Electronic Preprint for Journal of Information Processing Vol.21 No.3

Regular Paper

An Improved Classification Strategy for Filtering Relevant Tweets Using Bag-of-Word Classifiers

Muhammad Asif Hossain Khan[1,a)] Masayuki Iwai[2,b)] Kaoru Sezaki[3,c)]

Received: October 29, 2012, Accepted: March 1, 2013

Abstract: In this paper we have presented a classification framework for classifying tweets relevant to some specific target sectors. Due to the imposed length restriction on an individual tweet, tweet classification faces some additional challenges which are not present in most other short text classification problems, needless to say in classification of standard written text. Hence, bag-of-word classifiers, which have been successfully leveraged for text classification in other domains, fail to achieve a similar level of accuracy in classifying tweets. In this paper, we have proposed a collocation feature selection algorithm for tweet classification. Moreover, we have proposed a strategy, built on our selected collocation features, for identifying and removing *confounding outliers* from a training set. An Evaluation on two real world datasets shows that the proposed model yields a better accuracy than the unigram model, uni-bigram model and also a partially supervised topic model on two different classification tasks.

Keywords: short text classification, microblog analysis, tweet filtering, bag-of-word classifiers, social networks

1. Introduction

Twitter, a microblogging site, which also falls under the broad category of social networks, has grown at a staggering rate since its advent in 2006. It is no longer restricted as a mere virtual environment for social communication among friends; rather it has become a platform for diversified usage like posting personal status updates, sharing multimedia contents, propagating news and even advertising products and services [1]. The spatial resolution of its coverage has made it a promising network for assessing the evolution and dynamics of social systems. The richness of information content of Twitter has allured many researchers over the years to study different social phenomena by trying to establish a correlation between the vast public opinion expressed in Twitter and physical events taking place in society [2], [3], [4], [5]. Starting from predicting stock indices [6], researchers have tried to predict box office hits [7], propagation of earthquake [8] and even outbreak of infection diseases in communities [9], [10], [11].

Generally these endeavors involve two major steps: a classification or filtering step followed by a prediction or inference step. In the classification step the researchers try to identify and segregate tweets relevant to the target sector they are trying to assess or predict. In the prediction step, they try to come up with some models that establish some correlation between the information content in the segregated tweets and the target sector, often by using some regression mechanism. Considering the huge traffic in Twitter, tweets relevant to a specific topic is really scarce [2]. Isolating the large volume of irrelevant tweets is one of the paramount challenges to be solved [3], [12] and is still an active research challenge.

In many occasions [2], [3], [8], [10], [13], [14] researchers have adopted *Bag-of-Word* (*BOW*) classifiers for filtering topic–relevant tweets. BOW classifiers are statistical learning algorithms that use textual features like *n–grams* (e.g., words, phrases) and their derivatives (e.g., frequencies). They have been successfully leveraged for many short text classification tasks like sentiment classification in user reviews [15], disease identification from abstracts of medical journals [16], etc. Many traditional classifiers like Naïve Bayes, Maximum Entropy, Support Vector Machine etc. fall in this category. The performances of these classifiers are often influenced by the feature selection strategy and also the quality of the training set; i.e., its representativeness to the further unseen text.

Twitter imposes 140 character length restriction on its posts which makes tweets different from most other short texts like user reviews, search snippets or journal abstracts that have been successfully classified using BOW classifiers. The length restriction forces its users to use unstructured language, non–grammatical sentences and non–dictionary vocabulary often referring to some URL containing more elaborate information regarding the topic of the post. This increases the complexity of generating a representative training set by many folds. On one hand, it gives rise to an almost infinite vocabulary and on the other hand it induces the sparseness of textual features in an individual tweet [17] – the only arsenal available to a BOW classifier. These added dimen-

[1] Graduate School of Information Science and Technology, Department of Information and Communication Engineering, The University of Tokyo, Meguro, Tokyo 153–8505, Japan
[2] School of Science and Technology for Future, Department of Information Systems and Multimedia Design, Tokyo Denki University, Adachi, Tokyo 120–8551, Japan
[3] Center for Spatial Information Science, The University of Tokyo, Meguro, Tokyo 153–8505, Japan
[a)] asif@mcl.iis.u-tokyo.ac.jp
[b)] masa@iis.u-tokyo.ac.jp
[c)] sezaki@iis.u-tokyo.ac.jp

sions of complexities naturally make tweet classification quite different from other short text classification. The aforementioned works, although could achieve impressive results, did not take any special measure to handle these complexities in the classification phase. As we shall see in the related work section, most of them adopted some well established feature selection and training set generation techniques that had been successfully deployed in some other kind of text classification job. In this paper we propose a simple but robust tweet classification strategy acknowledging two prominent limitations of tweets — a) the samples we can possibly incorporate in a training set are far from being representative due to the possibly infinite vocabulary and b) the sparseness amputates the BOW classifier's ability to discern context and connotation of words in the tweets. Hence, if not carefully annotated, the sparseness could easily confuse a BOW classifier causing ill performance.

Our main contributions are as follows:

(1) We have proposed a feature selection method, which takes into consideration the possibly infinite vocabulary in Twitter and thus discarding any collocation features that might have occurred by chance, even though they appear with relatively higher frequencies in the training set.

(2) We have also proposed a parameterized algorithm built on top of the selected collocation features for automatic detection and tweaking of noisy training instances, which we call *confounding outliers*. Evaluation results show that their removal contributes to the improvement of the classification performance.

(3) Moreover, as part of our feature selection process, we have proposed a stop–word selection strategy, which avoids including high–frequency discriminating features into the stoplist.

