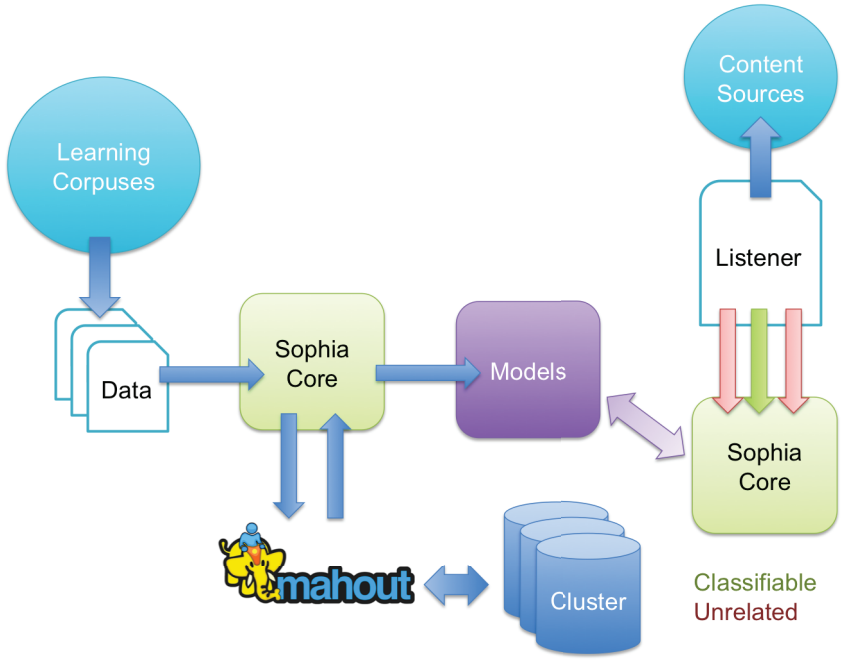

Figure 1.2: The learning architecture of Sophia.

Raw input data is processed, stemmed, morphemized, filtered and then analyzed by Sophia algorithms. In the case of big data, the learning data is converted to map-reduce usable input and passed to Hadoop facilities such as Apache Mahout, where some of our improved algorithms such as SAHB come at work.

any input data into adequate sequence files that can be used on a Hadoop map-reduce cluster for processing of big data, e.g with Apache Mahout, to which Sophia can connect directly to perform learning on the data with algorithms such as Transformed Weight-normalized Complementary Naive Bayes [Rennie et al., 2003] for classification, or hierarchical clustering of events. Learning facilities also incorporate means to easily test confidence of the models with precision and recalls, so that thresholds can be decided for the real-world classifiers.

The ability to make the input data map-reduce usable through the processing facilities of Sophia is important, but algorithms themselves need improvement, for example in the case of having to classify documents against a

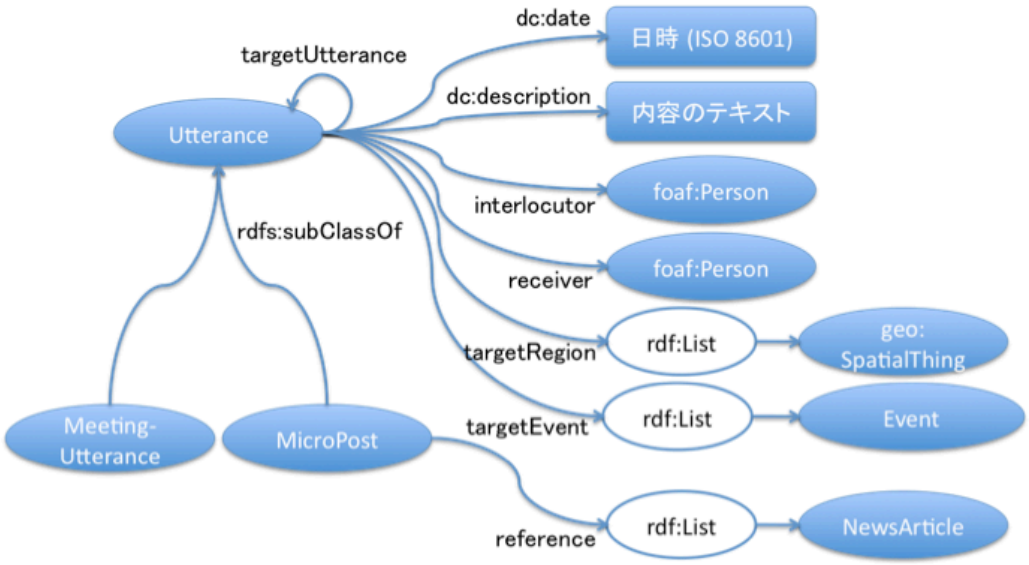

Figure 1.3: Socia: Ontology of debate.
Socia is a subset of Linked Open Data (LOD). In this figure, the sub-ontology of utterances in the opinion space, utilized by Sophia in its current case of application.

large number of classes (e.g geographical labels such as prefectures or cities). To this end, we developed Semantically-Aware Hierarchical Balancing as part of the classification research, an algorithm which outperforms straightforward approaches and Alpha Boost. SAHB has logarithmic time complexity when testing, being therefore very scalable for production, and performs with 94% accuracy on validation tests, against 70% for straightforward TWCNB.

1.5 Modeling and Structuring

Classified, clustered and scored information is then annotated according to a relevant ontology, which enables to structure data that was originally fuzzy. The data is modeled around different axises, e.g for debate support geographical/regional, semantic, and temporal. On Figure 1.3, we show an example of ontology, Socia, that is used to model and comprehend tweets as utterances.

We have developed Socia as a subset of Linked Open Data (LOD), based

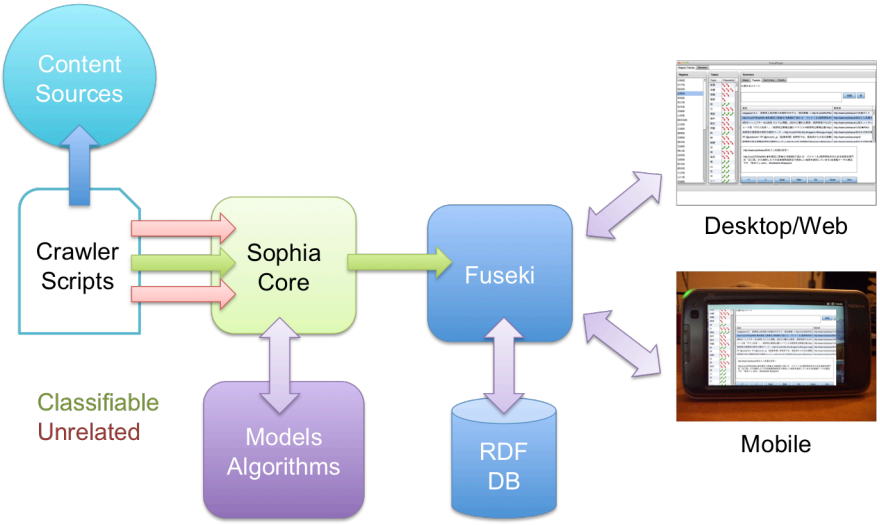

Figure 1.4: Support and comprehension by mining of various data sources. Data sources are processed through Sophia. Models trained with the learning and classification module are used to annotate data before it is passed to the graph RDF semantic database. The database is then utilized by end-user applications, either directly or through more restrictive APIs.

on various works such as the Open Meeting online debate system [Hurwitz and Mallery, 1995] and the Opinion Space [Faridani et al., 2010] online survey tool both developed for the U.S government, and it can be extended to meet more specific goals than our original project. It can for example be used to qualify discussion, utterances, etc, in situations such as business meetings. We use SPARQL as a query language for the Socia RDF, and Fuseki as a RDF database interface and server.

1.6 Support and Comprehension

As can be seen on Figure 1.4, structured data can be utilized directly through the RDF database, e.g for topic entities, pieces of contents such as microposts, or by intermediate Web APIs, e.g trend burst TfIdf scores. End-user interfaces are not limited to mobile and desktop

In our application to debate support, we have developed several sample

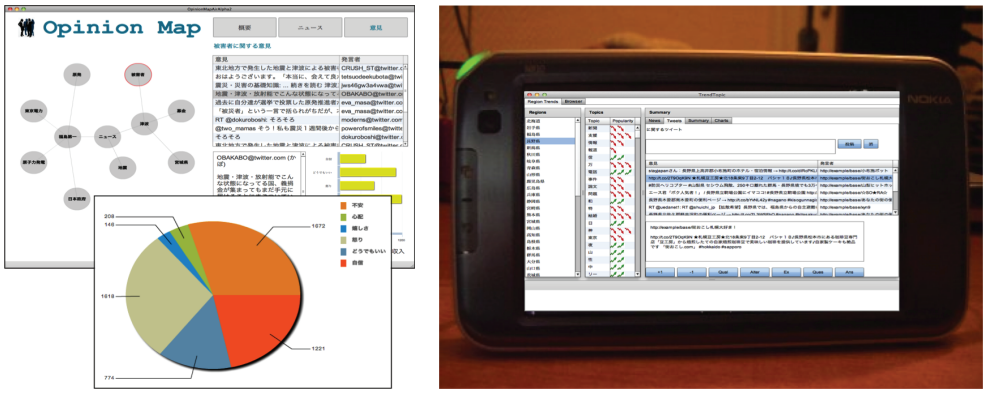

Figure 1.5: Two sample prototypes of support/comprehension of information utilizing Sophia.

On the left, OpinionMap, a topical graph explorer which clusters conversation around a long-term temporal and semantic axis. On the right, a Trending Topics user interface on Android, which clusters conversation by regions and short-term trends.

prototypes that utilize the Sophia modeling and structuring module directly through Web APIs. We present three representative prototypes: a related articles recommender, a topical graph browser, and a trending topics interface.

Related Articles Recommender

This related articles recommender, presented in a previous chapter [Swezey et al., 2011]), is a real-time dynamic recommender written in JavaScript which can run on any browser. The purpose of this application is to recommend relevant articles to the topic at hand on the currently being browsed Web page. For instance, if one is reading an article which relates to the Fukushima incident, it will recommend new articles on the subject. Differences with this prototype and other recommender systems resides in its real-time dynamism: even an old static Web page about the topic without any *related contents* cache is still valuable since it can use this client-side recommender to propose new articles on the subject, thus enabling the reader to develop background knowledge and knowledge about the topic's evolution.

Opinion Map

Whilst the recommender can be used to inform more the readers about background and evolution of ongoing topics when they browse a Web article, Opinion Map (on the left of Figure 1.5) is more oriented for concern assessment and opinion utterance in debate. The user navigates a graph of topics which are made of clustered events, then is able to browse a centralized history of news articles and utterances (tweets) related to it. The user can then participate in debate by uttering his own opinion on Twitter from the UI itself, and answering to interesting parts of the history. Currently in development for this prototype is a *mood temperature* facility, outputting the pie chart on Figure 1.5, which is a sentiment pie chart. A decision maker or any person interested in an ongoing social issues topic will be able to easily get the pulse of the crowd about a topic, based on sentiment analysis calculations done on the history of news and tweets by Sophia.

Trending Topics

The Trending Topics interface, as can be seen on the right of Figure 1.5, resembles Opinion Map in its finality, but it clusters topics around a much short-term temporal axis (the trend axis). Instead of a graph, it presents itself as a list of trends, the difference with common trend interfaces being that it can be more focused around a certain set of topics, e.g social issues in our broader project. Outside fuzzy discussion and information is clustered so that it becomes comprehensible and understandable. As for Opinion Map, it becomes easier to join a larger discussion. Trending Topics is actually intended to be a sub-application of Opinion Map.

Not only in citizen participation, but in more business-oriented applications such as financial sentiment analysis [Bollen et al., 2011], we believe that Sophia can be a reliable and perennial tool for data processing and intelligent support and comprehension. For example, the filtering and processing facilities can be used to trace and cluster opinions uttered about products that would not be routed through interfaces such as comments on retailer sites, and could constitute an interesting target as customer feedback since they would present different qualities, such as impulsivity in the opinion.

1.7 Conclusion and Future Work

In this chapter, we presented Sophia, a platform for intelligent processing, classification and structuring of fuzzy and unstructured data, so that it can be understood and used (e.g for debate) along various axises (e.g temporal, regional, semantic), by various distributions of population (e.g citizens and decision makers).

The platform is modular, makes use of open standards, and can be interconnected in different ways. The mining module can be extended easily to take new data sources in their original form as inputs, classification can be branched to various algorithms, modeling itself is a flexible ontology based on Linked Open Data, and support can be derived in various forms of interfaces. We gave examples of application in each research module that apply to the original aim of the system.

Although the platform is still young, it is currently being used in early production for its first case of application, concern assessment and debate support for social issues. Better tuning of facilities such as the learning algorithms in this precise case is needed to meet the objectives of the original project from which Sophia was born, but we believe the platform's architecture, as shown in this chapter, can help us achieve this goal.

Several works are currently in progress, which make extensive use of Sophia. An improved algorithm for geographical and topical classification, which makes use of Sophia and the open-source Apache Mahout for processing of big data. Another work concentrates on the evaluation of the interfaces for concern assessment and debate on various distributions of population, which is the concrete output of the overall project from which Sophia originated. When Sophia goes public and open-source, we will also provide a more developer-oriented technical work with several cases of application.

Chapter 2

Challenges in Real-World Contents Classification

2.1 Introduction

In order to be able to use the advantage of public corpora such as Wikipedia to address problems of classification by hierarchically structured topics with a large amount of classes, we propose an improvement of Naive Bayes based text classification algorithms which we call Semantically-Aware Hierarchical Balancing. SAHB addresses two issues in that specific use case with real-world applications, namely the large amount of topic labels to classify against, and the lack of balance in the hierarchy of the corpora. This meta-algorithm performs with better accuracy and log-time complexity than straightforward naive bayes text classification methods or specific document weighing techniques, whilst taking equivalent time to train, which makes it more efficient, and also scalable to process and classify big data.

The algorithm described in this chapter originally stemmed from a broader research project, of which the aim is to provide an intelligent platform for assessing concern from online users and increase public involvement of the Japanese population about social issues. Such a platform's end-user perspective consists in various interfaces which present coherent data out of fuzzy and random Web data which is neither annotated, nor structured or consistent. To this end, the data has to be modeled adequately, which begins with labeling the various contents that we need to use (articles, tweets, etc).

In this chapter, we focus mainly on our general approach to classify con-

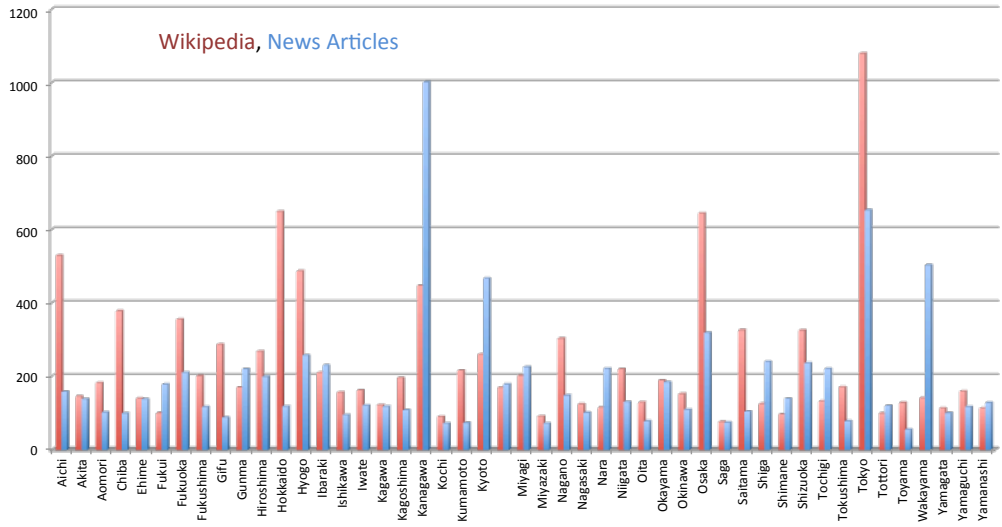

Figure 2.1: Imbalance in the distribution of regional classes

tents more efficiently and quickly according to geography by using open crowdsourced corpora that can be found online and out of which datasets can be easily built and rebuilt. We discuss the research background, then present the approach and early validation experiments that show promising results. We then discuss further research directions and conclude.

2.2 Related Works

2.2.1 The Challenge of Imbalance

In order to use advantages of crowdsourced corpora, one has to take account of several challenges. The corpus can be highly imbalanced, as such is the case for the Japanese Wikipedia corpus once labeled with confidence to geography: for each class c of a possible set of classes C such as topical classes or geographical classes, some classes contain a much larger number of documents than the others (Fig. 2.1). There can also be a lack of data that can be neglected in the case of the biggest Wikipedia corpora, but not in others. In smaller Wikipedia corpora, where the language is not popular, the advantages stated above may not apply.

As a straightforward approach and control experiment, we use a Transformed Weight-normalized Complementary Naive Bayes classifier (TWCNB) [Rennie et al., 2003] which has a natural better efficiency than Naive Bayes (NB), as well as a short training time compared to SVM, while retaining close performance. In our preliminary experiments for geographical classification, TWCNB shows better efficiency than NB, so it will be the straightforward algorithm that we use in our sample problem. Details on how TWCNB functions can be found in [Rennie et al., 2003].

As shown in our experiments in Section 2.4, TWCNB cannot be used straightforwardly in the case of geographical classification using the Japanese Wikipedia corpus. The bias created by class imbalance being too high, it needs to be supervised by a meta-algorithm. Such meta-algorithms for improving classifiers already exist, like AdaBoost. However, if the amount of classes is to become relatively large when compared to classical examples of multi-class Bayesian classification problems (20NewsGroups, Reuteurs, etc), even with AdaBoost improvement [Freund and Schapire, 1999] the testing algorithm has a linear time complexity of $O(n)$. Hence, any straightforward testing algorithm becomes non-optimal in the case of nation-scale geographical classification of text, when the order of magnitude of the number of classes is 100 times higher.

Moreover, when it comes to accuracy, even with AdaBoost improvement, it is intuitive to take advantage of the tree-structure of geographical topic nodes in order to improve efficiency. Hierarchical approaches are more intuitive and appropriate in such cases, and exist for text classification with SVMs [Dumais and Chen, 2000] as well as for known problems [Toutanova et al., 2001]. Here, we approach the problem of crowdsourced corpora and use clustering of existing hierarchy nodes together to counter bias when the hierarchy of classes is known. Oversampling and undersampling [Japkowicz and Stephen, 2002] also are known methods for alleviating class imbalance problems, however they can also affect the class prior calculation and other variables in Bayesian classification learning.

Other surveyed works such as [Lee et al., 2008], [Demichelis et al., 2006], [Estabrooks et al., 2004], [Liu et al., 2009], [Japkowicz and Stephen, 2002] show either no application to text classification nor methods based on trees along with clustering of classes. For example, SMOTE [Chawla et al., 2002] is an effective variant of mixed undersampling and oversampling with synthetic minority class examples, however it does not take advantage of an existing hierarchy of classes and does not address a problem of a very large and

imbalanced multinomial distribution of class labels.

2.2.2 Recent Works

[Lieberman and Samet, 2012] propose adaptive context features for toponym resolution that enable them to geo-tag news more accurately. However, they address a different problem in the sense that they want to predict longitudes and latitudes given toponyms retrieved from text. In our context, toponyms and their locations are already known and what we address a document classification to the right toponym.

[Lichtenwalter and Chawla, 2010] address the problem of distributional drift in data streams over time, i.e. how class imbalance varies over time in the streams and its effects on classification. This is relevant work for us because the degree of imbalance in streaming news indeed varies over short time windows. The main limitation of this work is that it addresses only binomial distributions, with experiments on only one text dataset. Addressing the same problem, [Hoens and Chawla, 2012] propose a method for dealing with concept drift in data streams which also suffer from class imbalance, with the same limitation in our context.

[Yen and Lee, 2006] devise an approach that is similar to our own in that it uses clusters, but it is a cluster-based under-sampling approach that does not consider multinomial distributions nor an existing hierarchy of classes. Still in the context of binomial distributions only, [Wang et al., 2013] devise the sample cutting method for dealing with high density neighborhoods for Support Vector Machines with imbalanced data.

[Zhou et al., 2012] propose compressed labeling to tackle the imbalance, dependence and high dimensionality of the label matrices in a multi-label classification problem. They address a variety of problems using Support Vector Machine classification. They also analyze class dependency which is of interest to us. However in our context we use Bayesian classification for its speed and accuracy on text classification problems. Also, the class dependency can be observed from the hierarchy. Likewise, [Tahir et al., 2012] propose a generalist ensemble method but in datasets where there is no hierarchy and no large number of classes.

Figure 2.2: Input classes with no structure.

2.3 System and Algorithm

We propose our method, which we call Semantically-Aware Hierarchical Balancing (SAHB). SAHB is a variant of resampling which creates upper-level class nodes by clustering semantically close sets of documents from lower-layer classes in a higher-layer classification problem and keeps the variance in the class size at its lowest possible, hence hierarchically balanced.

In training, the meta-algorithm takes all the predefined classes and groups semantically close classes under a same parent node. In testing, each node is a class (label) and a test for classification is applied at each depth level of the tree. This method is *semantically aware* because child nodes are clustered under a parent node which has semantical relation to it. It is *hierarchically balanced* because it takes account of the overall amount of classes for optimal classification. Bias in the class size affects the calculation of TfIdf weights, thus we cluster classes in a tree hierarchy to counter this bias.

We initially begin with a set of class label nodes that are not structured (Fig. 2.2). In our classification example, at depth level n of the classification tree, the set of classes C_n is made of Tokyo, which holds a very high number of documents, and other prefectures from the Kanto region.

$$C_n = \{Tokyo, Chiba, Ibaraki, Saitama, ...\}$$

Because of high bias, a lot of documents are likely to be misclassified in Tokyo when testing. We indeed observed a quite heavy column for Tokyo

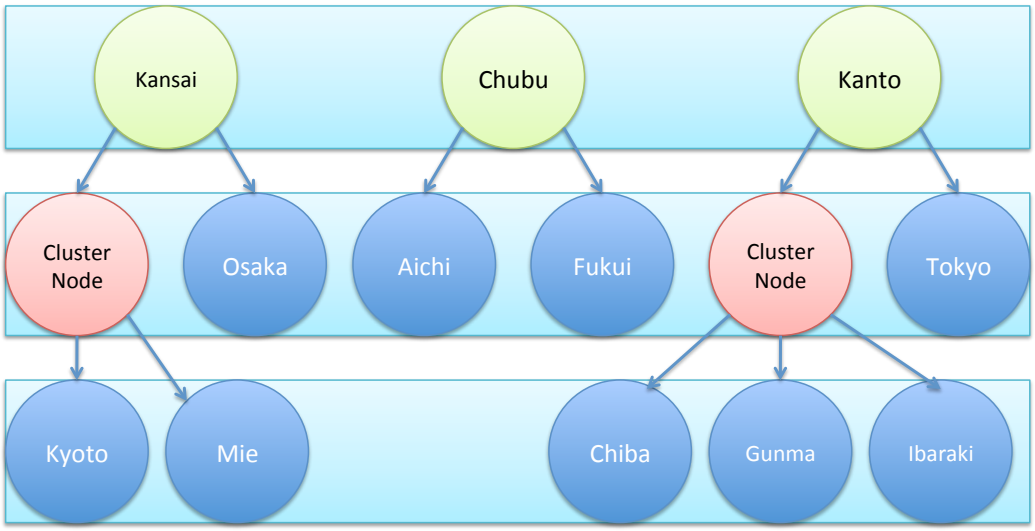

Figure 2.3: Output class label nodes structured against imbalance.

in the confusion matrix when testing the classifier *flatly* (as opposed to hierarchically), i.e when SAHB was not applied. Hence, it is better to cluster classes geographically close to Tokyo against the Tokyo class itself to reduce the variance, with a set such as

$$C_n = \{Tokyo, \{Chiba, Ibaraki, Saitama, ...\}\}$$

The set at depth level $n+1$, if the document was not classified in Tokyo, is then

$$C_{n+1} = \{Chiba, Ibaraki, Saitama, ...\}$$

. This gives us a set of structured class label nodes as seen in Fig. 2.3.

As a testing algorithm, SAHB also shows better time complexity than straightforward TWCNB (or AdaBoost-improved TWCNB) does. The time complexity is

$$\beta * log_\beta(n)$$

where β is the average branching factor of the class tree, which is considered stable and not a function of depth, and n the number of leaf classes to be tested against, which corresponds to multiplying β testing iterations by the β-ary search through the tree. Hence, it corresponds to a

$$O(log(n))$$

which is very suitable if n grows very large, for example if we need to classify not only against prefectures, but also cities and even districts.

2.4 Evaluation

In the following experiments, we describe the algorithm that we used with:

(Name of Meta-Algorithm)-(Depth of Class Tree)LH-(Name of Sub-Algorithm)

LH stands for *Layer Hierarchical*. We call *Flat* the case where there is no depth in the class tree, i.e when classification is done by testing the document against all label nodes at the same time. When no meta-algorithm is described, we simply used the natural hierarchy given by Wikipedia, without any clustering against bias.

The dataset for training and testing was built from the Japanese Wikipedia corpus, by adding each article as a document for the prefecture, city, or district class it relates to. The Japanese Wikipedia constitutes a fairly rich corpus, but it also has the flaw of high imbalance, with majority classes such as the Tokyo label.

2.4.1 Validation Experiments

First, we conducted straightforward experiments to assess the existence of a problem related to class imbalance. Results are shown in Table 2.2.

In a per-city hierarchical classification experiment, we used the natural hierarchy given by Wikipedia, without any clustering against bias. This was also done in a previous chapter [Swezey et al., 2011] in which classification still needed improvement. For 2LH-TWCNB on Japanese cities, we obtained 66.602% accuracy with 38ms of average time to classify an instance, against respectively 42.086% accuracy and 522ms processing time in Flat-TWCNB.

As can be expected, a hierarchical classifier ensemble method outperforms the flat classifier in validation, notably in processing time performance, although its accuracy needs improvements. This is also important when determining confidence faster in the real-world testing of contents: if there is no confidence for the classifier in choosing between classes over the initial cluster of classes (regions which are clusters of prefectures), it is very unlikely that it will gain more confidence at a lower level of the class tree. Thus it also

Table 2.1: Cluster nodes built with SAHB in the prefecture experiment.

First-level Cluster Node	Child Nodes
Hokkaido-Tohoku	Hokkaido, Tohoku
Kanto	Tokyo, KantoSub
Chubu	Aichi, Fukui, Gifu, Ishikawa, Nagano, Nigata, Shizuoka, Toyama, Yamanashi
Kansai	Osaka, KansaiSub
Chugoku-Shikoku-Kyushu	Hiroshima, Okayama, Shimane, Tottori, Yamaguchi, Ehime, Kagawa, Kochi, Tokushima, Fukuoka, Kagoshima, Kumamoto, Miyazaki, Nagasaki, Oita, Saga, Okinawa
Second-level Cluster Node	Child Nodes
Tohoku	Akita, Aomori, Fukushima, Iwate, Miyagi, Yamagata
KantoSub	Chiba, Gunma, Ibaraki, Kanagawa, Saitama, Tochigi
KansaiSub	Hyogo, Kyoto, Mie, Nara, Shiga, Wakayama
Leaf Nodes	All 47 prefectures

improves time complexity, speed and accuracy when testing confidence for analysis of the geographic property of contents, namely testing if contents are related to geography or not at all. Therefore, the discarding of numerous contents which are not geographically-relevant will also be significantly faster.

Finally, in a per-prefecture classification experiment, we use SAHB to cluster semantically-close label classes in cluster nodes, which are region nodes created for reducing bias. Nodes are built in the Japanese Wikipedia geographical classification example as described in Table 2.1. To better visualize the layers and restructured class label nodes, the reader can also refer to Fig. 2.4 and Fig. 2.5 which graphically show how geographical class labels are restructured to counter balance issues.

We witness, as for hierarchical classification, a gain in processing time, but here also more notably a significant gain in accuracy with SAHB on Table 2.4 when compared to simple hierarchical classification on Table 2.3.

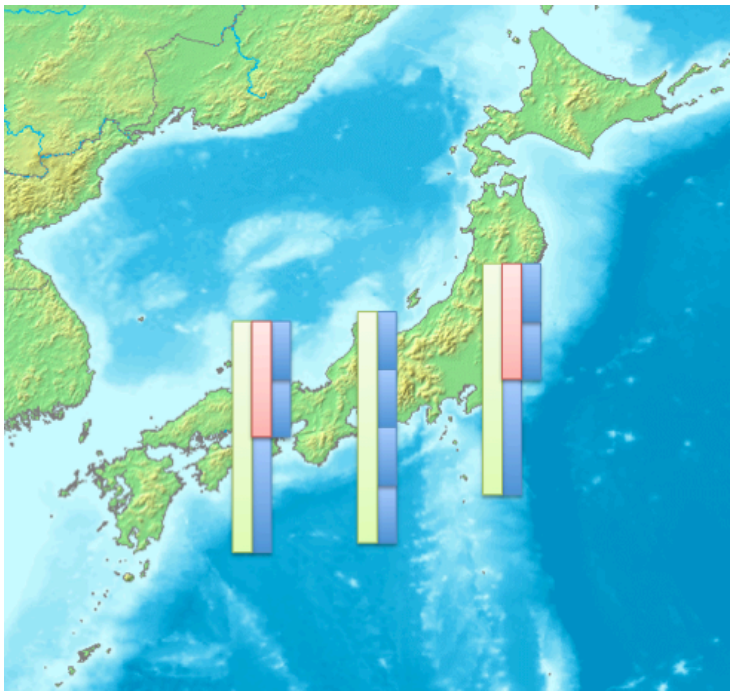

Figure 2.4: Example of input dataset showing class sizes

2.5 Proposed Automation

We propose an automation of SAHB as follows:

- Let $|C|$ be the number of class labels
- Input: Original training set
- Output: Training set with minimal class imbalance
- Algorithm
  - Group into L layers using original hierarchy e.g (1) regions, (2) prefectures, (3) cities
  - For each layer:
    - Isolate the majority classes with threshold

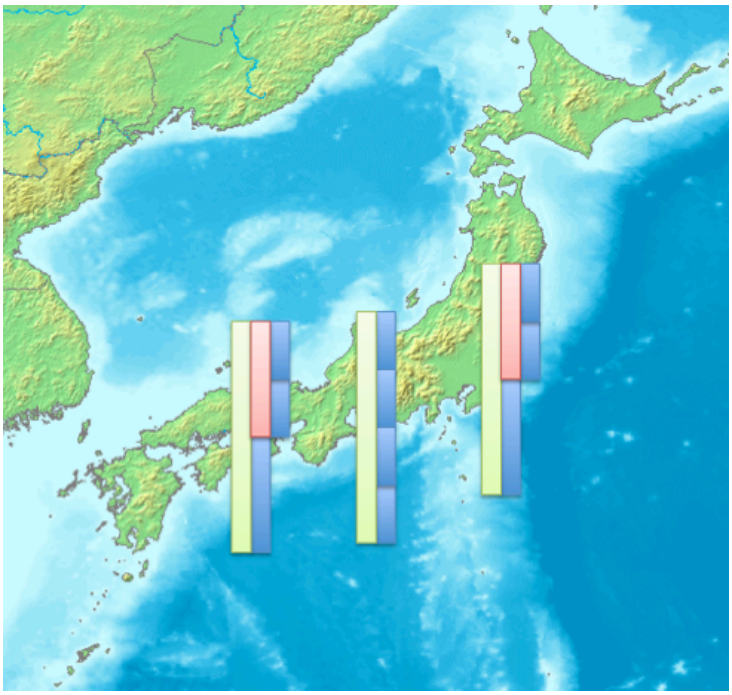

Figure 2.5: Restructured output dataset

* For $k \in [2, |C_{currentlayer} \setminus \{majorityclasses\}|]$
* Run k-means clustering on geographical and/or semantic dimensions
* Obtain clusters of class labels as new group classes
* Retain σ_k^2, the variance in class size obtained
* Obtain ideal set such that: $k_{ideal} = argmin_{k \in K} \sigma_k^2$

2.6 Conclusion and Future Work

In this chapter, we presented SAHB, an improvement to straightforward Naive Bayes text classification methods, which is a meta-algorithm that can be trained in an equivalent time to the subsequent classification algorithms it uses, and takes advantage of a known hierarchy of classes as well as semantic

Table 2.2: Performance of Flat TWCNB on Prefectures

Classes	47 prefectures
Correctly Classified Instances	69944 (70.250%)
Incorrectly Classified Instances	35074 (29.750%)
Total Classified Instances	11173
Avg. Time to Classify one Instance	12,4 ms

Table 2.3: Performance of 3LH-TWCNB on Prefectures

Training Classes	55 classes: 47 prefectures child nodes 8 region cluster nodes
Test Classes	47 prefectures
Correctly Classified Instances	10565 (75.360%)
Incorrectly Classified Instances	608 (24.640%)
Total Classified Instances	11173
Avg. Time to Classify one Instance	3,9 ms

proximity. The idea for the algorithm stemmed from a broader project in which one of the research milestones is to classify by using the advantages of online open crowdsourced corpora.

The algorithm needs further generalization and comparisons with other methods, and tests on other datasets, as it is limited in this case to a sample problem. Other cases include but are not limited to: having topic nodes belonging to multiple parents, bias such as equilibrium by clustering parent nodes together cannot be reached, which may require mixing with other techniques e.g oversampling. Finally, it also needs to be tested up to per-city classification in our example, although it is recursively intuitive that accuracy will still be better than straightforward use of TWCNB.

Table 2.4: Performance of SAHB-3LH-TWCNB on Prefectures

Training Classes	55 classes: 47 prefectures child nodes 8 region cluster nodes
Test Classes	47 prefectures
Correctly Classified Instances	10565 (94.558%)
Incorrectly Classified Instances	608 (5.442%)
Total Classified Instances	11173
Avg. Time to Classify one Instance	3,3 ms

Chapter 3

Automatic Detection of News Articles of Interest to Regional Communities

3.1 Introduction

In this chapter, 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: Naive Bayes, Complementary Naive 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.

We developed an e-Participation Web platform, O_2, for regional communities. The platform aims at supporting citizen e-Participation in ongoing regional debates by gathering and openly publishing news and opinions. Structuring citizensçłűï£¡awareness of regional issues and sharing structured data are two requirements in conducting productive discussions about vari-

ous issues. O_2 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 chapter, 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 chapter is organized as follows. In section 2, we present related works. In section 3, we introduce our systemçłűåĘś 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 chapter in section 5.

3.2 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 chapter, this is the 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 Naive Bayes (NB) [Zhang, 2004] and Transformed Weight-normalized Complementary Naive Bayes classification (TWCNB) as devised in [Rennie et al., 2003]. 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 [Japkowicz and Stephen, 2002].

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 [Swezey et al., 2012b]), 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 [Freund and Schapire, 1999] can lead to relatively longer training time over several iterations. Oversampling and undersampling [Japkowicz and Stephen, 2002] 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 [Hastie et al., 2003] and latent Dirichlet allocation [Blei et al., 2003] 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.

[Anastácio et al., 2009] devise a method for the task of classifying local (related to geography) against global. We address this problem in Chapter 3 by using a confidence threshold in the classification task. The threshold is determined using a noise corpus and does not require training of another classifier. Our method proves very effective and accurate.

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.

3.3 Proposed Approach

3.3.1 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 chapter describes the filtering and classification tasks. Clustering is the subject of another chapter [Hirata et al., 2012].

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.

3.3.2 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.

3.3.3 Classification by Region

Recall the overall process of Sophia on Fig. 3.1.

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 $TF \times IDF(t, d, D) = tf(t, d) \times idf(t, D)$, also known as the TF*IDF metric, defined by:

$$\text{tfidf}(t, \mathbf{d}, D) = \text{tf}(t, \mathbf{d}) \times \text{idf}(t, D) \tag{3.1}$$

with

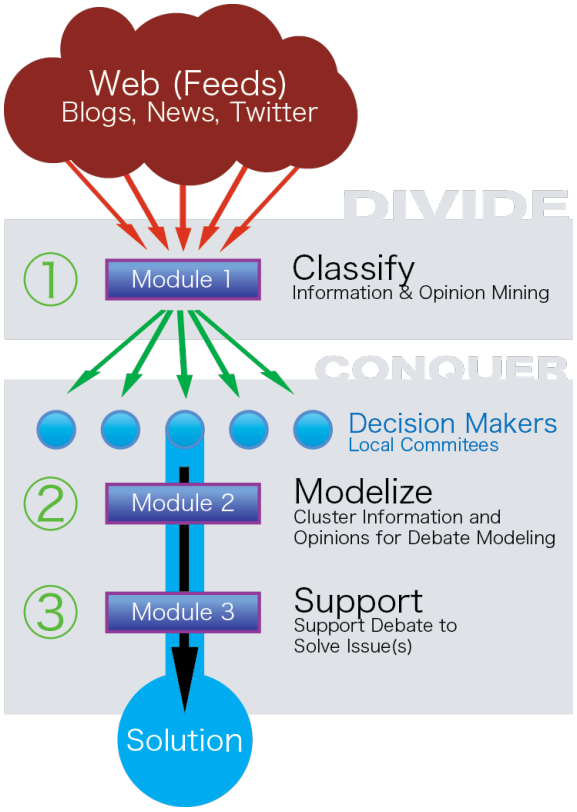

Figure 3.1: Overall process conducted by Sophia.

$$\mathrm{tf}(t, \mathbf{d}) = \mathbf{d} \cdot \mathbf{t}$$

$$\mathrm{idf}(t, D) = \log \frac{|D|}{1 + |\{\mathbf{d} \in D : \mathbf{d} \cdot \mathbf{t} > 0\}|}$$

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:

$$\mathrm{P}(c|\mathbf{d}) = \frac{P(\mathbf{d}|c) \times P(c)}{P(\mathbf{d})} \tag{3.2}$$

A variant of Naive Bayes text classification is the Transformed Weight-normalized Complementary Naive Bayes algorithm [Rennie et al., 2003], where the score for classifying a document d into a class C is calculated as:

$$\hat{\mathrm{w}}(c|\mathbf{d}) = \log P(c|\mathbf{d}) - \sum_{t} \mathrm{tf}(t,\mathbf{d}) log(\frac{1+\sum_{k=1}^{|C|}\mathrm{tf}(t,c_k)}{N+\sum_{k=1}^{|C|}\sum_{x=1}^{N}\mathrm{tf}(x,c_k)}) \tag{3.3}$$

where N denotes the size of the vocabulary. Further improvements of the classification algorithm are the subject of the previous chapter.