2. Related Work

Different researchers have adopted various techniques in tweet classification step. While some have used simply hand-picked keywords or phrases for filtering tweets [6], [11], others have sought recourse in text classifiers [2], [3], [8], [10], [13], [14].

Lampos et al. [11] collected 1,560 disease and symptom related keywords from web articles related to influenza and from Wikipedia. Using LASSO regression, they restricted the feature space and reduced the keywords from 1,560 to 97. Selecting keywords from domains like Wikipedia or online forums where no length restriction is imposed on posts, runs the risk of facing the problem of 'domain adaptation'. Hence, instead of borrowing knowledge from other domains, we have opted for selecting textual features from Twitter corpus alone. Phan et al. [16] also used Wikipedia for identifying latent topics in web snippets using LDA. However, web snippets are excerpts of web pages. Hence, the domain of textual features are similar to that of Wikipedia.

Some researchers have used different fast-filtering techniques for reducing the amount of irrelevant tweets from Twitter corpus before applying any classification technique. While some have used keyword based filtering [8], [10], [14], others have used information contained in the attached URL [3] or mention of predefined event related persons or venues [4]. Ramage et al. [17] dropped all tweets from a training set having less than a predefined number of terms. Those short tweets just added noise to the training set without contributing anything to the learning process.

Culotta [14], in his attempt to predict the intensity of influenza in New York, used four hand chosen keywords to select 206 tweets which he manually categorized into 160 +ve and 46 -ve examples. He then trained a Naïve Bayes classifier to classify further tweets. Aramaki et al. [10] made a similar effort to establish a correlation between Twitter messages and influenza epidemics in Japan. They extracted influenza related tweets using a simple word lookup of 'influenza'. They manually annotated 5,000 tweets as +ve or -ve and trained a BOW classifier to classify further unseen tweets. Sakaki et al. [8] used Twitter posts containing the keyword 'earthquake' to detect occurrences of earthquake in Japan. Like the aforementioned researchers, they too prepared a small training set by manual classification of tweets into positive and negative classes. They trained a support vector machine (SVM) classifier to classify further tweets.

Some researchers have used partially supervised learning algorithms. Use of partially supervised models have twofold advantages: in one hand, it eliminates the need for any manual labeling which is both time consuming and expensive and on the other hand it allows incorporation of a very large training set, which certainly leads to a more robust learning process. Ramage et al. [17] used a supervised topic model called LLDA, which is a labeled version of LDA [18], to map contents of Twitter feed into 4 categories namely substance (about events and ideas), social (recognizing language used towards a social end), status (denoting personal updates) and style (broader trends in language usage). They used hashtags as one of many user defined labeled dimensions (some of the other dimensions being emoticons, user references, reply, question marks etc.) and by combining these dimensions with identified latent dimensions, they tried to determine the most appropriate category for the tweet. The authors used a heuristic approach to subdivide each of these labels into 10 sub-labels and stated that it is still to be determined how best to select the number of sub-labels per label type. The 4 categories had been chosen to represent 4 different usage of language in the Twitter corpus. As the authors found, roughly all tweets that were either a reply or had some user references (@user) were assigned to "social" category, those having emoticons were grouped under "style" category and tweets that contained any hashtag had been categorized as "substance" category. In our two experiment scenarios and in many other tweet classification tasks, separation of tweets based on such usage of language is a far shot. For example, in one of our experiments, we tried to identify tweets reporting authors' self-infection to influenza. Tweets similar to '*I am having #flu :(. . . wish @cristine were here*' were not uncommon in our training set. This example tweet, which evidently is a status update, contains a user reference, an emoticon and a hashtag. Moreover, both tweets reporting self-illness and those not reporting any illness might contain user references, hashtags and emoticons.

3. Detail Problem Statement

As we have alluded in previous sections, the length restriction on tweets incurs some unique classification challenges. The two we tried to circumvent are as follows:

3.1 Sparsity

Twitter users have adapted to its imposed length restriction by using abbreviation, truncating words which they believe can be understood from context and often by referring to a more elaborate source of information through URLs. The follower-following paradigm of twitter sets a platform where the author of the tweet may assume that the target audience is already familiar with the context. A human reader essentially uses the textual features of a tweet, the connotation, the context and also his background knowledge about the topic to interpret a tweet. However, the only arsenal available to a BOW classifier are the textual features and their derivatives. In a length unrestricted document, the associated words, phrases and sentences provide required support for word sense disambiguation and development of context and connotation. However, for tweets not much support is available due to their brevity. Moreover, as the language used in Twitter is unstructured, different users use different techniques for abridging their text, making it extremely difficult to find a pattern out of it. Hence, unlike a follower, a BOW classifier does not have the required dimensions for interpreting some of the tweets correctly. Let us consider that we intend to classify tweets as flu +ve or -ve and encounter the following two tweets:

(1) "*In bed . . . listening to Theraflu . . .*"

(2) "*In bed . . . hope this Theraflu works . . .*"

In the first tweet, the user is referring to a popular music track of DJ Khaled, titled 'Theraflu'. However, in the second tweet the user is referring to a common drug used in US against seasonal flu whose brand name is also 'Theraflu'. Now the following three cases are worth mentioning, with case (ii) and (iii) being identical with role reversion:

(i) Tweets with similar textual features as (1) are not uncommon among -ve training instances and those similar to (2) are not uncommon among +ve training instances.

(ii) Tweets with similar textual features as (1) are not uncommon among -ve training instances but those similar to example (2) are extremely rare among +ve training instances.

(iii) Tweets with similar textual features as (1) are extremely rare among -ve training instances but those similar to example (2) are not uncommon among +ve training instances.

In case of (i), the performance of the classifier in finding an optimal classification boundary would be largely dependent on the feature selection strategy. The classifier would try to work around by assigning different weights to the selected features. Unigrams like 'Theraflu' or 'bed' would not play any significant role if they appear with near equal frequency among the training instances of both classes. Depending on the feature selection strategy, they actually might be dropped from feature set.

However, if (ii) is the case, then we call tweet (2) a *confounding outlier*.

Definition 1 *Confounding Outlier*: A Confounding outlier is a tweet that satisfies both the following properties:

a. It is an outlier among training instances of its own class

b. It has remarkable resemblance in terms of textual features with majority of training instance of some other class(es).

Edgar et al. [20] analyzed the effect of outlier detection on the performance of classifiers and found that the misclassification error rate decreases after removing outliers. In this paper, we have proposed a method, built on top of our feature selection procedure, for identifying and tweaking confounding outliers from a training set. Evaluation results show that, though confounding outliers comprise a very small fraction of the training set, their removal improves the classification accuracy, which agrees with the findings of Acuña et al. [20].

3.2 Loose Coupling

People post their opinion, news etc. about their domain of interest in Twitter. It is quite expected that by the passage of time, physical events would cause some of the domains come closer to the other topics. For example, the *Occupy-Wall-Street* (*OWS*) movement either intentionally or unintentionally had some influence on real estate prices, job market etc. and NY police had a busy time deterring those protesters. Hence tweets like "*Ex-Philly police captain arrested at NY Occupy rally is warned not to wear uniform at protests: http://t.co/6FmTvWOp #OWS #NYPD*" or "*Police arresting people for no apparent reason: http://t.co/Csytb3yr #OWS*" were common among our crawled tweets. Textual feature of the first tweet has several discriminating features of OWS related tweets (e.g., 'occupy', 'rally', 'protests' etc.), however the same cannot be said about the second one. A classification problem would face no challenges if these overlapping domains map to the same class. However, had two of our different classes corresponded to tweets relevant to OWS movement and those relevant to normal police activities in New York, then considering the significant proportion of police action related tweets in the OWS corpus, it seems that the classification problem would have been similar to the one explained in case (i) in the 'Sparsity' subsection. In our experiment we had two different classes, one mapping tweets from OWS domain and the other mapping those from real estate domain. The following two tweets were picked up as confounding outliers by our tweaking module:

(1) "*Euro is not to blame for crisis: Martin Webber: We have had this Occupy Wall Street movement emerging http://t.co/epNk3ejl* ***#realestate***"

(2) "*Our campaign against @REBNY's attempted park closures makes The Real Deal top real estate stories of 2011: http://t.co/J9DbOVa5* ***#ows***"

The hashtags affixed to the above tweets are not inappropriate in the sense that the events reported in the tweets actually correspond to the topics indicated by the hashtags. However, at the same time they also satisfy both the properties of confounding outliers, and hence tweaked from the training set. We shall discuss the tweaking process in detail in the next section.

4. Proposed Method

Figure 1 shows the general framework of our proposed model. 'Feature selection' and 'Tweaking' are the two key sub–modules of the framework, which are responsible for selecting textual feature and removing confounding outliers respectively. We describe their functional details in the following subsections.

4.1 Feature Selection Unit

4.1.1 Unigram Feature Selection

Let, C be the set of all classes and T_c be the set of all unigrams encountered among the training instances for class c. Let, $\mathcal{U} = \{t : t \in T_c \text{ for all } c \in C\}$.

4.1.1.1 Preprocessing Step: Let $\mathcal{U}^* \subseteq \mathcal{U}$ be the set of unigrams satisfying any of the following conditions:

- its length is less than 3
- it is a hashtag, URL or user reference (@user)
- it is a numeral or time expression

4.1.1.2 Stop-word Selection: The most common terms are effectively a corpus–specific collection of stop–words [17]. A common practice in NLP is to consider the top k terms with the highest frequency as stop–words, with typically k ranging from 30 to 50. However, selecting solely by term frequency causes content-bearing words to be added to the stoplist [21]. Hence, instead of simply selecting the terms with highest frequency, we define a new measure for selecting stop–words based on *term–frequency*, *doc–frequency* and *inclination* of each term.

Definition 2 *Term–frequency*: Number of times a term appears in a particular tweet collection. For each term t, let tf_t be a length $|C|$ vector holding its term–frequency in the training corpus of each class.

Definition 3 *Doc–frequency*: Number of tweets in a collection containing the term. For each term t, let df_t be a length $|C|$ vector holding its doc–frequency in the training corpus of each class.

Definition 4 *Inclination*: To measure the degree of inclination of a term toward the training corpus of any particular class, we define a new function γ_t:

$$\gamma_t = \frac{\|df_t\|_1 - max(df_t)}{\|df_t\|_1}$$

where, $\|.\|_1$ represents the L_1 norm. If $\frac{df_t}{\|df_t\|_1}$ is a uniform distribution, then γ_t will be equal to $\frac{|C|-1}{|C|}$, and if the term appears only in the training corpus of a single class, then γ_t will be equal to 0.

Our set of selected stop–words for unigram features:

$$\mathcal{S}_u = \{\arg\max_{t \in \mathcal{U}-\mathcal{U}^*} \|tf_t\|_1 : \gamma_t > 0.15 \text{ and } |\mathcal{S}_u| = 40\}.$$

4.1.1.3 Rare Unigram Identification: Terms appearing in very few documents in the training set are often referred to as *rare terms*. Removing rare terms from the unigram feature set is also a common practice adopted by many researchers including Ramage et al. [17]. Our selected set of rare unigrams $\mathcal{R}_u = \{t : \|df_t\|_1 < 4 \text{ and } t \in \mathcal{U} - \mathcal{U}^*\}$.