3.3.4 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 ï¿Őï¡ş is determined by:

$$\gamma(\mathbf{d}) = \hat{\mathrm{w}}(c_1|\mathbf{d}) - \hat{\mathrm{w}}(c_2|\mathbf{d}) \tag{3.4}$$

with:

$$\mathrm{c}_1 = \arg\max_{c\in C} \hat{\mathrm{w}}(c|\mathbf{d}), \mathrm{c}_2 = \arg\max_{c\in C\setminus\{c_1\}} \hat{\mathrm{w}}(c|\mathbf{d})$$

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.

3.3.5 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. 3.2.

```
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.
```

Figure 3.2: Histogram of Class Size by Regional Class Label

3.4 Experimental Results

3.4.1 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. 3.3 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.

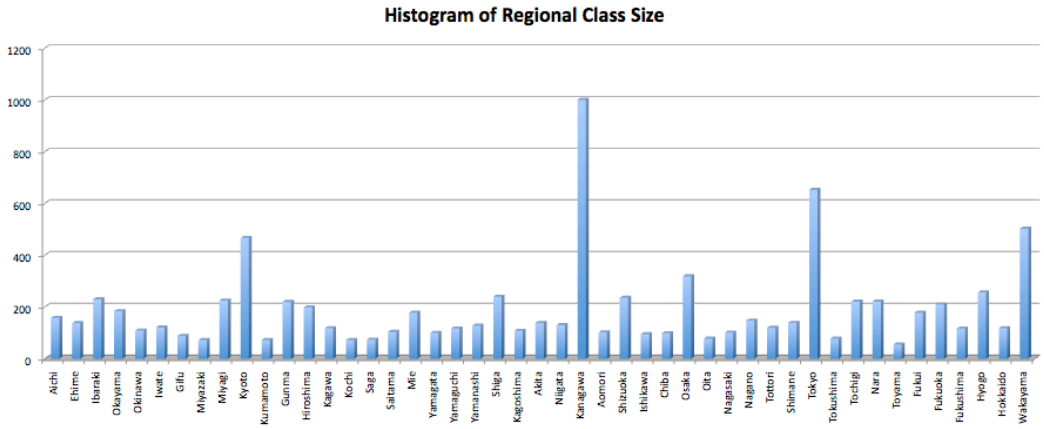

Figure 3.3: Histogram of Class Size by Regional Class Label

3.4.2 Classification Experiment

Method. We test two Bayesian text classification algorithms, Naive Bayes (abbreviated as NB) and Transformed Weight-normalized Complementary Naive 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 [Hastie et al., 2003]. 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,

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"

Table 3.1: Performance in closed tests with oversampling and gramization.

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"

Table 3.2: Performance in 10-fold tests with oversampling and gramization.

the training 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:

$$P = \sum_{c \in C} \left(\frac{TP}{TP + FP} \right)_c = \frac{\sum_{c \in C} TP_c}{D_{training}} \tag{3.5}$$

where TP and 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 3.1. 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.

3.4.3 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 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:

$$\gamma_i = \frac{i}{\tau} \tag{3.6}$$

with:

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

where 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 γ set to 0.03. The precision P and recall R of the system are given by the formulae below.

$$P = \frac{\sum_{c \in C} TP_c}{\sum_{c \in C} (TP + FP)_c + FP_c} \tag{3.7}$$

$$R = \frac{\sum_{c \in C} TP_c}{\sum_{c \in C} (TP + FN)_c + FN_c} \tag{3.8}$$

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

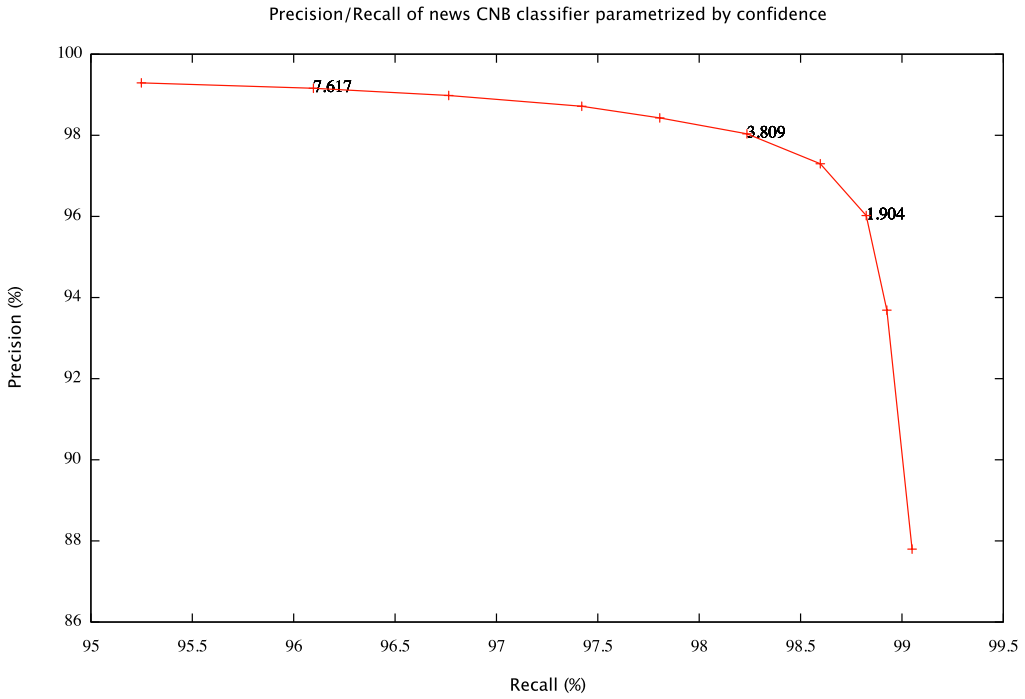

Figure 3.4: Performance in 10-fold tests with oversampling and gramization.

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 depending on γ. The F1 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:

$$F_1 = 2\frac{PR}{P+R} \tag{3.9}$$

Results. Figure 3.4 shows precision P in function of recall R according to the confidence threshold parameter γ. Measures including the F1 score

Table 3.3: F1 score, Precision, Recall, Threshold, Domain-FP, Domain-FN

F_1	P	R	γ	FP_C	FN_C
93.08	87.79	99.04	0	1133	0
96.23	93.68	98.92	0.95	521	27
97.4	96.02	98.82	1.9	301	43
97.94	97.29	98.59	2.85	191	73
98.13	98.03	98.23	3.8	125	107
98.11	98.42	97.8	4.76	92	148
98.06	98.71	97.42	5.71	73	189
97.86	98.98	96.76	6.66	51	249
97.6	99.15	96.09	7.61	38	311
97.22	99.29	95.24	8.56	31	391

are given on Table 3.3. 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.

3.5 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 [Japkowicz and Stephen, 2002]. 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

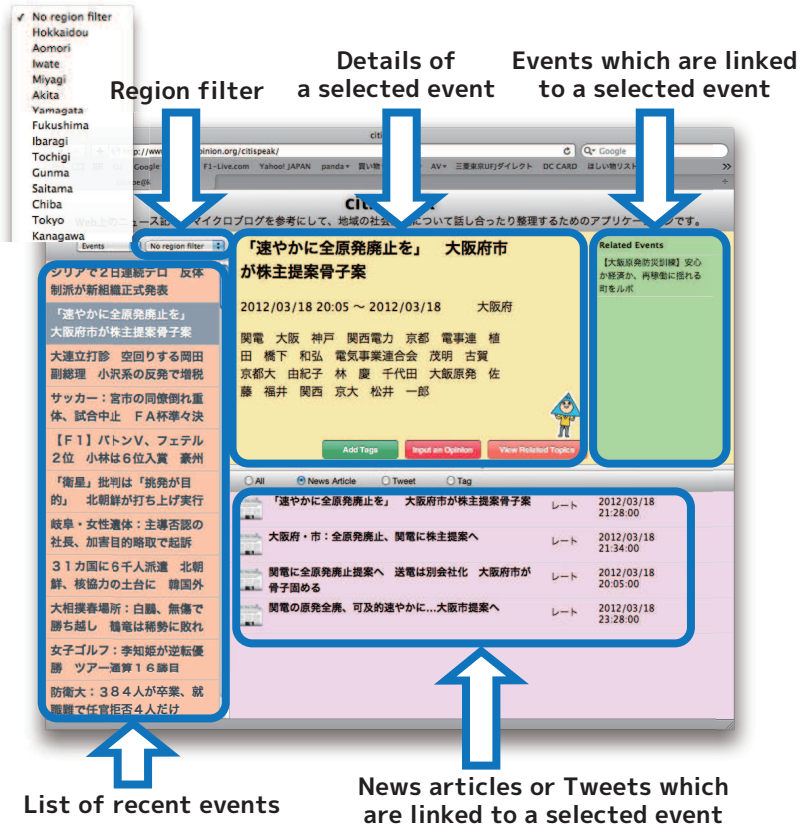

Figure 3.5: Screenshot of citispe@k.

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.

3.6 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. 3.5 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 [Swezey et al., 2012a].

More details are given on the interface in the next chapter.

3.7 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: Naive Bayes, Complementary Naive 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.

Chapter 4

Information Exchange and Input Probing Through Debate Support

4.1 Introduction

In this chapter, we present an e-Participation Web system for various platforms and mobile phones, based on linked open data, classification and clustering. The system aims at supporting citizen e-Participation in ongoing regional debates by gathering and openly publishing news and opinions from the Web for easy comprehension and commenting. Our study helps us define relevant evaluation criteria for an adequate citizen discussion system in the new context of open government, the Web, and mobile computing. We present the system, O2, and its application citispe@k, as well as its underlying components: ontology structure, classification and clustering. We then conduct a comparison with existing systems and find that our system is a better approach for efficient citizen e-participation when compared to current existing systems.

We developed an e-Participation Web platform, O_2, 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 the regional issues and sharing the structured data are two requirements to conduct productive discussions about various issues. O_2 consists of three tools: Sophia, SOCIA, and citispe@k. Sophia is a mining and intelligent pre-processing platform which classifies and clusters news articles and tweets. SOCIA is a data set and its ontology for debate,

based on Linked Open Data (LOD), 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 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 LOD-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 functionality.

The system first collects news articles and microblog posts (in this work, tweets) 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. Next, the system extracts target events from the news articles and microblogs, and links them using the ontology. Citizens can then add further links to events, news articles and microblogs, by creating relevant topics and debate about them by inputting their opinions, polling, or sharing further resources. Those resources and new links are also incorporated in the data set, as are the opinions and the discussion. This creates a virtuous circle where the intelligent platform, by creating understandable and relevant discussion seeds, involves citizens in e-Participation. The citizens add further data to the data set, making it grow over time, and this data can be used as input again (e.g for training better learning models or developing better ontologies).

The rest of this chapter is organized as follows. In section 4.2, we briefly describe related works. The details of the data set SOCIA are introduced in section 4.3. The application of SOCIA is described in section 4.4. We compare our system and current systems which can be used for debates in section 4.5 to insist superiority of our system. And in section 4.6, we summarize our contributions and conclude the chapter.

4.2 Related Works

In this section, we present related works to introduce the context in which we will present and evaluate our system for regional citizen participation involvement.

4.2.1 e-Government and e-Participation

e-Government, as defined in [for Public Administration et al., 2010], consists in the employment of the Internet and the world-wide-web for delivering government information and services to the citizens. Mainly, it refers to the use of new technologies by governments to reach and interact with citizens.

Actual use of these tools to extend the scope of e-Government by including citizen engagement and participation in governance, i.e use of information and communication technologies to achieve better governance, is referred to as e-Governance. Finally, use of e-Participation tools in the decision-making process of democratic government organizations is referred to as e-Democracy [Macintosh, 2004].

e-Participation is the use of information and communication technologies to broaden and deepen political participation by enabling citizens to connect with one another and with their elected representative [Macintosh, 2004].

4.2.2 e-Participation Tools and Technologies

Wimmer et al. [Wimmer, 2007], with Macintosh et al. [Macintosh et al., 2005], group the existing e-Participation tools into three main clusters: core e-participation tools, tools used in e-participation but not specific to it, and basic tools to support e-participation.

Core e-participation tools are actually tools that use a goal-specific definition relatively to e-government: e-Participation chatrooms, e-Participation message boards, decision-making games, virtual communities, online surgeries, e-Panels, e-Petitioning, e-Deliberative Polling, e-Consultation, e-Voting, suggestion tools for planning procedures. The first four are not specific to e-Participation but in a goal-oriented domain they could be regrouped as e-Participation discussion and involvement tools, which is what O_2 is. As well, tools used in e-Participation are technology-specific tools, namely webcasts, podcasts, wikis, blogs, quick polls, surveys, GIS-tools. Tools of support are

actually legacy technology of the environment of e-Participation, used by citizens on their own account but not provided by the system: search engines, alert services, newsletters, FAQs, portals, groupware tools.

This study helps us give a more precise context to O_2, which is an e-Participation discussion and involvement tool (goal) with an innovative technological approach relying on structuring, modeling and presentation of data (technology) with support by using outside data. In the later comparison with other tools that we conduct in Section 4.5, this helps us define relevant comparison criteria for O_2.

4.2.3 Open Meeting and Further Initiatives

Open Meeting constitutes a special case of an e-Participation support system. Developed for vice president Al Gore's Open Meeting project on National Performance Review, the Open Meeting system [Hurwitz and Mallery, 1995] made use of knowledge representation, hypertext grammar, and rules for commenting. The system was meant to empower users with the ability to conduct policy conversations without any agency boundaries. The research conducted for the development of the system first identified the interactions needed for productive discussion in a large group. For example, specialists could only access texts relevant from their interests and link comments on those texts. Open Meeting was built around a very precise structure of debate, with a hypertext grammar that made it possible, whereas systems such as newsgroups and traditional bulletin boards were not suited for such participative debate. In the context of regional productive citizen discussion, we thus consider Open Meeting as an ground-laying and inspiring work for SOCIA, since it shows that relevant discussion structuring through technological innovation leads to effective citizen collaboration and e-Participation.

Next in line is the MIT Deliberatorium [Klein, 2011]. Deliberatorium is a system to enable better collaborative deliberation. It works through systematic exploration evaluation and convergence on solution ideas, including stakeholders and experts. Different from systems such as message boards where interaction is time-centric, it aims at solving the problems of scattering of points on a topic, balkanism (the clustering of like-minded users on threads), and the soapbox problem where the last to speak is the last to be heard, and small voices tend to get left out. To address this issue of noise, Deliberatorium relies on argument mapping, in a manner very similar to Open Meeting. This results in no scattering, no soapbox problem,

and bias towards well-founded arguments. While Deliberatorium structures discussion over long-time spanning topics, O_2 chooses a more event-centric approach. The reason is that events tend to create a flock of individuals, described as swarming [Della Porta and Diani, 2006], as a reaction to punctual events that directly concerns them. When repeated, ad-hoc movements born from swarming can engage in participation on a more long-term basis. Past research was also conducted on the use of swarming towards service issues in hope to create a participation habit [Arrivals et al., 2005]. Since O_2 also aims at structuring debate, it adds structure to the discussion through the use of various tags (see Sec 4.4).

Finally, another work of interest is Cohere [Liddo and Shum, 2010]. Cohere is a social, semantic web application described as a working prototype based on the rationale of contested collective intelligence. Cohere focuses on sensemaking, and its is to help users make sense out of data, by connecting ideas and annotating Web pages, then linking them through an ontology. According to its own definition, it sits at the intersection of Web annotation, social bookmarking, and mindmapping. Cohere differs from our project in the sense that it is strongly based on idea connection and knowledge mapping and sharing, whereas O_2 focuses on regional citizen participation using current events as discussion seeds.

4.3 Platform:O_2/SOCIA

4.3.1 O_2/Sophia/SOCIA

O_2 is a Web platform for citizen participation in debates about regional issues. O_2 is an abbreviation for Open Opinion. Fig. 4.1 shows the outline of O_2 platform. O_2 has three stages. In stage (1), the mining and pre-processing system Sophia crawls the Web and gathers informations such as news articles or microblogs, which can be used for debates from the Web. In stage (2), the system tries geographical classification and event clustering to structure the gathered data. Relevant data of interest is then structured and stored in the data set SOCIA according to the SOCIA ontology, as openly published Linked Open Data. Stage (3) is the application of SOCIA's purpose, debate support. Several applications can actually branch to Sophia/SOCIA, not only debate supporting systems but also an e-Meeting system we developed. In this chapter, we focus on the e-Participation system for supporting debates

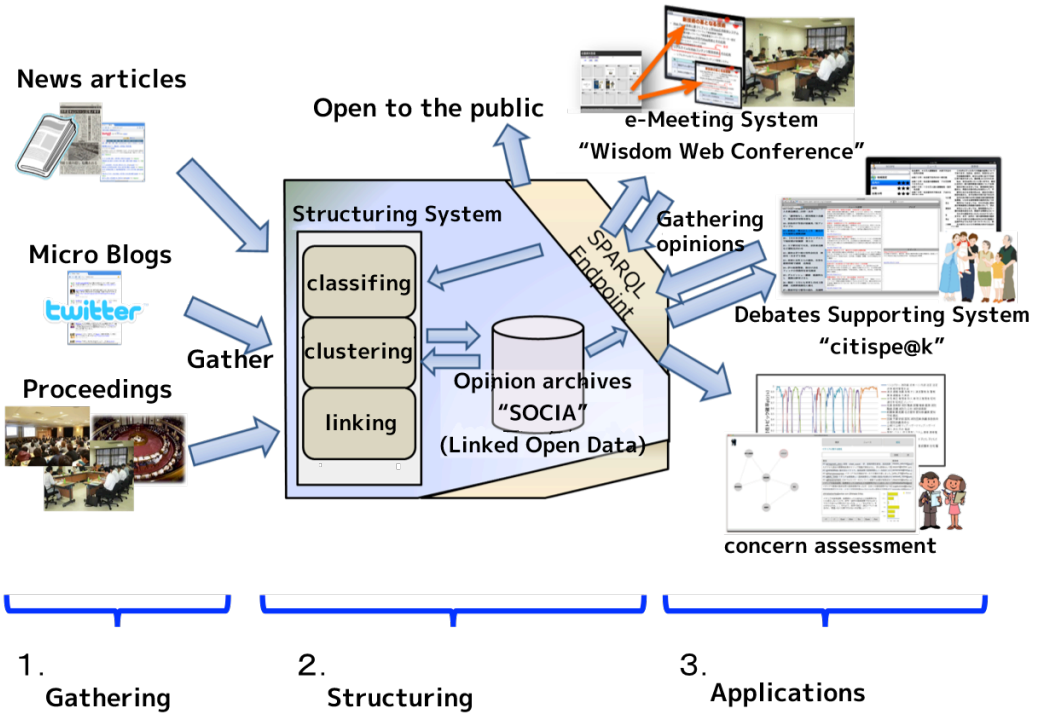

Figure 4.1: Outline of O_2

about regional issues.

4.3.2 Classification by Region

After mining, we perform classification of news articles and tweets by geography (against the 47 prefectures of Japan). To this end, we use Naive Bayes text classification. The metric chosen is the Tf-Idf metric as follows:

$$\text{tfidf}(t, \mathbf{d}, D) = \text{tf}(t, \mathbf{d}) \times \text{idf}(t, D) \tag{4.1}$$

with

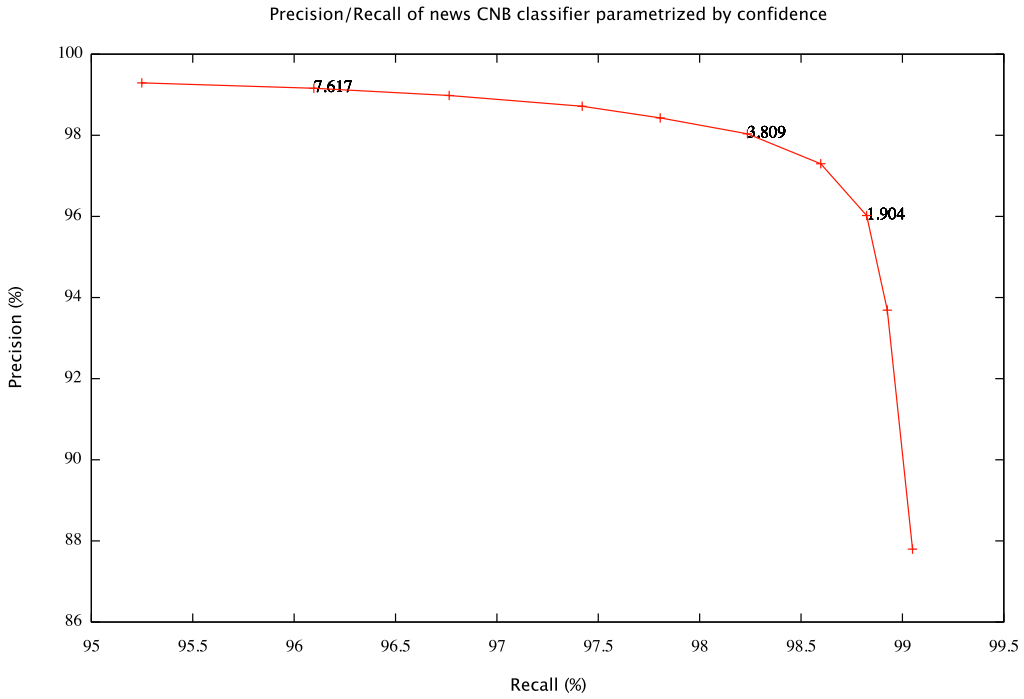

Figure 4.2: Precision/Recall parametrized by confidence in classification of news articles

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

where $|D|$ is the cardinality of the corpus of documents, and $|\{\mathbf{d} \in D : \mathbf{d} \cdot \mathbf{t} > 0\}|$ is the number of documents $\mathbf{d}$ where the term t appears, with $\mathbf{t}$ its unit vector representation.

Naive Bayes itself assumes that words are drawn independently from a multinomial distribution and that, given a class label c from the prefectures of Japan:

$$\mathrm{P}(c|\mathbf{d}) = \frac{P(\mathbf{d}|c) \times P(c)}{P(\mathbf{d})} \tag{4.2}$$

However, straightforward use of Tf-Idf metrics with Naive Bayes not being efficient and accurate enough, we use bi-gram classification where the **t**'s are bi-morheme combination features drawn from the text. The Naive Bayes algorithm as well does not perform well enough for text classification, so we instead use a Transformed Weight-normalized Complementary Naive Bayes algorithm [Rennie et al., 2003]. Further improvements of the classification algorithm are the subject of Chapters 2 and 3. The score for classifying a document d into a class c is calculated as:

$$\begin{aligned}\hat{\mathrm{w}}(c|\mathbf{d}) &= \log P(c|\mathbf{d}) \\ &\quad - \sum_t \mathrm{tf}(t,\mathbf{d}) log(\frac{1+\sum_{k=1}^{|C|}\mathrm{tf}(t,c_k)}{N+\sum_{k=1}^{|C|}\sum_{x=1}^{N}\mathrm{tf}(x,c_k)})\end{aligned} \tag{4.3}$$

To decide whether or not contents should be filtered out, we use a confidence threshold where the classifier confidence is determined by:

$$\gamma(\mathbf{d}) = \hat{\mathrm{w}}(c_1|\mathbf{d}) - \hat{\mathrm{w}}(c_2|\mathbf{d}) \tag{4.4}$$

with

$$\mathrm{c}_1 = \arg\max_{c\in C} \hat{\mathrm{w}}(c|\mathbf{d}), \mathrm{c}_2 = \arg\max_{c\in C\setminus\{c_1\}} \hat{\mathrm{w}}(c|\mathbf{d})$$

where c_1 and c_2 are respectively the first and second class labels where **d** weighs the most, the first two classes for which the classifier is most confident.

We conducted a classification experiment through varying threshold of confidence value, using 8,811 news articles related to Japanese prefectures crawled from Yahoo! Japan News[1] during Jun. 13 to Jul. 12, 2011.

Figure 4.2 shows the confidence of the classifier when tested with random noise text that is likely to be mined in production. We can then decide an appropriate threshold by favoring precision over recall. This strategy enables us to filter contents irrelevant to the interests of regional communities.

4.3.3 Clustering by Events

To extract events from news articles, the system uses a cosine measure based on tf·idf. Each dimension of a document vector corresponds to a separate

[1] http://headlines.yahoo.co.jp/hl?c=loc

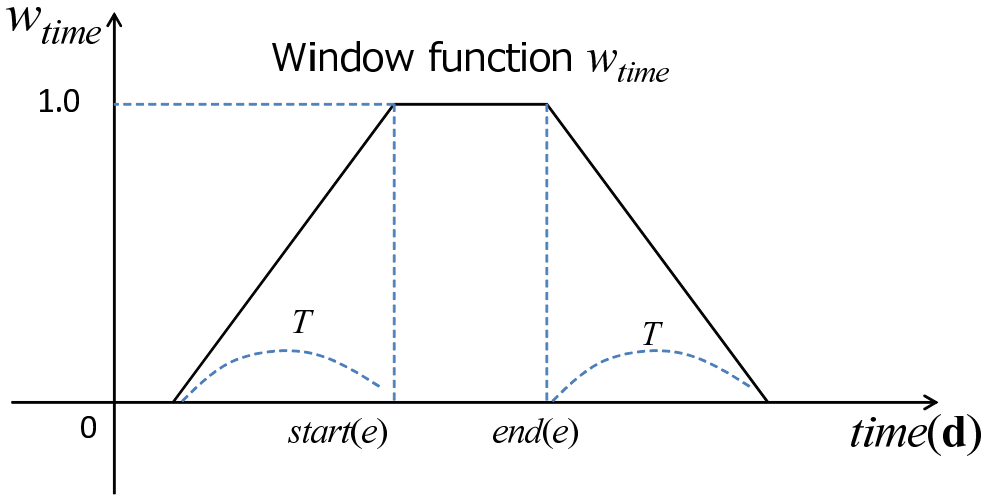

Figure 4.3: Window function for considering dates/times the news articles were published

term, and each component corresponds to an evaluation of the term. However, the system performs the calculation by assigning a certain weight to the term based on the time the articles are delivered.

The similarities $\mathrm{sim}(d, e)$ with a new news article d and each event e on SOCIA are calculated when the article is saved on SOCIA. If the similarity is greater than a threshold θ defined empirically, the article is linked to the event. If all similarities are less than the predefined threshold, the system makes a new event from the article. The similarity is calculated with bag-of-words vectors of d and e consisting of TF*IDF values, term weights w_{term}, and window function w_{time} (shown in Fig. 4.3) as follows:

$$\mathrm{sim}(\mathbf{d}, e) = w_{\mathrm{time}}(time(\mathbf{d})|e)\times \frac{\sum_t w_{\mathrm{term}}(t|\mathbf{d})\mathrm{tfidf}(t, \mathbf{d}, D)\mathrm{tfidf}(t, e, D)}{\sqrt{\sum_t w_{\mathrm{term}}(t|\mathbf{d})^2\mathrm{tfidf}(t, \mathbf{d}, D)^2}\sqrt{\sum_t \mathrm{tfidf}(t, e, D)^2}} \tag{4.5}$$

with

$$w_{\text{term}}(t|\mathbf{d}) = \begin{cases} \alpha & > 1, \text{ if the term } t \text{ appears in } \mathbf{d} \text{ 's title} \\ 1 & \text{otherwise,} \end{cases}$$

$$w_{\text{time}}(time(\mathbf{d})|e) = \begin{cases} 1 & \text{if } start(e) < time(\mathbf{d}) < end(e) \\ \frac{end(e)+T-time(\mathbf{d})}{T} & \text{if } end(e) < time(\mathbf{d}) < end(e) + T \\ \frac{time(\mathbf{d})-(start(e)-T)}{T} & \text{if } start(e) - T < time(\mathbf{d}) \\ & < start(e) \\ 0 & \text{otherwise,} \end{cases}$$

where let $time(\mathbf{d})$ be a published time of $\mathbf{d}$, $start(e)$ be a published time of the earliest article included in e, and $end(e)$ be a published time of the latest article included in e. The similarity threshold and the weight of terms appear in news title were empirically set as follows: $\theta = 0.4$ and $\alpha = 3.0$.

SOCIA stored 54,854 news articles and about 13,000 ones classified to prefictures[2]. Fig. 4.4, the distribution of news articles per event, shows that 34,971 events are extracted through clustering the 54,854 articles.

The system also calcurates the similarity scores between all events stored on SOCIA. For example, the similarity between event e_i and event e_j gets greater score than a threshold, the system treats that event e_i is related to event e_j. The similarity is formulated as follows:

$$sim(e_i, e_j) = \frac{\sum_{k=1}^{N} w_1(e_i, n_k) \cdot \sum_{k=1}^{N} w_2(n_k, e_j)}{\sqrt{\sum_{k=1}^{N} w_1(e_i, n_k)^2} \cdot \sqrt{\sum_{k=1}^{N} w_2(n_k, e_j)^2}} \tag{4.6}$$

In this formula, $n_k (k = 1, 2, \cdots N)$ are news articles linked to both events e_i and e_j. $w_1(e_i, n_k)$ is the similarity of e_i and n_k calucrated by cosine measure, $w_2(n_k, e_j)$ is that of e_j and n_k.

[2]The number of news articles stored in SOCIA was counted on Mar. 16, 2012. It has been constantly increasing.

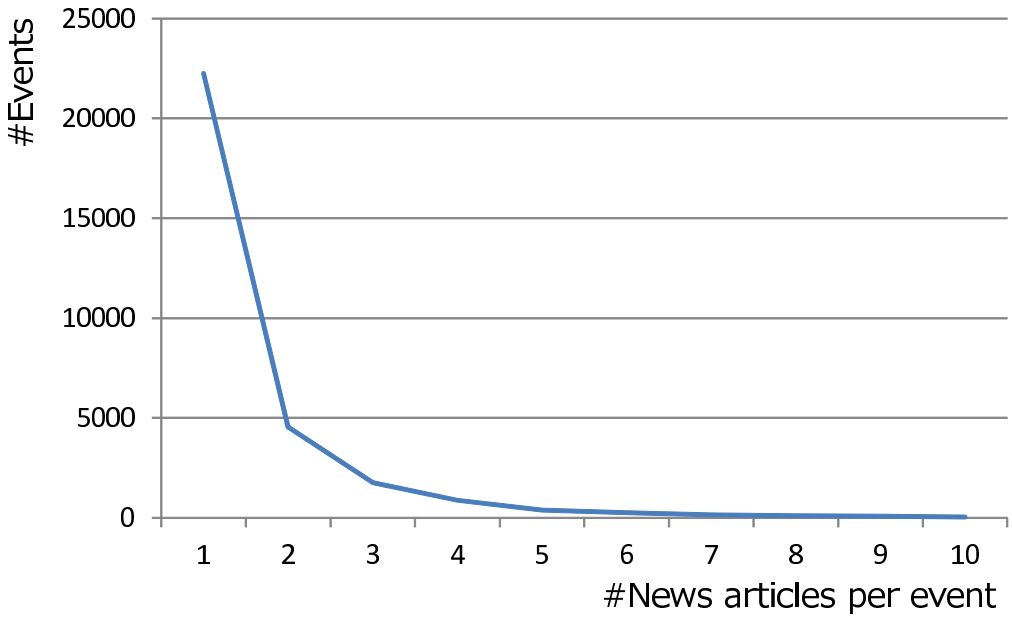

Figure 4.4: Distribution of news article counts per event

4.4 Application:citispe@k

4.4.1 Summary of the system

Citispe@k is a debate support system based on SOCIA, implemented as a Web application, usable on Web browsers. For mobility and reach, Web browsers running on smart phones and tablets are supported.

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 about or sort out regional issues with referencing news articles, tweets or other relevant resources on the Web by 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.4.5 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

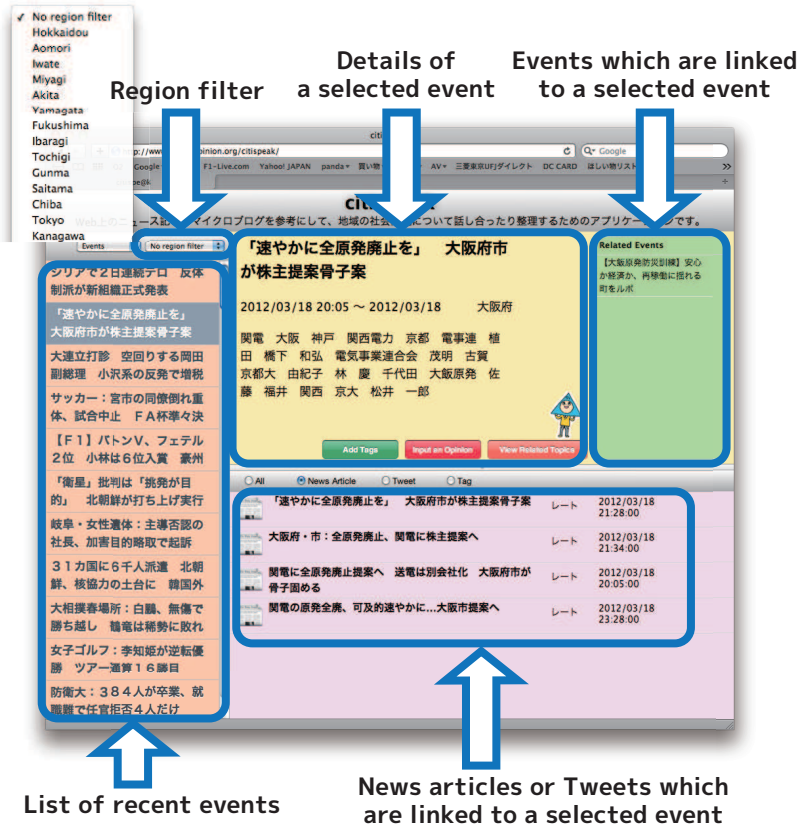

Figure 4.5: The screenshot of citispe@k

from the list, information about the event is shown on the right side of the screenshot. Information consists of news articles, tweets, events related to the event. Those resources can be easily shown and visualized in an iFrame without leaving the system. Fig.4.6 shows a screenshot of a user selecting a news article from the list. In Fig.4.6, the system header has three buttons. The button "Add Tags" is used to add tags to the news article, the button "Input an Opinion" is used to add a comment to the news article and the button "View related topics" is used to see topics which are linked to the news article. The added comment is treated and structured as an opinion. The comment can also be posted to Twitter (via @citispeak for now) to further its reach, and is stored on SOCIA. Optional Twitter accounts will be supported

Figure 4.6: Add an opinion to the news article which a user is viewing

soon.

Users can create discussion topics with relations to events, news articles and tweets. "View related topics" button in Fig.4.7 is used to see topics which are related to the event when users are viewing a event. Users can make a new discussion topic about the event by clicking the "Make a new topic" button. The cycle of the discussions in citispe@k is that users browse events, get related topics about an event, and add their opinion to the topic they are interested in. The system supports to add some Web contents to topics as information sources for discussion, not only add opinions to topics.

4.4.2 Practical use for concern assessment

We define concern assessment as follows: to analyze a trend in citizens' awareness or anxieties of the social issues. For example, committee for science verification about road construction in Aioiyama-Ryokuchi Park in Nagoya City, analyzes road construction[3]. A report of construction analysis was made based on five criterias, "economic chance", "life, educational or cultural chance", "safe, security" and so on. Thus, classifing opinions based on criterias is effective for concern adjustment.

[3]http://www.city.nagoya.jp/shisei/category/53-3-7-4-0-0-0-0-0-0.html

Citispe@k provides above-mentioned tags. Concretely, users can add tags composed with criteria and polarity to them such as "Environment +" or "Environment -". citispe@k also provides tags which can be used to express the kind of opinions like "Question", "Idea", "Refutation". If events or news article have many these tags, the tags will support the analysis of concerns. Viewed in supporting debates, citispe@k has similarities to the model of QOC[MacLean et al., 1991] or Deliberatorium[Klein, 2011]. In the QOC model, an issue is structured by theree nodes "Question", "Option" and "Criteria". Deliberatorium supports debates or opinion adjustments by classifying opinions into "Agree" or "Disagree" and kinds of opinions. Tagging to opinions are voluntary in citispe@k due to getting many opinions. If tagging were required in citispe@k, users would input little opinions. Users carefree input their opinions with no tag, but the opinions with tags are well conjugated in concern assessment.

Fig. 4.8 shows the example of tagging to an event. The "Add tags" button will show you the tagging view. In the tagging view, existing tags are listed in the popup menu. Users select appropriate tags from it.

4.5 Comparison with Other Tools

4.5.1 Comparison Criteria

Based on our study, we propose the following criteria driven by effective e-Government and e-Participation requirements for comparing e-Participation support systems for regional communities:

1. Transparency
2. Data Re-Usability
3. Pervasiveness
4. Discussion Appeal
5. Structuring

Transparency is a requirement of all Government 2.0 initiatives. It consists in the open publication of available data by organizations. We extend this definition to include data inherent to the discussion system (comments,

Table 4.1: Comparing O_2 with Existing Tools

e-Participation Tool	**Transparency**	**Data Re-Usability**
Message Boards	Site-dependent	Format-dependent
Chatrooms	Site-dependent	Logs
Questionnaires	Public Polls	Figures
SMS Questionnaires	Public Polls	Figures
Site Comments	Site-dependent	Format-dependent
Microblogs	User-dependent	API (Twitter)
MIT Deliberatorium	Public (login)	N/A
Cohere	User-dependent	Dataset (planned)
O_2	Public	Dataset, Ontology, API

Pervasiveness	**Discussion Appeal**	**Structuring**
Mainly Web	Site-dependent	Time-centric
Requires Real-time	Site-dependent	None
Medium-dependent	Not proactive	Dependent
High	Not proactive	Dependent
High	High	Article-centric
Very high	High	Hashtags
Mainly Web	Medium	Issue-centric
Mainly Web	(out of domain)	Knowledge-centric
High (mobile-oriented)	High	Automated Event-centric

discussions). In most cases, transparency is either site-dependent, or making data available in a different form (polls). User-dependent transparency means that a level of privacy exists for which users can conduct private interactions (e.g Twitter). When the transparency is public, this means that all data from the system is made public.