Our finally selected unigram feature set $\mathcal{F}_u = \mathcal{U} - \mathcal{U}^* - \mathcal{S}_u - \mathcal{R}_u$.

4.1.2 Bigram Feature Selection

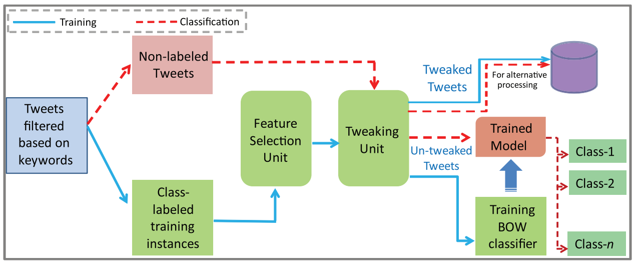


Fig. 1 General framework of the proposed model.

4.1.2.1 Selecting Unigrams for Bigram Construction: Several earlier researchers reported better results when considering unigrams of some specific part–of–speech for constructing bigrams [29], [30]. We adopted a similar approach and only unigrams in the set $\mathcal{U}_p = \{t : t \in \mathcal{U} - \mathcal{U}^* \text{ and } \mathcal{POS}(t) \in \{verb, noun, adjective\}\}$ participated in the bigram construction process, where $\mathcal{POS}(t)$ returns the part–of–speech of a unigram, which we determined using the 'Stanford Log–linear Part–Of–Speech Tagger' [27].

4.1.2.2 Capturing Bigrams with a Flexible Structure: As Twitter users often do not follow any standard grammar for their posts, we look for bigrams where the two unigrams stand in more flexible relationship to one another. Hence, instead of looking for pair of words immediately following each other, we use a 'collocation window' of 3 – thus considering each word pair in the window as a potential bigram. For example, the phrase "powerful personal computer" would produce three bigrams; 'powerful personal', 'personal computer' and 'powerful computer' when the collocation window is set to 2 or higher. As has been shown by Smadja [22], this method is quite successful at terminology extraction and determining appropriate phrases for natural language generation. Let, $\hat{\mathcal{B}}_c = \{(t_1, t_2) : t_1, t_2 \in \mathcal{U}_p \text{ and } dist(t_1, t_2) \leqslant 3\}$.

4.1.2.3 Identifying Bigrams with Structural Importance: Smadja [22] actually used a variance based method for reducing the feature space. Of course, considering all possible bigrams encountered in the training set would make the feature space extremely sparse. To determine whether the bigram has some real structural importance, we have adopted the 'Likelihood Ratio' approach for hypothesis testing of independence proposed by Dunning [23], which takes into account the volume of data that has been considered for calculating the frequency of the bigram as well as the frequency of the individual words comprising the bigram. For sparse data (as in case of Twitter) this approach is more appropriate than the χ^2 test [24].

Likelihood ratio is a number that tells how much more likely one hypothesis is over another. Our first hypothesis $\mathcal{H}_1$ states that there is no association between the words beyond chance occurrence; i.e., in the bigram w_1w_2, the words w_1 and w_2 are generated completely independently of each other. The second hypothesis $\mathcal{H}_2$ states that there is a structural dependence between w_1 and w_2. Formally,

Hypothesis 1 ($\mathcal{H}_1$): $\mathrm{P}(w_2|w_1) = p = \mathrm{P}(w_2|\neg w_1)$
Hypothesis 2 ($\mathcal{H}_2$): $\mathrm{P}(w_2|w_1) = p_1 \neq p_2 = \mathrm{P}(w_2|\neg w_1)$

We have used the usual maximum likelihood estimation for calculating p, p_1 and p_2. Let, n_1, n_2 and n_{12} be the number of occurrences of w_1, w_2 and w_1w_2 in the text corpus respectively and N be the total number of terms. The likelihood of getting the counts w_1, w_2 and w_1w_2 in the current corpus is

$$L(H_1) = bin(n_{12}; n_1, p)bin(n_2 - n_{12}; N - n_1, p) \text{ and}$$
$$L(H_2) = bin(n_{12}; n_1, p_1)bin(n_2 - n_{12}; N - n_1, p_2).$$

Here, $bin(x; n, p) = \binom{n}{x} p^x (1-p)^{n-x}$ represents binomial distribution. We then calculate the likelihood ratio $\lambda = \frac{L(\mathcal{H}_1)}{L(\mathcal{H}_2)}$ of the two hypotheses. The quantity $(-2\log\lambda)$ is asymptotically χ^2 distribution. We reject the hypothesis of independence $\mathcal{H}_1$ for a bigram with 95% confidence if $-2\log\lambda \geq 7.88$, which is the critical value for χ^2 distribution with 1-degree of freedom at confidence level $\alpha = 0.005$. Let, $\mathcal{B}_c = \{b : b \in \hat{\mathcal{B}}_c$ and b is structurally important$\}$.

4.1.2.4 Bigram Stoplist Construction: Let, $\hat{\mathcal{F}}_b = \bigcup_{c\in C} \mathcal{B}_c$. Our selected set of bigrams in the stoplist:

$$\mathcal{S}_b = \{\arg\max_{b\in\hat{\mathcal{F}}_b} \|tf_b\|_1 : \gamma_b > 0.15 \text{ and } |\mathcal{S}_b| = 10\}.$$

4.1.2.5 Rare Bigram Selection: Our selected set of rare bigrams $\mathcal{R}_b = \{t : \|df_b\|_1 < 4 \text{ and } b \in \hat{\mathcal{F}}_b\}$.