Data Re-Usability relates to the potential of data to be re-used without technology-specific boundaries. As e-Government tools are not limited to the Internet (e.g use of SMS) or one specific application, data that is relevant to the citizens ought to be free from such boundaries so that it can be accessed by various tools and applications. It should also be structured according to the domain of debate, rather than a tool-specific domain. N/A (for non-available information) means that some form of interoperability for data has been suggested but was not found in our tentative use.

Pervasiveness requires the system to be ubiquitous in its accessible interfaces. Here, it describes a system that can be used from any terminal, most notably tablets and mobile phones since they have become the first computing platform and thus a much easier means to reach citizens. This qualifies the capacity of the tool to penetrate everyday use according to technology trends.

Discussion appeal is defined as the capacity of the system to engage its users in proactive discussion by interesting them to a topic at hand. For example, citizens are more eager to react to ongoing events and current news than they are to use e-participative involvement tools that require a certain degree of initiative and interest for debate. Thus, from this point of view, capacity of news sites to involve can be for example assessed through the number of comments that readers post on news articles. Particular events often being a catalyst for regional communication and interaction, in this sense regional news sites when they exist are more able to involve than are dedicated e-Participation systems. In this sense, we believe that a system with higher chance of using the swarming effect (see Sec 4.2) has a higher chance of involving citizens, a higher discussion appeal.

Structuring is an important vector of debate, as it helps focus the discussion for every participant. It involves 1. adequate separation of items relevant to different regional stakeholders 2. clustering of similar or related items in an understandable way to avoid redundancy but also provide numerous information sources relatively to one topic of discussion so that bias is avoided. E.g, news aggregation systems provide such clustering, but do not allow commenting on the clusters themselves.

4.5.2 Discussion

We conduct a qualitative comparison in Table 4.1 between O_2 and a panel of eParticipation discussion support tools: message boards, chatrooms, questionnaires, site comments, microblogs, the MIT Deliberatorium, and Cohere.

The data of message boards, when it is publicly available, is normally highly format-dependent, depending on the message board software used (phpBB, vBulletin, Futaba, Facebook groups, etc), and the view in case one wants to perform page scrapping. Also, its structure is mostly time-centric, as criticized by the MIT Deliberatorium research. The use of swarming to involve users also depends on the message board. A message board addressed to a local community that provides discussion opportunities about current events regularly would have a good capacity to involve citizens, despite its lack of automation and structuring. The lack of automation requires a lot of community management work to find articles, tweets and other background information to enrich threads. The lack of structuring could lead to soapbox and balkanism issues. Finally, message boards are used mainly on the desktop-PC Web and their use is declining overall, in profit of news sites commenting systems and social networks such as microblogs.

Chatrooms are highly site-dependent when it comes to transparency, depending on whether or not logs are made public, and private logs retrieved from users raise the question of trustability. There is also no structuring whatsoever and the data must be sequentially cut. This can be done using previous research we have conducted [Shun Shiramatsu and Okuno, 2010], however although this is useful for discourse analysis, the problem of lack of overall structure remains. Finally, chats may be pervasive depending on the system, but they require real-time involvement, which is an important drawback compared to other systems where discussion can be conducted asynchronously.

Questionnaires and SMS questionnaires, by being highly focused on specific problems, and normally pervasive (especially in the case of SMS questionnaires), are guided discussions that have high capacity for involvement. Flocks can be created although users are not proactive in this form of swarming (if there is no questionnaire, there is no flock). However, they normally have no discourse structure, and their ability to create long-term discourse communities after a flock is subject to doubt. There is appeal to answer and emit opinions, but not specifically to discuss. Finally, data that is made transparent normally consists in public poll figures. In short, questionnaires

cannot be considered as a complete eParticipative tool for communities, although they are best as part of one and useful for input probing [Phang and Kankanhalli, 2008].

Site comments basically suffer from the same drawbacks as message boards in terms of public data and format dependence. Important news sites in Japan require a paying subscription to access archives and their comments. Also, page scrapping can be rendered difficult by the Javascript-controlled display of comments. However, news sites are pervasive since a vast majority propose mobile interfaces and/or applications. They also show a very high swarming effect and discussion appeal on local news that communities feel concerned about, only it is not leveraged to create long-term debate. Finally, the main drawback of site comments is that they are article-centric and centralized on only one news site at a time. There is no possibility to comment a cluster of news as an event, and comments (which constitute basic discourse) have no exploitable structure.

Microblogs are probably the most pervasive, highly used on mobile interfaces by a various panel of citizens, their main representative being Twitter. The swarming effect is also important, as hashtag trends emerge in correlation with important events in time, and the discussion appeal on microblogs is high. However, as for site comments, there is few leverage of the swarming effect for structured discourse for regional communities, despite on-line formation of Twitter-localized communities. Still, because it is highly pervasive and has a highly exploitable flock, Twitter through its API data re-usability is mined by O_2 in hope to utilize tweets as discussion seeds and contributions.

The MIT Deliberatorium is mainly a desktop Web interface with issue-centric structure. Although in theory it addresses long-term general issues, when we experienced the site in May 2012, present discussion was focused on punctual events. Thus it could exploit swarming, however there is no automation and discussion seeds have to be created manually.

Cohere was added in this comparison because it is probably the closest system to O_2 in that it chooses a strict ontological approach for structure. As Twitter, it has public and hidden data. According to [Liddo and Shum, 2010], dataset intercompatibility was also planned. Still, it is highly knowledge-centric, focused on ideas and viewpoints. Although it is appealing to share concepts and knowledge, we believe it is slightly off-domain to constitute an eParticipative tool for local communities.

Lastly, we will explain why we believe O_2 is a better fit in the context of involvement of communities and eParticipation according to the criteria. First,

O_2/citispe@k focuses on being as transparent as possible, as per instructed for the Government 2.0 initiatives. The data is entirely publicly accessible and reusable through its dataset, ontology and API. Finally, by employing an automated event-centric structure through regional classification and event clustering of automatically mined news articles and tweets, O_2 aims at leveraging the swarming effect. Second, the interface is mobile-oriented (but can be used through a desktop browser as well), which allows it to be more pervasive according to current technology trends. Third and finally, since it is easier to engage citizens through short-term flocks [Della Porta and Diani, 2006] e.g those of service issues or current events, it is fair to assume that engaging debate and discussion through communities on geographically and timely local issues can lead to more involvement on the long term.

4.6 Conclusion

We have presented the e-Participation Web platform O_2, for public involvement of citizens from regional communities in dialogue and participative debate through the use of an innovative technological approach. Following the delivery models of e-Government, the scope of this research covers the informing and involving of the citizens in discussion about regional issues. Representatives can then utilize the tool and the data is automatically openly published in a LOD subset we call SOCIA. The innovative point of the platform is to allow building discussion topics for regional communities by directly commenting on news clustered as events and classified automatically by geography. By assessing the goals of an e-Participative system in the context of e-Government and mobility, we proposed relevant comparison criteria for eParticipative tools to be used by regional communities, and then compared O_2 to existing technologies for supporting regional debate among citizens. Upon this qualitative study, we claim that O_2 can constitute a better approach than existing tools of support, through high focus on openness, data re-usability, pervasiveness, discussion appeal and automated event-centric structuring.

Further research directions that are considered include: sentimental analysis on to visualize concern more easily, development of an opinion search engine, deeper structuring of debate through ontology following previous works [Hurwitz and Mallery, 1995], insight detection on comments based on generative models [Blei et al., 2003].

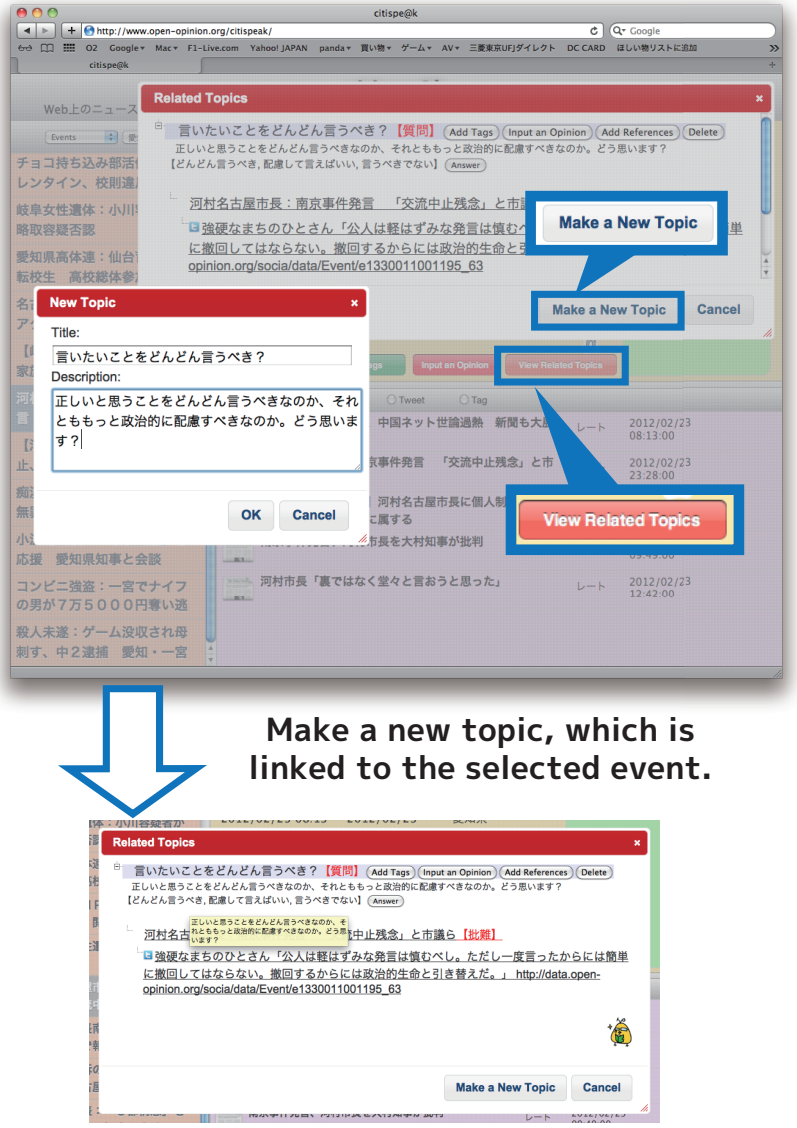

Figure 4.7: Make a new discussion topic to the selected event

Figure 4.8: Add tags to the selected event

Chapter 5

Recommending Contents for Live Discussion Support

5.1 Introduction

In a previous chapter [Swezey et al., 2010], we proposed an architecture for a Multi-Agent System which would assist bloggers and webmasters of topic-oriented content sites to find suited and fair affiliation proposals for their affiliate page slots. We aimed at reducing the load in the affiliate search process and helping the mutually winning strategy when exchanging respective links on websites. In another chapter, we have presented as work in progress the system on which we rely for heavy data harvesting and on-the-fly web page modification with push-type technology [Nakamura et al., 2009a]. This system allowed us to push content in real-time at precise locations on a web page, using a server-side scheduler.

In this chapter, we present a system which lets Web pages stay up-to-date by using Real-Time Affiliate Content Agents. The implications of this technology are broader, but in the case at hand and current chapter we name them Recommender Agents.

This system makes use of Page Agents, Agents nested inside Web Pages, which assist in improving the contents of the *related articles* section of a given article. More than helping find useful affiliates for shared sections of the site such as the *blogroll* [Swezey et al., 2010], it actually selects the latest up-to-date related articles in real-time for the pages the agents run on. The set from which the related affiliate content can be open (all existing web

content in the system) or closed (affiliate content only).

With a sample implementation as part of the development of a Web platform of which the objective is to involve citizens in regional debate in Japan, we show that our architecture is valid for real-world implementation, if realistic performance requirements are met with the classifier. In the sample implementation, the latest up-to-date news and articles can be fetched from the open Web, and then classified for each city of Japan, then pushed on articles and news related to the same city.

Firstly, we will begin by stating the context of the research. Secondly, we detail our architecture and how our current system works. Thirdly, we will show the sample implementation of our system, which fetches latest affiliate news contents related to 47 Japanese prefectures and over 1700 Japanese cities. We detail technical challenges and show achievements and performance. We then conclude and discuss further research about the subject.

5.2 Context

5.2.1 Affiliation

We call affiliation, the process of linking between websites, when this linking holds more meaning that of a simple reference. Besides friendship or common interest, affiliation seeks to share visitors, as well as raising site awareness mutually. Affiliation is also most commonly defined by commercial affiliation, i.e a generally unidirectional link from an editor (blog, website) to an announcer (commercial or promotion site), in a textual or graphic fashion.

The first important dimensions in settling community or commercial affiliation, consists in the statistics of the editor. If this is a community affiliation, then it generally concerns both sites since they are both editors. Those statistics are considered more or less importantly in community affiliation, but they are vital in commercial affiliation, especially when there is no third-party broker between the editor and the announcer. They are also important for the webmaster to study the audience. This dimension was assessed using the Agents described in section 5.2.3.

The second important dimension is context. Affiliate Content does not solely consists in community affiliate links or advertisement blocks, it can also be imported textual content such as news feed content. At any rate, we define affiliate content by contents which are related more or less to each page's

topic, or to the reader's profile (targeted advertisement). Therefore, affiliate content that can be pushed dynamically depending on the context of the page is of high value to advertisers, or news/topics background and continuity investigators such as those concerned by our sample implementation.

5.2.2 Limitations of Recommended Contents Sections

For commercial affiliate contents:

Most context-related affiliate content today is provided through a system known as sponsored search auctions [Lahaie et al., 2007]. This system however is flawed in the sense that any bidder with enough bidding power can win the auction and show totally unrelated content for any keyword, so long as its bid for it is high enough.

For related topic contents:

Often, *related contents* sections on Web articles will present flaws such as the following:

- It is frozen in time: cached at the time of page generation, or used as a source at the time of writing. It does not recommend latest up-to-date content on the topic, if the article is too old but the topic itself still has ongoing events.
- It is limited to the domain/site at hand and does not recommend content from other sources.

On recommender systems and synchronous Javascript:

Though little research seems to have been done on these issues, we can notice a general trend in real-time recommendation that is aimed at live information streams such as Twitter [Chen et al., 2010, Phelan et al., 2009]. Our system has the same objective, only that it recommends related contents in real-time directly when users are browsing article pages relevant to a given topic.

5.2.3 Technology Developed Until Now

Using our previous research utilizing channelizing technology and server-side scheduling [Nakamura et al., 2009a] (see Fig. 5.1), website or blog editors can

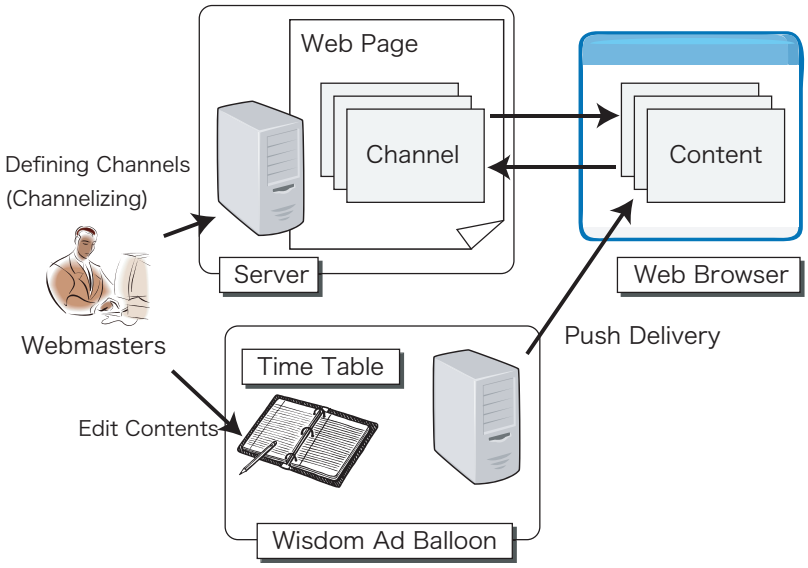

Figure 5.1: Push-type Content Delivery Mechanism on Wisdom Ad Balloon

measure the utility of each page slot on the web page, for the various affiliate content blocks in which contents can be recommended. This utility can be measured gathering statistics from various data, such as page scrolling, mouse movement, and so forth. Secondly, the event data is analyzed and interpreted to modelize reader behavior. Those functionalities are implemented using machine learning technology [Velayathan and Yamada, 2006]. They can push contents in real-time at the location they want on their Web pages, at the time they wish.

This Agent technology can be used independently on CMS-generated Web pages, such as typical blog pages or news sites, as well as on plain HTML pages. One of the system's strong points is that it requires only to modify the HTML page, or the layout template in case of a CMS. In the latter case, since modifying the layout template is generally much easier and do not require any advanced programming skill at all. We have further improved it in the research at hand, with Recommender Agents.

Table 5.1: Modules

Feed Engine	Aggregates RSS feeds from various sources and process new articles
Agent Server	Server interface for the Recommender Agents
Classifier Engine	Classifies the new articles: can output a single class or a vector of classes for the articles (in case we want to recommend by cosine similarity)
Database	Known articles and their respective classes
Recommender Agents	Page Agents running on instances (Web pages opened by users) of articles and recommending content accordingly

5.3 General Architecture

In this section, we talk about our new framework and its architecture. We introduce a new architecture for real-time recommendation of contents. Hence, in the rest of this chapter, we assimilate the terms *Page Agent* and *Recommender Agent*.

5.3.1 Modules

Our system relies on the following modules:

5.3.2 Agent Interface

Using our previously implemented system and considering other research (see previous section), we have taken our Page Agent framework a step further, by allowing it to use synchronized Javascript objects over numerous instances of opened pages in many browsers.

The Agent consists of two parts:

- The Agent itself, which interacts with the page and replaces page slots with content by channelizing, and can store his own history and local data.
- The shared memory, a synchronized Javascript object shared by the Agents over all article pages opened in browsers which relate to the same

class (topic). We call this object a Wisdom Shared Object (WSO).

As all operations on the synchronized socket objects are performed on the client side, it is also important to take into account the atomicity of some of these operations. Data corruption can occur if two or more browser agents (i.e execution threads in the general case) try to write into a shared property of each synchronized agent, or if one agent reads a shared property and begins an operation on it before another has finished writing into it. A trivial example being that of a counter incrementation, with the counter stored in a shared property. In the operation counter = counter + 1, run by agents A and B using the shared property counter, a save (write) can occur between the load (read) and save (write) of this operation. To address this synchronization issue, several traditional methods which can be adapted exist [Lamport, 1974, Peterson, 1981], but we choose a timestamp/revision number technique [Cannon and Wohlstadter, 2010] which is more up-to-date and best suited for client-server handling of conflicts.

5.3.3 Workflow

Client-Server workflow:

1. A user opens the article a in a Web browser.

2. The Agent located on a sends a's URI and contents to the Agent Server when it is opened on the Web for the first time, for example by his author. This article can also be input by other sources such as a Feed Engine.

3. The Agent Server interrogates the Database to find out a's topic class c_a. In case no result is found, i.e a is new to the system, the Agent Server retrieves a class c_a from the Classifier Engine and saves (a, c_a) to the Database.
   In the case of a vector implementation, instead of just c_a we have a vector of pairs $(c_i, w_i)_a$ with $w_{i,a}$ the weight of the geographic class c_i in relation to a.

4. In response to the Recommender's Request, the Agent Server pushes c_a, or the vector of pairs $(c_i, w_i)_a$ in the case of a vector implementation.

5. The Recommender fetches a Javascript shared object, building its identifier depending on c_a or the vector $(c_i, w_i)_a$.

This object contains recommendations related to a's geography, and can be fetched from the same Agent Server or another one.

Recommender Agent workflow:

1. The Agent builds automatically an Article Object out of the Web Page it runs on, by extracting HTML tags marked with specific ID's. The final object has the same properties as a RSS feed item or HTML5 article: URI, title, article text, etc.

2. As seen on the previous workflow, the Agent sends the Article Object to the Agent Server in order to get either its most probable Topic Class as a scalar, or a Topic Class vector.

3. After having been given a Topic Class ID by the Agent Server, the Agent loads a Shared Object (see Sect. 5.3.2) associated with the ID of the Topic Class, which is a synchronized shared Javacript object between Agents running on similar topic pages.
   In the case of a vector implementation, the shared Javascript object is associated to the closest centroid vector of clustered articles dimensioned by the geographic classes, which we can find with cosine similarity for the article at hand.

4. The Agent, as well as Agents using the same shared object, reorganizes the recommended contents in it. They can be extended to use their own heuristics: article's date of publication, visitor counter for each article, and other heuristics depending on the Agent.

5. The Recommender Agent shows the recommended contents, and modifies it in real-time if necessary, in a *related contents* section of the Web Page.

5.4 Sample Implementation and Results

5.4.1 Aim and Original Project

We implemented a sample system based on this architecture, as part of the development of a Web platform of which the objective is to involve citizens in regional debate, based on [Macintosh et al., 2009]. This platform is itself a

project funded by the Japanese Ministry of Internal Affairs and Communications, as part of the SCOPE [1] competitive nationwide research development programme in Japan. In this sample implementation, we aim to fetch the latest up-to-date news and articles from the open Web, classify them for each city of Japan, and push them on articles and news related to the same city, to local keep citizens informed and aware.

5.4.2 Classifier Module

For the Classifier Module, we chose to use a Complementary Naive Bayes classifier (CNB) [Rennie et al., 2003], because of the lack of data for certain classes, e.g cities and prefectures of Japan which are not popular and do not provide with an amount of training/testing data that can be compared against, for example, the wards of Tokyo metropolis.

The dataset for training and testing was constructed from the Japanese Wikipedia article base, by adding each article as a document for the prefecture, city, or district class it relates to. For all 2024 classes [2], a total of 105018 documents was gathered, which amounts to 299889 documents when building the hierarchical set. Documents from each city are duplicated for the prefecture they belong to, and documents from each ward are as well duplicated for the city they belong to [3]. We conducted a closed-test.

To train and test the classifier, we used Apache Mahout [4] over an Apache Hadoop [5] cluster of 3 machines. Mahout implements CNB as a Transformed Weight-normalized Complement Naive Bayes (TWCNB) as described in [Rennie et al., 2003] (steps 1-8). We modified the CNB algorithm of Mahout and added a few other features so that the classifier could be tested in two fashions: flatly and hierarchically. Because our geographic topical classes can be broken down into a three-level class tree (prefectures/cities/districts), and because of the large amount of classes, we felt that implementing a hierarchical testing of documents [Demichelis et al., 2006, Langseth and Nielsen, 2006], as opposed to a naive *flat* testing, would let us achieve better results

[1] SCOPE: Strategic Information and Communications Research & Development Promotion Programme

[2] 47 prefectures, 1811 cities including the Tokyo metropolis wards, and 166 wards

[3] During the hierarchical classification, the metropolises of Tokyo, Osaka, Kyoto count as prefectures, and their districts (wards) as independent cities.

[4] http://mahout.apache.org/

[5] http://hadoop.apache.org/

Table 5.2: Performance of Flat Classifier

Classes	1810 cities
Classes for flat training	1810 (cities only)
Correctly Classified Instances	69944 (42.086%)
Incorrectly Classified Instances	35074 (57.914%)
Total Classified Instances	105018
Avg. Time to Classify one Instance	522 ms

Table 5.3: Performance of Hierarchical Classifier

Classes	2024: 47 prefectures 1810 cities 167 districts
Correctly Classified Instances	69944 (66.602%)
Incorrectly Classified Instances	35074 (33.398%)
Total Classified Instances	105018
Avg. Time to Classify one Instance	38,6 ms

in:

- **Accuracy:** If the classes can be broken into a class tree, especially when the number of classes is important, we have more chances of achieving better accuracy if we can classify under one common/parent category (the prefecture) before looking for the right child one (the city), instead of directly looking for the city.

- **Performance:** The first classification check is against 47 prefecture, and the second against $\frac{1810}{47} \approx 38$ cities, which amounts to roughly 85 score calculations for each document, if classes were equiprobable. If we did a flat classification directly over the 1810 cities, it would amount to 1810 score calculations, and result in a 2029% time cost increase.

We can see here that the hierarchical classifier here outperforms the flat classifier for real-world application, notably in processing time performance. However, the accuracy of the hierarchical classifier can be improved, it will be the object of further research. We also intend on doing open tests as well as N-fold tests of the classifier.

Table 5.4: Javascript *Wisdom Shared Object* Delegates

open()	Actions to perform when the shared object is opened on the current Web page (e.g populate the *related contents* list next to the article)
close()	Actions to perform when the shared object is closed on the current Web page
update()	Actions to perform when the shared object is modified by this or another Agent using the object, and its new state pushed to the Page
error()	Actions to perform when an error occurs

5.4.3 Page Agent

The Page Agent Module was implemented using the WebSocket protocol of HTML5 (draft 76 [6]).

After having followed the protocol described in Section 5.3.3 and fetched the shared synchronized Javascript object related to the Topic Class (here, a city) related to the article currently displayed, the Agent implements the following delegates of the object:

One of the interesting actions that the Recommender Agent can perform outside the real-time modification of the related contents section of the Web Page is the modification of the Page template itself, according to what is indicated for the Topic Class of the shared object it fetches.

Extension to custom affiliate template modification:

This type of affiliate content is usually more characteristic of commercial affiliation. It basically consists in modifying the appearance (background, colors, images), namely the template, of the main page, or a specific subset of pages, depending on the contract (see Fig. 5.2). The resulting appearance helps the affiliate by raising its brand awareness, e.g showing the brand name or advertising about a product of the brand. One good example of this custom affiliate template strategy can often be seen on video game news sites, promoting an editor's new game.

One example application in the case at hand is, for example, when reading an article about a particular city of Japan, recommending the forthcoming

[6] http://tools.ietf.org/html/draft-hixie-thewebsocketprotocol-76

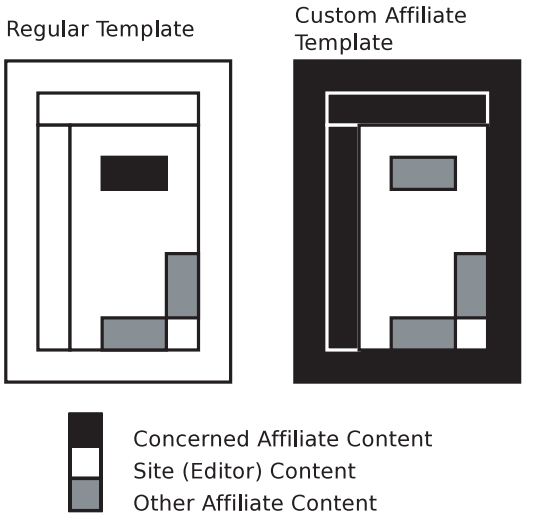

Figure 5.2: Differences between Regular Affiliate Content Blocks and Custom Affiliate Templates

matsuri [7] of this city by changing the appearance of the news site or blog being browsed. Another example is a more visible alert in case of localized disasters such as earthquakes (we have developed such a system, which we call Wisdom Alert).

5.5 Conclusion

In this chapter, we have introduced a novel method for recommending the latest up-to-date contents in a set of related events or topics, dynamically and in real-time, on any article Web page that is related to the events or topics. We made use of persistent and synchronized recommender agents which can easily be extended to implement a lot of functionalities.

We considered accuracy, processing time and ease of implementation in our approach. We have addressed each of these challenges in a example implementation of our architecture. In this implementation, we classify articles from more than 47 Japanese prefectures and over 1700 Japanese cities. By making use of hierarchical classification, we are able to categorize articles

[7] Town Festival. Most Japanese cities have their own.

faster and more accurately, and demonstrate that such a system is scalable and usable in real-world applications. As for the recommender in the client Web pages, it is based on Javascript agents with synchronized shared memory as well as local memory, which require no server code, only a request handler or proxy on the same domain. The algorithm can also be used with traditional AJAX and long-polling to suppress this need.

The challenges in research to come are the following: improve placement on the recommender slots not only according to, but through more subtle heuristics which can incorporate but are not limited to: CTR, impact factor of the article, page ranking, etc. We also need to improve the efficiency of our Complementary Naive Bayes classifier in the sample implementation. For the improvements, we plan on testing with bi-grams and tri-grams, and dealing with the classes holding the biggest amount of training documents (Tokyo, Osaka, Hokkaido). As for the testing of the classifier, we are planning to do 10-fold cross validation tests.

Finally, in further chapters, we will present new modules in our system for supporting large-scale participative debate.

Chapter 6

Conclusion and Summary

Recall the main contributions of this work:

1. After conducting a survey we propose the meta-algorithm SAHB (Chapter 2) for dataset restructuring which is a novel method of re-sampling for Bayesian text classification based on trees along with clustering of classes. The method shows promising results with low bias on a training fit on the Wikipedia Japanese dataset. When generalized this algorithm can be the better approach for problems with very large numbers of classes that are imbalanced, thanks to its hierarchization, node clustering approach and log-time complexity.

2. We have devised a method to use only one trained Bayesian model to be able to filter out irrelevant contents without having to train one more algorithm (e.g. Support Vector Machines), to filter local against global contents. Our method shows excellent results in precision/recall in Chapter 3.

3. For text classification, increasing the number of grams and using the better algorithm TWCNB (the straightforward approach which we use as a control experiment) is not enough: dataset structure refinement through adequate re-sampling (Chapter 3) and/or hierarchization (Chapters 2 and 6) is necessary, otherwise there can actually be an accuracy and performance drop when using better methods than simple Naive Bayes and more features which would normally help on a training fit.

4. We contributed a novel e-Participation system in Chapter 4 that utilizes classification, clustering and annotation with an adequate ontology.

Through a qualitative comparison we believe the system is best fit to support debate among local citizens in the current context of the Internet, upon improvements.

In Chapter 1, we presented Sophia, a platform for intelligent processing, classification and structuring of fuzzy and unstructured data, so that it can be understood and used (e.g for debate) along various axises (e.g temporal, regional, semantic), by various distributions of population (e.g citizens and decision makers). We presented mainly the technical aspects of the platform.

The platform is modular, makes use of open standards, and can be interconnected in different ways. The mining module can be extended easily to take new data sources in their original form as inputs, classification can be branched to various algorithms, modeling itself is a flexible ontology based on Linked Open Data, and support can be derived in various forms of interfaces. We gave examples of application in each research module that apply to the original aim of the system.

Although the platform is still young, it is currently being used in early production for its first case of application, concern assessment and debate support for social issues. Better tuning of facilities such as the learning algorithms in this precise case is needed to meet the objectives of the original project from which Sophia was born, but we believe the platform's architecture, as shown in this chapter, can help us achieve this goal.

Several works are currently in progress, which make extensive use of Sophia. An improved algorithm for geographical and topical classification, which makes use of Sophia and the open-source Apache Mahout for processing of big data. Another work concentrates on the evaluation of the interfaces for concern assessment and debate on various distributions of population, which is the concrete output of the overall project from which Sophia originated. When Sophia goes public and open-source, we will also provide a more developer-oriented technical work with several cases of application.

In Chapter 2, we presented SAHB, an improvement to straightforward Naive Bayes text classification methods, which is a meta-algorithm that can be trained in an equivalent time to the subsequent classification algorithms it uses, and takes advantage of a known hierarchy of classes as well as semantic proximity. The idea for the algorithm stemmed from a broader project in which one of the research milestones is to classify by using the advantages of online open crowdsourced corpora.

The algorithm needs further generalization and comparisons with other methods, and tests on other datasets, as it is limited in this case to a sample problem. Other cases include but are not limited to: having topic nodes belonging to multiple parents, bias such as equilibrium by clustering parent nodes together cannot be reached, which may require mixing with other techniques e.g oversampling. Finally, it also needs to be tested up to per-city classification in our example, although it is recursively intuitive that accuracy will still be better than straightforward use of TWCNB.

Chapter 3 focuses on the 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: Naive Bayes, Complementary Naive 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.

Chapter 4 introduces the e-Participation Web platform O_2, for public involvement of citizens from regional communities in dialogue and participative debate through the use of an innovative technological approach. Following the delivery models of e-Government, the scope of this research covers the informing and involving of the citizens in discussion about regional issues. Representatives can then utilize the tool and the data is automatically openly published in a LOD subset we call SOCIA. The innovative point of the platform is to allow building discussion topics for regional communities by directly commenting on news clustered as events and classified automatically by geography. By assessing the goals of an e-Participative system in the context of e-Government and mobility, we proposed relevant compari-

son criteria for eParticipative tools to be used by regional communities, and then compared O_2 to existing technologies for supporting regional debate among citizens. Upon this qualitative study, we claim that O_2 can constitute a better approach than existing tools of support, through high focus on openness, data re-usability, pervasiveness, discussion appeal and automated event-centric structuring.

Further research directions that are considered include: sentiment analysis on to visualize concern more easily, development of an opinion search engine, deeper structuring of debate through ontology following previous works [Hurwitz and Mallery, 1995], insight detection on comments based on generative models [Blei et al., 2003].

Finally, in Chapter 6 we have introduced a novel method for recommending the latest up-to-date contents in a set of related events or topics, dynamically and in real-time, on any article Web page that is related to the events or topics. We made use of persistent and synchronized recommender agents which can easily be extended to implement a lot of functionalities.

We considered accuracy, processing time and ease of implementation in our approach. We have addressed each of these challenges in a example implementation of our architecture. In this implementation, we classify articles from more than 47 Japanese prefectures and over 1700 Japanese cities. By making use of hierarchical classification, we are able to categorize articles faster and more accurately, and demonstrate that such a system is scalable and usable in real-world applications. As for the recommender in the client Web pages, it is based on Javascript agents with synchronized shared memory as well as local memory, which require no server code, only a request handler or proxy on the same domain. The algorithm can also be used with traditional AJAX and long-polling to suppress this need.

Chapter 7

Annex: Study on Blog Communities

Original title: Architecture for Automated Search and Negotiation in Affiliation among Community Websites and Blogs.

In this chapter, we present a multi-agent architecture which can reduce user's load when searching for affiliates in a network of community websites. We give a precise definition of the environment, networks of community websites. The system's architecture is designed with scalability and easy interfacing for brokers and matchmakers in mind. We have developed a simulator to see how sites or blogs evolve with affiliation. We also show an example of its output results after a 1500-iteration long experiment on a network of community blogs. In our conclusion, we state several applications of critical interest and further research paths on the subject.

7.1 Introduction

7.1.1 Aim of the Present Research

Our goal in this research is to provide with a multi-agent architecture capable of automating the process of affiliation in networks of Community Websites (CWs). This process leads to an increase of visitor revenue as well as quality for CWs. Improvement on quality of writing is measured by feedback from visitors, potential affiliates, and page ranking. Increase of the visitor revenue comes from interlinking itself.

Affiliation processes, always part of the process of launching a CW, give birth to a need for automation. Furthermore, new blogs being born everyday make the number of potential affiliates soar. Finding the right affiliate can prove difficult. Affiliation links indeed require an explicit effort compared to that required for permalink ones [Marlow, 2004a].

In a previous chapter [Swezey et al., 2009], we proposed a multi-agent architecture capable of automating this activity, to reduce user's load in time-consuming research and negotiation processes for affiliation. Our system searches for potential affiliate CWs and deals with the issue of equity in partnership, as well as quality expectations, before proposing affiliates to the users. The whole research and negotiation part of the affiliation process becomes automated, thereby saving time for the user. In the case of blogs, it takes the idea of blogs being agents [Takeda, 2007] one step further.

In this chapter, we deepen the definition of the environment (Sect. 7.2), and show how our system's simulator can be easily configured and run. We show sample results for a blog community, thereby demonstrating that in this sample case the practical implementation of our system can efficiently reduce user's load in the affiliation process.

7.1.2 System Description

The practical use case goes as follows:

1. User connects to the broker/provider. If necessary, the user registers the site and information about it, unless the provider is already the blog/site's hosting service.

2. User requests a list of potential affiliates, entering the following data:

   (a) Desired minimum fairness of the affiliation - in general, or in terms of visitor revenue, quality, relative importance, and so forth.

   (b) Available spaces to show affiliate links on his own site.

3. System outputs a list of Potential Affiliates (PAs), or none if too unfair.

4. User requests affiliation to one or more PAs and wait for their approval.

The partner PAs need not worry about negotiating, or the user's site being irrelevant to their own expectations. In very simple terms, this use case resembles that of a social networking site, but for community websites.

Our system is also a simulator which generates a cluster of agents, and sample sites based on patterns of particular statistics or expectations. It can be run for any number of iterations. The initial data, as well as the heuristics, depends on the environment chosen. When the simulation ends, the system shows the general evolution of the sites in terms of visitor revenue, quality revenue, and other data if needed.

7.2 Definitions

7.2.1 Community Website

A CW is defined by the following characteristics:

1. Its contents are relevant to one particular subject, or several subject of the same category. Ex: one programming language, or mainstream IT. If they are personal sites, they express only one facet of the webmaster [Cardon et al., 2007].

2. It aims for quality of opinions and utterance of relevant and specialist information.

Sites such as daily-life blogs are therefore excluded by this definition. However, blogs of sociological type III [Cardon et al., 2007]), defined as community blogs, are CWs according to our definition.

CWs are set up on the Internet in order to share knowledge and opinions, but before altruism, one of the main objectives is attention in the community [Dessalles, 1998] (it does not necessarily relates to ego). Therefore, most of the time setting up a community site or blog calls for the process of finding affiliates. Ranking and being well-referenced on search engines calls for quality.

7.2.2 Affiliation

Affiliation consists in interlinking two websites. For blogs, it is a contract passed between the two, a social tie omnipresent in nowadays' blogs. By

placing a link to another weblog in one's blogroll, one assumes that the author either endorses that weblog, wishes to promote it, or claims to read it on a regular basis [Marlow, 2004b]. This has also been true ever since websites existed on the Internet. Besides friendship or common interest, affiliation seeks to share visitors, as well as raising site awareness mutually. What we define further in this chapter as a category is close in concept to an affiliation group [Zheleva et al., 2009].

Respective placement 7.4.3 of each other's link on the blogs/websites is considered as settlement of the affiliation deal.

7.2.3 Visitor Revenue

We call Visitor Revenue (VR) the revenue in popularity that two websites engaging in affiliation seek to increase. It is a general variable that can be associated to hits, pageviews, unique visitors, or real human visitor revenue. In the two latter cases, defining the resource becomes more complex as it may require to define a *reader's behavior*, possibly with reader agents.

7.2.4 Quality

Quality is the other objective of CWs. It defines the relevance and richness of information itself. Quality can be a ranking based on human classification, usage information, connectivity, or non-affiliated experts [Bharat and Mihaila, 2002]. To simplify this notion we consider quality being equivalent to a page ranking, be it in the system itself by different users, or a public page rank. In the experiment featured in the present chapter, quality is emulated as a logarithmic function of the visitor revenue, as it can be done approximately for search engines' public page rankings.