Our finally selected bigram feature set $\mathcal{F}_b = \hat{\mathcal{F}}_b - \mathcal{S}_b - \mathcal{R}_b$. We refer to this set as '*fidels*'.

4.2 Tweaking Unit

This unit is responsible for identifying and removing confounding tweets from the training set by using the selected features.

4.2.1 Determining Most Appropriate Class for Overlapping Bigrams

Ratios of relative frequencies between two or more different corpora can be used to discover collocations that are characteristics of a corpus when compared to another corpus [26]. For each bigram in $\mathcal{F}_b$, we tried to determine the class for which the bigram is more appropriate as a characteristic feature. For such bigrams, we checked the 'ratios of relative frequencies' between two or more classes to determine the most appropriate class for the bigram. Let, n_1 and n_2 be the frequencies of a bigram b in the training corpus of classes X and Y respectively. Let, N_1 and N_2 be the total number of terms identified from classes X and Y. Then the relative frequency ratio is $r = \frac{n_1/N_1}{n_2/N_2}$. If $r \geq 1$, then b's most appropriate class is X, otherwise it is Y. $\mathcal{M}_b$ is a length $|C|$ vector holding the relative frequency of *fidel* b for each class $c \in C$.

4.2.2 Identifying and Tweaking Confounding outliers from a Training Set

We have developed algorithm 1 to identify confounding outliers in a training set. For every tweet in the training corpus of a class c, we determine the number of bigrams in the tweet for which c is its most appropriate class (lines 8–10). In the algorithm $\delta \in [0, 1]$ is a design parameter, which controls the strength–accuracy tradeoff of the model. If δ is set too close to 1, each tweet in the training set will contain bigram features from exactly one class. Along with the confounding outliers, all tweets on the class boundaries will be culled. Though the classifier will show better accuracy on the training set, it will not be robust against actual tweets. Similarly, setting δ close to 0 will not tweak any tweets from the training set thus leaving the confounding outliers behind. If for less than δ fraction of the bigrams in a tweet, the bigrams' most appropriate classes are different from the class assigned to the tweet, we discard that tweet from the training set (lines 13–15). In our experiment we have used $\delta = 0.3$. Hence, a tweet is declared as a confounding outlier, if approximately more than $\frac{2}{3}$-rd of its bigram features' most appropriate classes are different from its own class. A higher value of δ would run the risk of tweaking many instances near class boundaries, which would result in over-fitting. The objective of the tweaking process is to tweak those tweets whose textual features suggest a different class than that currently assigned to it. In one of our experiments we used hashtags as class labels and the tweaking process culled 1.3% training instances. In our other experiment, we manually labeled training instances and the tweaking process culled only 0.28% tweets. It substantiates that when the training instances are carefully annotated, tweaking would be hardly necessary. Though the inter-class distances were quite marginal in the later experiment too, the tweaking process did not try to widen them by culling tweets near class boundaries.

Algorithm 1 Tweak Confounding Outliers from Training Set

```
1.  $trainingSet \leftarrow null$
2.  for each class $c \in C$ do
3.      for each tweet $\mathcal{T} \in$ training corpus of $c$ do
4.          $total \leftarrow 0$ $local \leftarrow 0$
5.          for each bigram $b \in \mathcal{T}$ do
6.              if $b \in \mathcal{F}_b$ then
7.                  $total \leftarrow total + 1$
8.                  if $\mathcal{M}_{b,c} = \max(\mathcal{M}_b)$ then
9.                      $local \leftarrow local + 1$
10.                 end if
11.             end if
12.         end for
13.         if $total = 0$ or $\frac{local}{total} \geq \delta$ then
14.             $trainingSet \leftarrow trainingSet \cup \{\mathcal{T}\}$
15.         end if
16.     end for
17. end for
```

5. Evaluation

To evaluate the performance of the proposed model, we conducted two experiments.

Experiment 1: In the first experiment we tried to identify tweets reporting some illness of the author. The algorithm learned on a small manually labeled training set. Following an approach similar to Culotta [14] and Aramaki et al. [10], we considered tweets having the keywords 'influenza' or 'flu' only and then classified those tweets into those reporting some illness (+ve class) and those not reporting any illness of the author (–ve class). We actually divided the –ve class into two sub–classes – one reporting some flu related news, e.g., those reporting outbreak of an epidemic somewhere in the world or reporting some important research finding regarding flu etc. and the other just having the keywords but not reporting any illness.

Experiment 2: In the second experiment, we tried to determine appropriate 'hashtags' for tweets. 'Hashtags' are twitter provided mechanism for self-annotation of tweets by their authors. A user can add the '#' sign before any word in his/her tweet to convert the word into a hashtag. Twitter users optionally affix hashtags to their tweets for making the topic of the tweet more explicit and also for other users with similar interest to easily track these tweets. So, one way to filter tweets relevant to a particular topic would be to look for tweets with relevant hashtags. However, a significant proportion of tweets do not contain any hashtags [28]. Hence, in this task we tried to classify un–tagged tweets to their appropriate 'hashtag class'. The use of hashtags as class labels has several advantages:

- This eliminates the need for any manual labeling of the training set; thus offers a way to avoid the cost of a large training set generation.
- A topic often begets many sub–topics. For example, the classifier trained to filter earthquake relevant tweets should be able to filter out tweets relevant to the earthquake of Japan on March 11, 2011. However, it would fail to filter most tweets relevant to the Fukushima nuclear power plant disaster, which had been a consequence of the earthquake. Again, along the course of time, tweets relevant to the Fukushima disaster also changed sub–topics. During the first few days, the discussion topics were dominated by the possible impact of tentative meltdown of some reactors. Gradually over the time topics changed from contamination of plants, animals and fishes to lack of transparency by TEPCO and so on. Designing a new classifier using manual labeling of training instances each time the sub–topic changes would incur a lot of time and cost. However, as the proposed approach of using hashtags as class labels reduces the time and cost for generating a training set to the minimum (as opposed to manual labeling of training set, which requires significant time and human labor), it offers a way for quick development of classifiers on–demand targeting sub-topics with a minimal cost.