7.3 Society of Community Websites

7.3.1 Matchmaking and Brokering Architecture

We define a Matchmaking and Brokering Architecture (MBA) (see Fig. 7.1) fairly similar to that of [Decker et al., 1996], with a Requester, a Broker (or Server), and a Matchmaker. However, in simulation, we choose not to

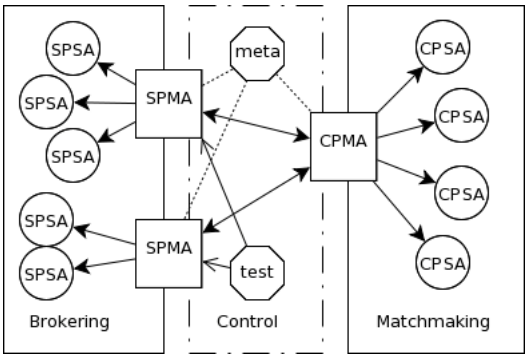

Figure 7.1: System Architecture

implement the Requester as a single-threaded agent such as in [Woolridge and Jennings, 1995].

It is unrealistic that the Requester, which we describe in our architecture as the Community Website Agent, should own its own thread running indefinitely on a machine. To simulate intelligent agents and real-time, processing an array of state machines [Woolridge, 2001] in a loop is sufficient. On each pass, the unit's object is checked for its state and an action course to decide.

7.3.2 Community Website Agent Layer

The CW Agent consists of two parts:

1. The user, holding expectations, and will to request.

2. The knowledge base about the CW. It is stored in a Site Pool Slave Agent (Sect. 7.3.3).

In practical application, the Requester is indeed a human agent. In simulation, we assimilate them as the same entity: incentives and knowledge are both stored and processed in the Site Pool Layer. Whichever the case, the difference between our architecture and past data mining agent architectures such as [Kargupta et al., 1997] is that the data needs not be accessed by external agents which will generate a lot of read accesses and make each request; that part of processing is saved since it is done by the slave database agents themselves (see following section). Since every affiliation request would have to generate a write access either way, to make sure data is up-to-date before

sending the request, we use this access to set up a flag for request, and make no unneeded read accesses.

7.3.3 Brokering: Site Pool Layer

The Site Pool Slave Agents (SPSAs) are agents with a knowledge base about a number of sites. Since every site in each SPSA is independent from the other, it becomes scalable as horizontal sharding . Furthermore, the Platform [Fukuta et al., 2001] we use for the simulator allows to easily spawn and duplicate agents over networked computers. An infinite loop runs on the SPSAs to simulate virtual Requesters (Sect. 7.3.1). Once an incentive is set to 1 for one of the sites in the SPSA's array, the site will be part of the next joint request of the current iteration in the SPSA, to the SPMA.

The Site Pool Master Agents (SPMAs) provide control over one or several SPSAs, as well as service or connection to the user's Web interface in practical application. The SPMAs take requests from users, update the incentive state of the site in the concerned SPSA, then get a joint request from the SPSAs that they will split and submit to the Category Pool Layer (Sect. 7.3.4) (CPL). They get a response and transmit it to the user. Upon affiliation agreement, they update data again in the SPSAs and submit new placement data (Sect. 7.4.2) to the CPL. The internal state agents make the final decision and update.

7.3.4 Matchmaking: Category Pool Layer

The Category Pool Slave Agents (CPSAs) hold data similar to that contained in SPSAs, but the sharding is done by category (Sect. 7.4.2). They receive requests transmitted by SPMAs to Category Pool Master Agents (CPMAs) from the latter, and are the ones who run the matchmaking algorithm. CPSAs also update their data every N iterations. For example, if an iteration is a day, it is sufficient to update every month. Upon an affiliation agreement, they update the placement data on another request from SPMAs.

The Category Pool Master Agents (CPMAs) are used by the SPMAs to find new affiliation opportunities for the requester site. They control the CPSAs.

7.3.5 Control Layer

The Meta Agent acts as a directory of all Master Agents in the system so that brokers can find matchmakers.

The Test Coordinator Agent monitors the runs in the simulation and collects data from SPSAs.

7.3.6 Interface.

Agents in the brokering and matchmaking layers share a common inter-agent request interface, for CW and blogging platforms (ex: Blogger, Wordpress) to provide easily with the service to their users. Matchmakers can be independent as well (ex: Blogcatalog). This interface can be seen as a transparent web service.

7.4 Model

7.4.1 User Load Reduction Hypothesis

The most common pattern in the affiliation process goes as follows:

1. Incentive of looking for a Potential Affiliate (PA)
2. Actual search for a PA
   (a) Decide for a tool: search engine, directory, contacts, social networks.
   (b) Find PA with related topics.
   (c) Judge the quality of the PA.
   (d) Match the PA's number of visitors against personal expectation.
3. Negotiations for equity on both sides.
   (a) Find the right contact for affiliation procedures if there are several.
   (b) Quality evaluation from the PA.
   (c) Popularity evaluation from the PA.

(d) Consider placement of each one's link on the other's page.

4. Final agreement.

Whereas only the following steps should be needed, as in our system:

1. Incentive of looking for a Potential Affiliate (PA).
2. Define expected quality and/or visitor revenue from partner.
3. Go through output of the system, make contact immediately.
4. Final agreement.

Therefore, we assume that the system significantly reduces user's load in the search for affiliates and the following negotiations, if it can succeed in simulation when configured with proper initial data and heuristics about the target environment.

7.4.2 Knowledge Base

This is the knowledge base contained in every SPSA about each site.

7.4.3 Placement

The space available to put each other's link on the page, as well as its position and format, is an important matter. There exists no general rule to determine which placement is best on a web page in general: it is influenced by presentation, as well as the number of affiliates already present, and numerous other factors.

For placement of the link, we use a set of probabilities (H) of being accessed from the affiliate, which is independent from the position, format, space, presentation, number of pages the link is to be shown on. In practical application, either the users can fix the values in their H, or this can be done automatically [Nakamura et al., 2009a].

7.4.4 Matchmaking Algorithm

The matchmaking layer receives all the data about the site from the brokering layer. The CPMA dispatches the requester site's data to the appropriate CPSAs' queues, merge the result arrays and send response to the SPMA. We name the utility function u.

Table 7.1: Knowledge base

S	The set of all sites in the system.
C	The set of all categories.
$\forall s \in S, s = (i, c, e, v, q, H_s, a)$	
i	s's identifier (URI)
$c \in C$	s's category, a set of keywords related to a similar general topic
$e \in]-\infty, +\infty[$	s's expectation of fairness
$v \in [0, +\infty[$	s's visitor revenue per iteration
$q \in [0, 10]$	s's quality rank
$H_s / \sum_{h \in H} h \leq 1$	The placement set
$h \in H_s \Rightarrow h \in [0, 1]$	Value associated to an available placement only
$a \in [0, iterations_{run}]$	Activity rate, an average period in Iterations after which s looks for new affiliates

$deals_s \leftarrow empty\ array$
for all $s' \in CPSA$ **do**
  $deals_{s,s'} \leftarrow empty\ array$
  $utility_{s'} \leftarrow 0$
  for all $h' \in H'$ **do**
    for all $h \in H$ **do**
      $u \leftarrow h' \times v' \times q' - h \times v \times q$
      if $e \leq u \leq -e'$ **then**
        $INSERT(deals_{s,s'}, [s', u, h', h])$
        $utility_{s'} \leftarrow utility_{s'} + u$
      end if
    end for
  end for
  if $deals_{s,s'} \neq empty\ array$ **then**
    $INSERT(deals_s, [deals_{s,s'}, utility_{s'}])$
  end if
end for
$SORT(deals_s, utilityx, desc)$
$RETURN deals_s$

Figure 7.2: Matchmaking algorithm

Table 7.2: Experiment parameters

Iterations	1500
S	500 blogs
C	20 categories, affected randomly
e	0 (mean), 100 (deviation) (signed Gaussian generation)
$v_{t=0}$	100 (mean), 1000 (deviation) (unsigned Gaussian generation)
q	Logarithmic function of v, interpolated from real page ranks: $q(0) = 0, q(500) = 1, q(1000) = 2, q(3000) = 3, ..., q(729000) = 8, ...$
H	Available slots in blogroll placed at half-height, being placed on a lower slot decreases exponentially click probability
a	Random variable: average of one request per blog in 10 iterations
Agents	1 meta agent, 1 SPMA, 8 SPSA, 1 CPMA, 8 CPSA, 1 test coordinator

7.5 Simulation and Sample Results

In this simulation we show the evolution of a young blog community network.

7.5.1 Initial Data and Heuristics

Since the goal of our system is to reduce user's load, we expect at least a behavior similar to that of a real blog community: when compared to a network where no affiliation occurs, overall inequality in visitor revenue as well as quality (here, a page ranking, simplified model and function of v), should shrink. Overall visitor revenue should increase.

7.5.2 Results

We made two runs of the system, one without affiliation process at all (first run, Fig. 7.3 and 7.4), and one with each blog requesting periodically for an affiliation (second run, Fig. 7.5 and 7.6). Each graph shows VR per Iteration

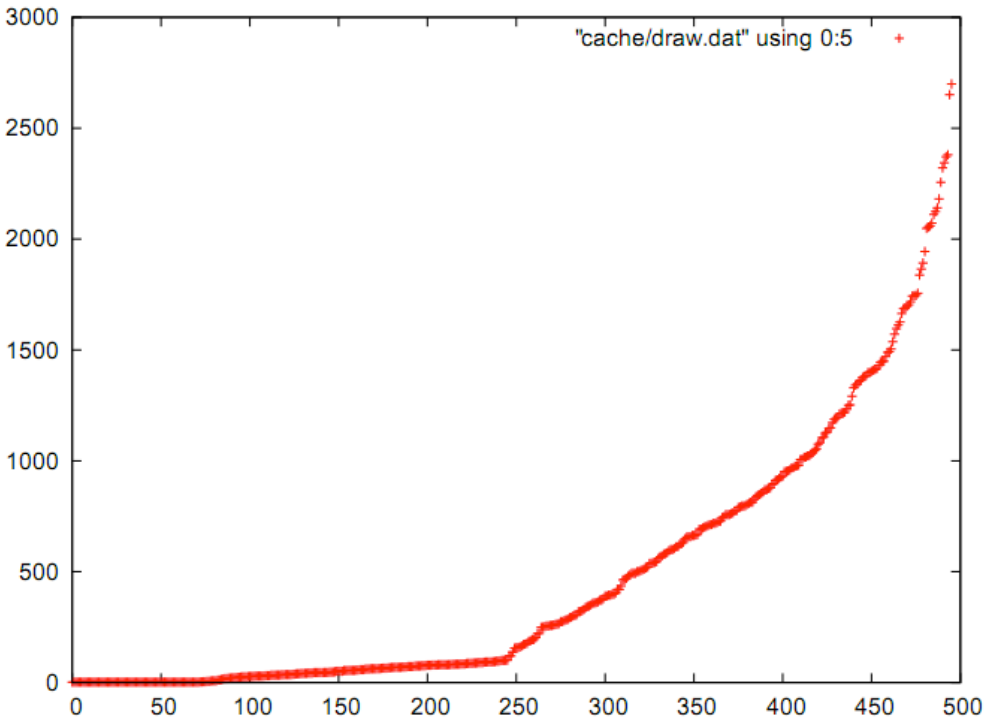

Figure 7.3: No Affiliation, t=1

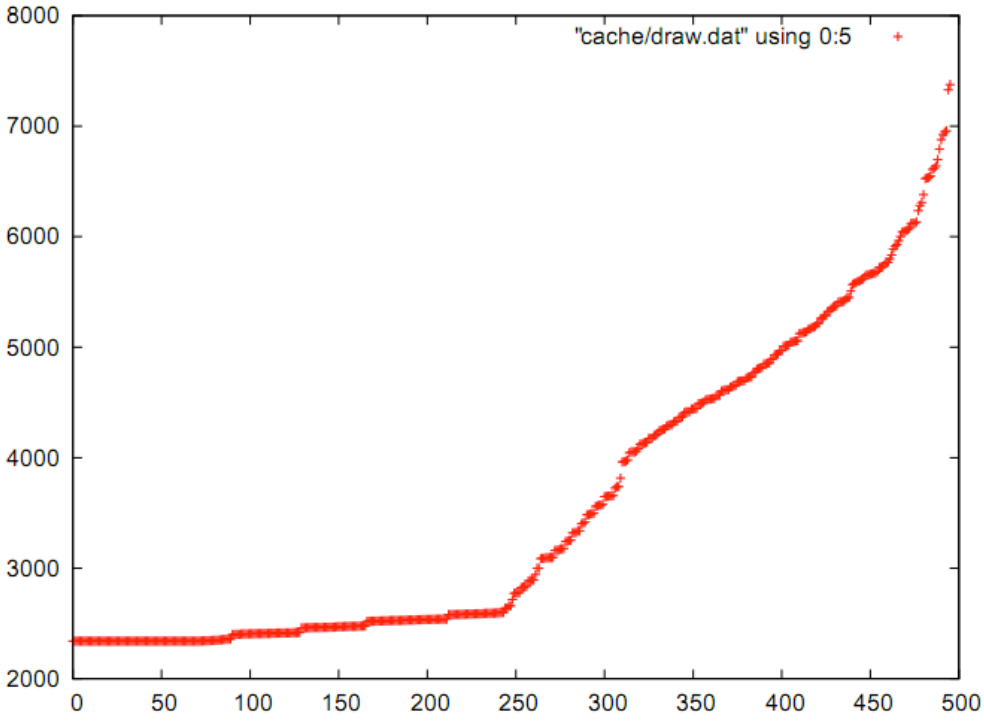

Figure 7.4: No Affiliation, t=1500

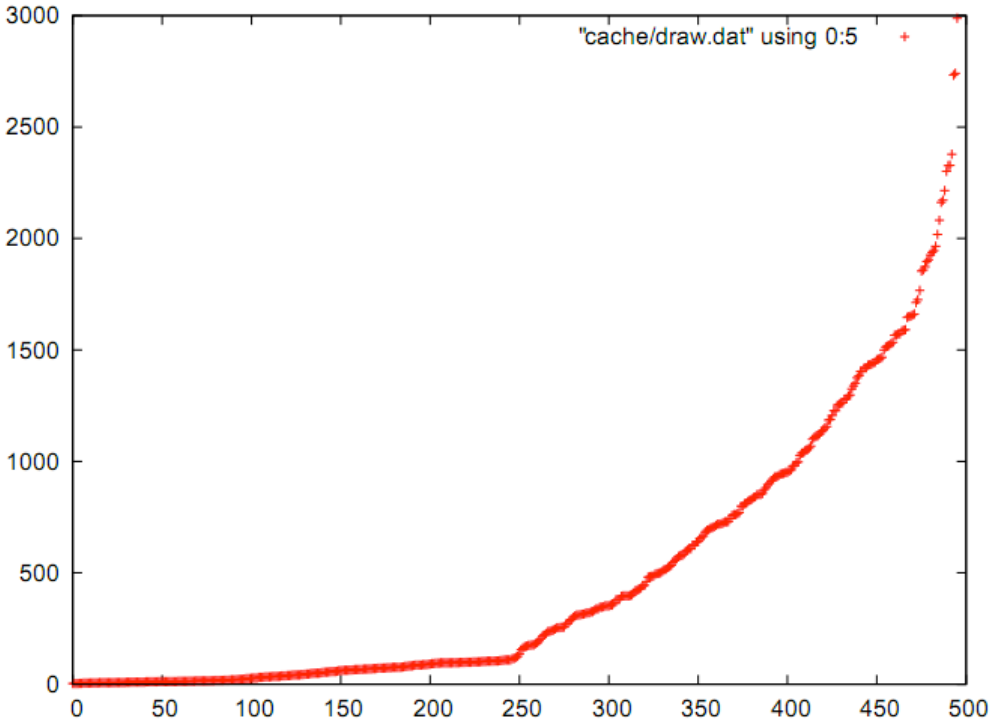

Figure 7.5: Affiliation, t=1

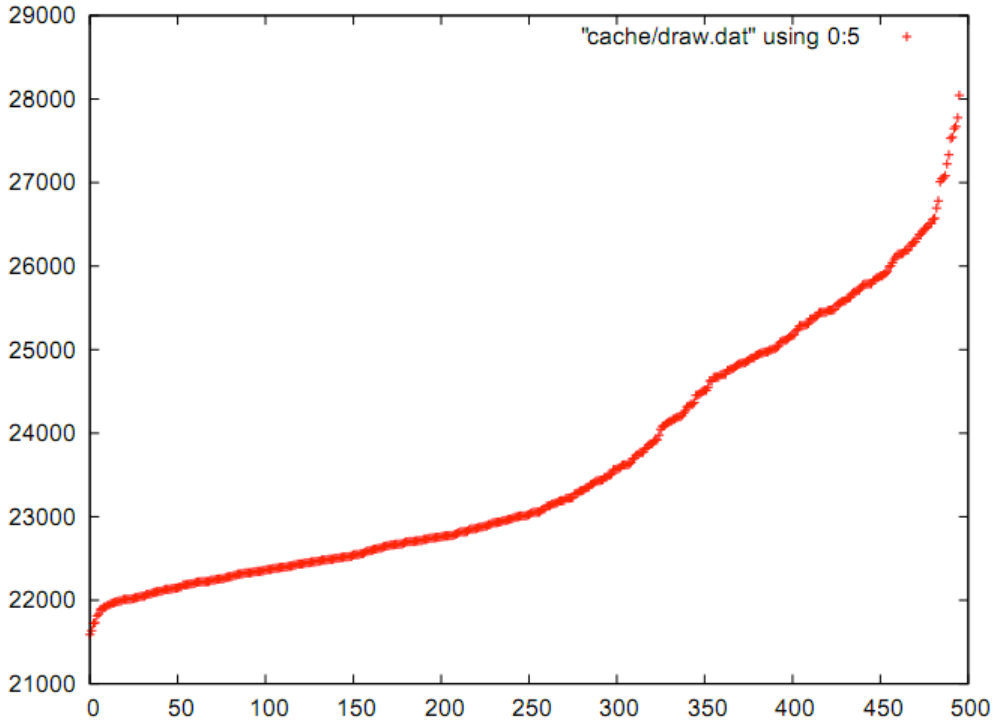

Figure 7.6: Affiliation, t=1500

(VRpI) of a blog in function of the blog id (500 ids). The ids are ordered ascendantly in function of the VRpI so that the graph can be easily read.

In the first run, blogs evolve independently and top quality blogs (right edge of the curve) reach almost 400% of the VRpI of non-popular blogs (left edge of the curve) with a VRpI of 8000 to 2500, with no exchanged visitors. Since there is no cooperation and the rich get richer, the shape of the curve is a long tail. In the second run, blogs help each others. This time, the VRpI of top blogs (right edge of the curve) is only 132% of the VRpI of least popular blogs, with a VRpI of 28000 to 22000. We can see, as expected, a huge overall increase in VR, thanks to the exchange of visitors coming from affiliation between the blogs. Therefore, the system meets our expectations in this sample experiment, it has managed in reducing VR inequality in a community of blogs.