  It is to be mentioned here that there is no specific guideline for affixing hashtags to a tweet. Hence, a topic might be covered by many hashtags. However, as authors of Refs. [8], [10], [14] used only 1 or 2 keywords for fast–filtering target–topic related tweets, we also adopted a similar approach and used 1 to 3 hashtags for each topic. Again, it is not impossible that a single hashtag is used in two completely different topic domains (equivocal hashtags). The proposed method would not be able to separate multiple topic classes sharing same hashtag.

5.1 Experiment Design

5.1.1 Twitter Data Collection

Twitter offers several APIs for crawling tweets from their servers. We have developed a Java–based crawler using Twitter's Search API and have been collecting tweets from several cities including New York. Search API allows to define a center (expressed in latitude and longitude) and a radius (expressed in kilometer) to define an area. It then returns tweets generated within that area. However, it imposes rate limitations on the number of times the API can be called per hour. Hence, the crawled tweets are only a sample of the total tweets generated within that region.

5.1.2 Models Considered for Evaluation

For both the experiments, we used the following four models:

- **Unigram Model:** Considers only unigram features.
- **Uni-Bigram Model:** Considers both unigram and bigram features.
- **Proposed model:** Considers unigram features and *fidels*.
- **Supervised Topic Model:** Uses LLDA [17] on unigram features. Class names serve as labels for the tweets.

Table 1 Term definition.

	Relevant	Non-relevant
Retrieved	True Positive (TP)	False Positive (FP)
Not Retrieved	False Negative (FN)	True Negative (TN)

All the models started with the same training set. However, every model finally considered only those training instances that contained at least 5 of its selected features. All the models discarded any term in the training set that was of less than 4 characters, a numeral, a hashtag, a URL or a user reference. The unigram model, uni-bigram model and the proposed model used the same unigram stoplist. The uni-bigram model and the proposed model shared the same bigram stoplist. However, unlike the proposed model, uni-bigram model did not filter bigrams based on part–of–speech tag of constituent terms.

5.1.3 Text Classifier

Unigram model, uni-bigram model and the proposed model used Naïve Bayes as the text classifier. We used WEKA's [25] multinomial Naïve Bayes implementation for carrying out the experiments. Despite its simplicity in its assumption of independence, Naïve Bayes often rivals and indeed outperforms more sophisticated classifiers on many datasets [24], [25]. The probability of a tweet d being in the class c is computed as:

$$P(c|d) \propto P(c) \prod_{1 \le k \le n_d} P(f_k|c)$$

where $P(f_k|c)$ is the conditional probability of feature f_k occurring in a tweet of class c and n_d is the number of features encountered in tweet d. $P(c)$ is the prior probability of a tweet occurring in class c, which is obtained through maximum likelihood estimates. The best class for tweet d is the *maximum a posteriori* (MAP) class c_{map}:

$$c_{map} = \arg\max_{c \in \mathbb{C}} \hat{P}(c|d) = \arg\max_{c \in \mathbb{C}} \hat{P}(c) \prod_{1 \le k \le n_d} \hat{P}(f_k|c)$$

The 10-fold cross validation method was adopted for assessing the classification performance.

The partially supervised topic model used Stanford Topic Modeling toolbox's [19] implementation of LLDA.

5.1.4 Evaluation Metrics

We have used precision, recall and F-measure for comparing the performances of the different models. *Precision* is the fraction of retrieved tweets that are relevant and is defined as $P = TP/(TP + FP)$. *Recall* is the fraction of relevant tweets that are retrieved and is defined as $R = TP/(TP + FN)$. The terms TP, TF, FP and FN are defined in **Table 1**. *F-measure*, also known as $F_1 - score$, is the harmonic mean of precision and recall and is a convenient way for measuring the classification performance using a single numeric value. It is defined as, $F = 2 * \frac{precision*recall}{precision+recall}$.

5.2 Experiment 1: Classifying Tweets with Self-reported Illness

5.2.1 Experiment Data

For the first task we used tweets originated in NY from December 06, 2011 to April 30, 2012. Similar to the approach of

Table 2 Per class statistics of training data.

	Self	News	False
Total Labeled Tweets	329	268	290
Tweets with inadequate features	163	181	188
Total Tweaked	0	0	1
Total Bigrams	1,856	2,171	2,459
Unique Bigrams	1,698	1,864	2,371
Bigrams with Structural Importance	233	319	326

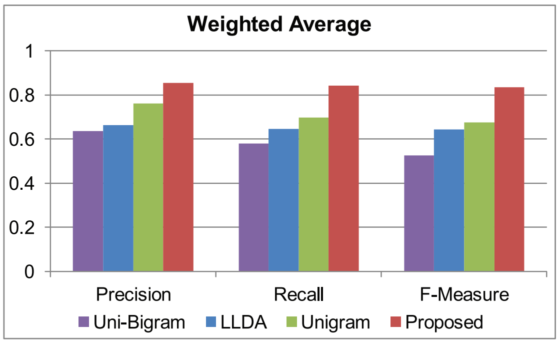


Fig. 2 Comparison among weighted average performance among the four models.

Culotta [14] and Aramaki et al. [10], we only considered tweets having either the keyword 'flu' or 'influenza'. A total of 3, 955 tweets had these keywords and we randomly selected 887 tweets for manually annotation into three classes: self (329), news (268) and false (290). Class definitions are as follows:

- **Self**: Tweets reporting self-infection. For example, *"Drinking Theraflu, trying to bit this . . . Wish someone were here taking care of me . . . "*
- **News**: Flu related news. For example, *"Chinese bus driver infected with H1N1 bird flu virus dies; countries first reported human case in 18 months"*
- **False**: Not reporting any infection. *"Don't understand why people take flu so seriously . . . "*

Table 2 shows some statistics of the training set.

5.2.2 Result and Discussion

Figure 2 shows the weighted average of precision, recall and F-measure achieved by the four models. The proposed model shows significant performance improvement over the other compared models. The difference between the uni-bigram and proposed models were in feature selection and tweaking. Hence, their performance difference (F-measure: 0.83 vs. 0.52) speaks volume in favor of our feature selection and tweaking algorithms. However, as only 0.28% tweets were tweaked, the performance improvement can be attributed solely to the feature selection method. Though the unigram model and the uni-bigram model share the same unigram features, unigram model performs much better than the uni-bigram model (F-measure: 0.67 vs. 0.52). It proves that adding additional structural features can actually decrease the performance if not selected efficiently. **Figure 3** shows the per–class F-measures for the four models.

The LLDA model's performances for the classes 'Self' and 'News' are almost identical to that of unigram model (Self: 0.71 vs. 0.73; News: 0.79 vs. 0.81). However, for class 'False' unigram model outperforms the LLDA model (0.43 vs. 0.52). Overall performance of unigram model is better than that of LLDA model in terms of all three performance matrices as shown in Fig. 2. This substantiates that for a small training set, Naïve Bayes can perform better than many sophisticated machine learning algorithms as claimed by earlier researchers [24], [25]. Ramage et al. [17] reported promising results in categorizing tweets using LLDA. However, there are two major differences between their experimental environment and ours. First, our training set is smaller than their training set by many folds and second, they used many labels per tweet as we described in the related work section. However, we used only the class names as labels for the tweets while using LLDA.

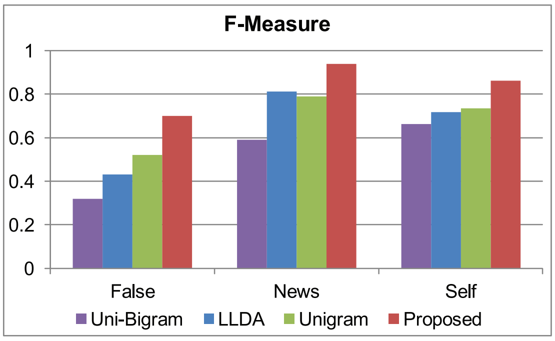


Fig. 3 Per class comparison of F-measure among the four models.

Table 3 Selected hashtags and the topics they represent.

Hashtags	Topic	Class
#job, #jobs	Job related news and advertisement	Job
#knick, #knicks	Baseball team New York Knicks	Knick
#nowplaying	Currently popular music tracks	Nowplaying
#occupywallstreet, #occupywallst, #ows	Occupy wall street movement	Ows
#realestate	Recent activities in real estate sector	Realestate

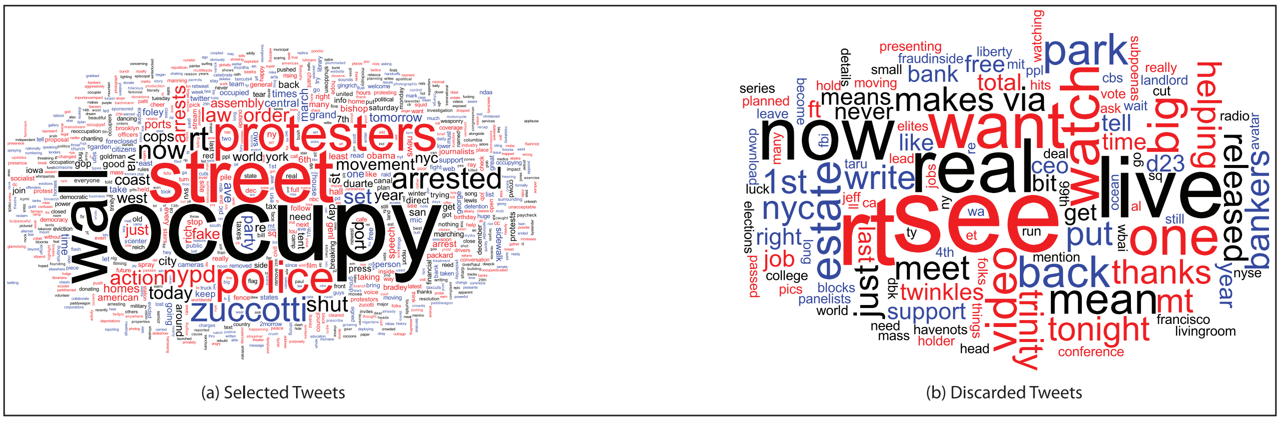
Table 4 Per class statistics of training data.