What is interesting is also that we have verified in simulation the results of the real-world BlogDex study [Marlow, 2004b], since the shape of our curve remains a long tail. Even if overall inequality has decreased, the rich still tend on getting richer. Encouraging change of this behavior will be the subject of a further chapter.

7.6 Conclusion

In this chapter, we have developed an architecture for a system capable of relieving the user of the load of searching and negotiating in the process of affiliation in a network of community websites. Our simulator, of which we have given sample results, makes us able to easily see the evolution of the blogs/sites in a network. Using such a system, users become able to target easily their affiliates and focus more on their writing and contents, and increase the quality of their sites.

We consider adding fair counterparts to affiliation contracts, for sites with other advantages than visitor revenue and quality. Also, other heuristics than public page ranks, such as the Eigen Rumor Algorithm [Fujimura et al., 2005], can be input in the simulator for our system. We also intend on testing other patterns of communities after harvesting more data. We will discuss the subject of quality, influence and authority more precisely in a further chapter.

In another chapter [Nakamura et al., 2009b], we have been developing a system capable of fetching detailed statistics about the real visitor revenue,

the click probabilities (see H in Sect. 7.4.3), and push advertisement links automatically. We consider plugging it on the present system, for affiliation. This will also be the subject of a chapter to come. Moreover, as the use of trackback and similar tools broadens on the Internet, it may prove useful to extend the system to the research and negotiation for trackbacks and references on article pages.

Finally, as of now, the system is not capable of finding sites which are exterior to it. This feature can be developed by setting up brokers/providers that will, instead of requiring registration, crawl the Web. As well, for keywords and categories, recent advances in relational learning [Tang and Liu, 2009] could be applied to communities, in order to sharpen the different fields. We are considering this research as well.

Acknowledgements

This research project would not have been possible without the support of many people.

The author wishes to express his gratitude to his supervisor, Prof. Dr. Shintani who was abundantly helpful and offered invaluable assistance, support and guidance. Deepest gratitude is expressed as well to the members of the supervisory committee and to Associate Prof. Dr. Ozono and Assistant Prof. Dr. Shiramatsu without whose knowledge and assistance this study would not have been successful.

The author would also like to thank his graduate friends, especially his partners from the SCOPE / Open Opinion research and development group, Norifumi Hirata and Hiroyuki Sano, as well as graduate friends: Masato Nakamura, Taiki Ito, Hiroaki Kakimoto, Toshimasa Kawai; Tatiana Zidrasco, Daiki Higashiguchi, Jun Takasaki, Tomotaka Tsujino, Kimihiro Kudo, Kenta Kato, Ken Shimizu; Tatsuya Doi, Ryoji Suzuki; Yusuke Niwa, Hiroyuki Yamada, Kenshiro Buma, Shota Itokawa, Shota Imai, Motonori Koizumi, and all other students; for help with the laboratory, literature, research, development, and day-to-day assistance. Warmest thanks also to 2008 alumni Shouhei Asami, Masaya Eki, Yusuke Kondo and 2007 alumna Yuki Taki. Warmest gratitude is also directed to interns Mahmoud Salim Bouyahyaoui and Matthieu Demus from EFREI. The author would also like to convey thanks to the Faculty of Nagoya Institute of Technology for providing the financial means, support and laboratory facilities, as well as to the International Support Center staff members Mrs. Kimiko Kondo, Rie Imai, Kaori Nagae, Chitose Arai and Mariko Wada for their day-to-day support.

Also, the author would like to particularly thank the Rotary Yoneyama Memorial Foundation and all the members of the Nagoya Moriyama Rotary Club, who have been of great help and support.

The author wishes to express her love and gratitude to his beloved families;

for their understanding endless love, through the duration of this work. His parents Judd & Annie, as well as his aunt Michele and uncle Jean-Louis and the rest of the family. Final and warmest thanks to his best friends who have always been there as well as the French and international community of Nagoya for their love and support in everyday life.

This work was supported and promoted by the Strategic Information and Communications R&D Promotion Programme (SCOPE), Ministry of Internal Affairs and Communications, Japan.

List of Tables

List of Figures

Bibliography

[Anastácio et al., 2009] Anastácio, I., Martins, B., and Calado, P. (2009). Classifying documents according to locational relevance. In *Proceedings of the 14th Portuguese Conference on Artificial Intelligence: Progress in Artificial Intelligence*, EPIA '09, pages 598–609, Berlin, Heidelberg. Springer-Verlag.

[Arrivals et al., 2005] Arrivals, N., Friends, M., and People, A. (2005). e-participation and governance: Widening the net. *Electronic Journal of E-government*, 3(1):39–48.

[Bharat and Mihaila, 2002] Bharat, K. and Mihaila, G. (2002). When experts agree: using non-affiliated experts to rank popular topics. *ACM Transactions on Information Systems (TOIS)*, 20(1):47–58.

[Blei et al., 2003] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

[Bollen et al., 2011] Bollen, J., Mao, H., and Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of Computational Science*.

[Cannon and Wohlstadter, 2010] Cannon, B. and Wohlstadter, E. (2010). Automated object persistence for javascript. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 191–200, New York, NY, USA. ACM.

[Cardon et al., 2007] Cardon, D., Delaunay-Teterel, H., Cédric, F., and Prieur, C. (2007). Sociological Typology of Personal Blogs. In *International Conference on Weblogs and Social Media, Boulder, Colorado, USA Available at http://www. icwsm. org/papers/paper43. html (accessed on April 27, 2007)*.

[Chawla et al., 2002] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357.

[Chen et al., 2010] Chen, J., Nairn, R., Nelson, L., Bernstein, M., and Chi, E. (2010). Short and tweet: experiments on recommending content from information streams. In *CHI '10: Proceedings of the 28th international conference on Human factors in computing systems*, pages 1185–1194, New York, NY, USA. ACM.

[Decker et al., 1996] Decker, K., Williamson, M., and Sycara, K. (1996). Matchmaking and brokering. In *Proceedings of the Second International Conference on Multi-Agent Systems (ICMAS-96)*, page 432. Citeseer.

[Della Porta and Diani, 2006] Della Porta, D. and Diani, M. (1999, 2006). *Social movements: An introduction*. Wiley-Blackwell.

[Demichelis et al., 2006] Demichelis, F., Magni, P., Piergiorgi, P., Rubin, M., and Bellazzi, R. (2006). A hierarchical naive bayes model for handling sample heterogeneity in classification problems: an application to tissue microarrays. *BMC Bioinformatics*, 7(1):514.

[Dessalles, 1998] Dessalles, J. (1998). Altruism, status and the origin of relevance. *Approaches to the Evolution of Language*, pages 130–147.

[Dumais and Chen, 2000] Dumais, S. and Chen, H. (2000). Hierarchical classification of web content. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '00, pages 256–263, New York, NY, USA. ACM.

[Estabrooks et al., 2004] Estabrooks, A., Jo, T., and Japkowicz, N. (2004). A multiple resampling method for learning from imbalanced data sets. *Computational Intelligence*, 20(1):18–36.

[Faridani et al., 2010] Faridani, S., Bitton, E., Ryokai, K., and Goldberg, K. (2010). Opinion space: a scalable tool for browsing online comments. In *Proceedings of the 28th international conference on Human factors in computing systems*, CHI '10, pages 1175–1184, New York, NY, USA. ACM.

[for Public Administration et al., 2010] for Public Administration, U. N. D., Management, D., of Economic, U. N. D., and Affairs, S. (2010). *United*

Nations E-government Survey: Leveraging E-government at a Time of Financial and Economic Crisis. United Nations.

[Freschi et al., 2009] Freschi, A., Medaglia, R., and Nørbjerg, J. (2009). A Tale of Six Countries: eParticipation Research from an Administration and Political Perspective. *Electronic Participation*, pages 36–45.

[Freund and Schapire, 1999] Freund, Y. and Schapire, R. E. (1999). A short introduction to boosting. In *In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 1401–1406. Morgan Kaufmann.

[Fujimura et al., 2005] Fujimura, Ko, and Tanimoto (2005). Ranking Weblogs by Eigen Rumor Algorithm. *Shakai Joho Shisutemugaku Shinpojiumu Gakujutsu Koen Ronbunshu*, 11:67–72.

[Fukuta et al., 2001] Fukuta, N., Ito, T., and Shintani, T. (2001). A logic-based framework for mobile intelligent information agents. In *the Proc. of WWW10*, pages 58–59. Citeseer.

[Greengard, 2011] Greengard, S. (2011). Living in a digital world. *Commun. ACM*, 54:17–19.

[Hastie et al., 2003] Hastie, T., Tibshirani, R., and Friedman, J. H. (2003). *The Elements of Statistical Learning*. Springer, corrected edition.

[Hirata et al., 2012] Hirata, N., Sano, H., Swezey, R., Shiramatsu, S., Ozono, T., and Shintani, T. (2012). 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)*.

[Hoens and Chawla, 2012] Hoens, T. R. and Chawla, N. V. (2012). Learning in non-stationary environments with class imbalance. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, pages 168–176, New York, NY, USA. ACM.

[Hurwitz and Mallery, 1995] Hurwitz, R. and Mallery, J. (1995). The Open Meeting: A Web-based system for conferencing and collaboration. In *Proceedings of the Fourth International Conference on The World-Wide Web*. Citeseer.

[Japkowicz and Stephen, 2002] Japkowicz, N. and Stephen, S. (2002). The class imbalance problem: A systematic study. *Intell. Data Anal.*, 6(5):429–449.

[Kargupta et al., 1997] Kargupta, H., Hamzaoglu, I., and Stafford, B. (1997). Scalable, distributed data mining using an agent based architecture. In *Proceedings the Third International Conference on the Knowledge Discovery and Data Mining, AAAI Press, Menlo Park, California*, pages 211–214. Citeseer.

[Klein, 2011] Klein, M. (2011). The mit deliberatorium: Enabling large-scale deliberation about complex systemic problems. In *Collaboration Technologies and Systems (CTS), 2011 International Conference on*, page 161.

[Lahaie et al., 2007] Lahaie, S., Pennock, D., Saberi, A., and Vohra, R. (2007). Sponsored search auctions. *Algorithmic Game Theory*, pages 699–716.

[Lamport, 1974] Lamport, L. (1974). A new solution of dijkstra's concurrent programming problem. *Commun. ACM*, 17:453–455.

[Langseth and Nielsen, 2006] Langseth, H. and Nielsen, T. D. (2006). Classification using hierarchical naive bayes models. *Mach. Learn.*, 63:135–159.

[Lee et al., 2008] Lee, K., Croft, W., and Allan, J. (2008). A cluster-based resampling method for pseudo-relevance feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 235–242. ACM.

[Lichtenwalter and Chawla, 2010] Lichtenwalter, R. N. and Chawla, N. V. (2010). Adaptive methods for classification in arbitrarily imbalanced and drifting data streams. In *Proceedings of the 13th Pacific-Asia international conference on Knowledge discovery and data mining: new frontiers in applied data mining*, PAKDD'09, pages 53–75, Berlin, Heidelberg. Springer-Verlag.

[Liddo and Shum, 2010] Liddo, A. D. and Shum, S. B. (2010). Cohere: A prototype for contested collective intelligence. In *ACM Computer Supported Cooperative Work (CSCW 2010) - Workshop: Collective Intelligence In Organizations - Toward a Research Agenda.*

[Lieberman and Samet, 2012] Lieberman, M. D. and Samet, H. (2012). Adaptive context features for toponym resolution in streaming news. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '12, pages 731–740, New York, NY, USA. ACM.

[Liu et al., 2009] Liu, X., Wu, J., and Zhou, Z. (2009). Exploratory undersampling for class-imbalance learning. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 39(2):539–550.

[Macintosh, 2004] Macintosh, A. (2004). Characterizing e-participation in policy-making. In *System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on*, pages 10–pp. IEEE.

[Macintosh et al., 2005] Macintosh, A., Coleman, S., and Lalljee, M. (2005). E-methods for public engagement: helping local authorities communicate with citizens. *Published by Bristol City Council for The Local eDemocracy National Project*.

[Macintosh et al., 2009] Macintosh, A., Gordon, T. F., and Renton, A. (2009). Providing Argument Support for E-Participation. *Journal of Information Technology and Politics*, 6(1):43–59.

[MacLean et al., 1991] MacLean, A., Young, R. M., Bellotti, V. M. E., and Moran, T. P. (1991). Questions, options, and criteria: elements of design space analysis. *Hum.-Comput. Interact.*, 6(3):201–250.

[Marlow, 2004a] Marlow, C. (2004a). Audience, structure and authority in the weblog community. In *International Communication Association Conference, May, 2004, New Orleans, LA*. Citeseer.

[Marlow, 2004b] Marlow, C. (2004b). Audience, structure and authority in the weblog community. In *International Communication Association Conference, May, 2004, New Orleans, LA*. Citeseer.

[Nakamura et al., 2009a] Nakamura, M., Asami, S., Ozono, T., and Shintani, T. (2009a). A Dynamic Rearrangement Mechanism of Web Page Layouts Using Web Agents. In *Proceedings of the 22nd International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems: Next-Generation Applied Intelligence*, page 643. Springer.

[Nakamura et al., 2009b] Nakamura, M., Asami, S., Ozono, T., and Shintani, T. (2009b). A Dynamic Rearrangement Mechanism of Web Page Layouts Using Web Agents. In *Proceedings of the 22nd International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems: Next-Generation Applied Intelligence*, page 643. Springer.

[Peterson, 1981] Peterson, G. (1981). Myths about the mutual exclusion problem. *Information Processing Letters*, 12(3):115–116.

[Phang and Kankanhalli, 2008] Phang, C. W. and Kankanhalli, A. (2008). A framework of ict exploitation for e-participation initiatives. *Commun. ACM*, 51(12):128–132.

[Phelan et al., 2009] Phelan, O., McCarthy, K., and Smyth, B. (2009). Using twitter to recommend real-time topical news. In *RecSys '09: Proceedings of the third ACM conference on Recommender systems*, pages 385–388, New York, NY, USA. ACM.

[Rennie et al., 2003] Rennie, J. D. M., Teevan, J., and Karger, D. R. (2003). Tackling the Poor Assumptions of Naive Bayes Text Classifiers. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 616–623.

[Shun Shiramatsu and Okuno, 2010] Shun Shiramatsu, Jun Takasaki, T. Z. T. O. T. S. and Okuno, H. G. (2010). System for supporting web-based public debate using transcripts of face-to-face meeting. In *Trends in Applied Intelligent Systems, Proceedings of the 23rd. International Conference on Industrial Engineering and Other Applications of Applied Intelligence Systems (IEA/AIE 2010), Part III*, volume 6098 of *Lecture Notes in Computer Science*, pages 311–320. Springer.

[Swezey et al., 2009] Swezey, R., Nakamura, M., Shiramatsu, S., Ozono, T., and Shintani, T. (2009). Intelligent and Cooperative Blog Communities. In *Proceedings of the 8th Forum on Information Technology*, page 2. IPSJ.

[Swezey et al., 2010] Swezey, R., Nakamura, M., Shiramatsu, S., Ozono, T., and Shintani, T. (2010). Architecture for automated search and negotiation in affiliation among community websites and blogs. In García-Pedrajas, N., Herrera, F., Fyfe, C., Benítez, J., and Ali, M., editors, *Trends in Applied Intelligent Systems*, volume 6097 of *Lecture Notes in Computer*

Science, pages 535–544. Springer Berlin / Heidelberg. 10.1007/978-3-642-13025-$0_5$5.

[Swezey et al., 2012a] Swezey, R., Sano, H., Hirata, N., Shiramatsu, S., Ozono, T., and Shintani, T. (2012a). An e-participation support system for regional communities based on linked open data, classification and clustering. In Press, I. C., editor, *Proceedings of the 11th IEEE International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC 2012)*.

[Swezey et al., 2011] Swezey, R., Shiramatsu, S., Ozono, T., and Shintani, T. (2011). Intelligent page recommender agents: Real-time content delivery for articles and pages related to similar topics. In Mehrotra, K., Mohan, C., Oh, J., Varshney, P., and Ali, M., editors, *Modern Approaches in Applied Intelligence*, volume 6704 of *Lecture Notes in Computer Science*, pages 173–182. Springer Berlin / Heidelberg. 10.1007/978-3-642-21827-$9_1$8.

[Swezey et al., 2012b] Swezey, R., Shiramatsu, S., Ozono, T., and Shintani, T. (2012b). An improvement for naive bayes text classification applied to online imbalanced crowdsourced corpuses. In Ding, W., Jiang, H., Ali, M., and Li, M., editors, *Modern Advances in Intelligent Systems and Tools*, volume 431 of *Studies in Computational Intelligence*, pages 147–152. Springer Berlin / Heidelberg. 10.1007/978-3-642-30732-$4_1$9.

[Tahir et al., 2012] Tahir, M. A., Kittler, J., and Bouridane, A. (2012). Multilabel classification using heterogeneous ensemble of multi-label classifiers. *Pattern Recogn. Lett.*, 33(5):513–523.

[Takeda, 2007] Takeda, H. (2007). Evolution of the Web, Agents, and Semantic Web. *IPSJ Magazine*, 48(3).

[Tang and Liu, 2009] Tang, L. and Liu, H. (2009). Relational learning via latent social dimensions. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 817–826. ACM.

[Toutanova et al., 2001] Toutanova, K., Chen, F., Popat, K., and Hofmann, T. (2001). Text classification in a hierarchical mixture model for small training sets. In *Proceedings of the tenth international conference on Information and knowledge management*, CIKM '01, pages 105–113, New York, NY, USA. ACM.

[Velayathan and Yamada, 2006] Velayathan, G. and Yamada, S. (2006). Behavior-based web page evaluation. In *Proceedings of the 2006 IEEE/WIC/ACM international conference on Web Intelligence and Intelligent Agent Technology*, pages 409–412. IEEE Computer Society.

[Wang et al., 2013] Wang, S., Li, D., Zhao, L., and Zhang, J. (2013). Sample cutting method for imbalanced text sentiment classification based on brc. *Know.-Based Syst.*, 37:451–461.

[Wimmer, 2007] Wimmer, M. A. (2007). Ontology for an e-participation virtual resource centre. In *Proceedings of the 1st international conference on Theory and practice of electronic governance*, ICEGOV '07, pages 89–98, New York, NY, USA. ACM.

[Woolridge, 2001] Woolridge, M. (2001). *Introduction to Multiagent Systems.* John Wiley & Sons, Inc, USA.

[Woolridge and Jennings, 1995] Woolridge, M. and Jennings, N. (1995). *Intelligent Agents: Theory and Practice.* Cambridge University Press.

[Yen and Lee, 2006] Yen, S.-J. and Lee, Y.-S. (2006). Cluster-based sampling approaches to imbalanced data distributions. In *Proceedings of the 8th international conference on Data Warehousing and Knowledge Discovery*, DaWaK'06, pages 427–436, Berlin, Heidelberg. Springer-Verlag.

[Zhang, 2004] Zhang, H. (2004). The optimality of naive bayes. In Barr, V. and Markov, Z., editors, *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference (FLAIRS 2004)*. AAAI Press.

[Zheleva et al., 2009] Zheleva, E., Sharara, H., and Getoor, L. (2009). Co-evolution of social and affiliation networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1007–1016. ACM New York, NY, USA.

[Zhou et al., 2012] Zhou, T., Tao, D., and Wu, X. (2012). Compressed labeling on distilled labelsets for multi-label learning. *Mach. Learn.*, 88(1-2):69–126.