	Job	Knick	Nowplaying	Ows	Realestate
Initial training instances	1,772	2,857	1,582	5,833	1,272
Tweets with inadequate features	273	314	631	617	108
Total Tweaked	8	24	27	63	57
Unique Bigrams	8,061	21,730	12,948	54,521	17,309
Bigrams with Structural Importance	3,065	8,240	4,511	26,576	7,130

5.3 Experiment 2: Automatic Identification of Appropriate Hashtags

5.3.1 Experiment Data

For this experiment we have considered tweets from New York crawled in the period from December 6, 2011 to January 14, 2012. Among the 20 most frequent hashtags in these tweets, we selected nine representing five different topics. **Table 3** lists the selected hashtags and the topic they represent. A total of 13,316 tweets were considered in the training set. **Table 4** shows their distribution among different classes and some statistics on the selected bigram features. A total of 18,093 unique unigrams were



(a) Selected Tweets

(b) Discarded Tweets

Fig. 5 Word cloud for discarded training instances for hashtag #ows.

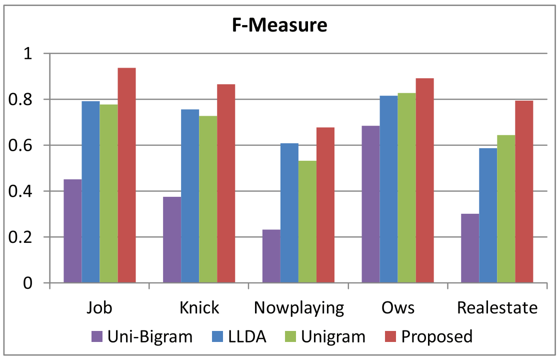


Fig. 4 Performance of the four models for an individual hashtag classes.

encountered in the training set. After pre–processing, removing stopwords and rare terms, a total of 4,603 unigrams constituted the final unigram feature set.

5.3.2 Results and Discussion

Figure 4 shows the performance comparison among the four models using F-measure. Like experiment 1, here also the proposed model outperforms the other three models by a large margin. The LLDA model performs better than the unigram model for three out of the five classes. As we pointed out in the discussion section of the previous experiment, the LLDA model's performance might be influenced by the size of the training set. The training set of this experiment is substantially larger than that of the earlier experiment.

Figure 5 shows the frequency distribution of unigrams in the tweets selected for the training set (on the left) and those discarded from the training set (on the right) for the class 'Ows'. The size of each word in the figure corresponds to its frequency in corresponding text corpus. Colors and orientation have no significance. As the figure reveals, high frequency words in the selected tweets (e.g., occupy, protesters, police, zuccotti, wall, street etc.) are much more intuitively closer to the topic 'OWS' than those in the discarded tweets (e.g., see, live, want, now, real etc.).

We also compared the reduction in misclassification rate due to the tweaking process. Two separate experiments were carried out using only the proposed model with the only difference between the experiments being tweaking and not tweaking of confounding outliers. **Table 5** and **Table 6** show the results. For both tables, the first column is the actual class of the tweet and the first row (due to lack of space, class 'Nowplaying' has been abbreviated to 'Now' and class 'Realestate' to 'Real') is the predicted class. By tweaking 1.34% tweets misclassification rate could be brought down by 0.42%. This finding agrees with the claim of Edgar et al. [20] that removing outliers from a training set can reduce the misclassification rate.

Table 5 Distribution of misclassified training instances before tweaking.

	Job	**Knick**	**Now**	**Ows**	**Real**	**Total**
Job	0	21	7	106	15	149
Knick	0	0	28	316	20	364
Nowplaying	1	135	0	212	11	359
Ows	10	97	137	0	194	438
Realestate	21	65	15	145	0	246
						1,556

Table 6 Distribution of misclassified training instances after tweaking.

	Job	**Knick**	**Now**	**Ows**	**Real**	**Total**
Job	0	18	9	102	13	142
Knick	0	0	24	304	20	348
Nowplaying	1	139	0	200	13	353
Ows	10	94	142	0	172	418
Realestate	23	68	11	132	0	234
						1,495

6. Conclusion

In this paper we put forward an argument that the length restriction imposed on individual posts makes tweets different from most other short texts like user reviews or web snippets. Hence, special measures need to be taken when training a BOW classifier to classify tweets. We propose two such measures here; one, better feature selection and two, identification and removal of confounding outliers. The decline of classifier performance from uni–bigram model to unigram model substantiates that just including bigram features might actually decrease the performance of the classifier. However, the improvement in performance from unigram model to the proposed model suggests that careful selection of structurally important bigrams can help the classifier discern the inter–class margin better. We have also proposed a stop–word selection method which prevents content–bearing terms from being included in the stoplist. Evaluation shows that performance can be improved even more by removal of confounding outliers from the training set. The sparsity of textual features in tweets and loose coupling of hashtags are the main sources of these outliers. Hence, if a semi–supervised approach is adopted for creating large training sets by using hashtags as class labels instead of using a manual labeling approach, the proposed tweaking method can be helpful in automatic identification of confounding outliers and eventual reduction of misclassification rates.

We are interested in extending the current work by including higher order n-gram features. We would also like to develop an automatic hashtag recommendation system based on the proposed model. However, our model parameters are still chosen based on heuristics. We would like to use optimization techniques for tuning those parameters in our future works.

References

[1] Naaman, M., Boase, J. and Lai, C.: Is It Really About Me?: Message Content in Social Awareness Streams, *CSCW '10 Proc. 2010 ACM Conf. Computer Supported Cooperative Work*, pp.189–192 (2010).

[2] Becker, H., Naaman, M. and Gravano, L.: Beyond Trending Topics: Real-world Event identification on twitter, *ICWSM '11 Proc. 5th Int. AAAI Conf. Weblogs and Social Media*, pp.438–441 (2011).

[3] Ilina, E., Hauff, C., Celik, I., Abel, F. and Houben, G.: Social Event Detection on Twitter, *ICWE '12 Proc. 12th Intl. Conf. Web Engineering*, pp.169–176 (2012).

[4] Benson, E., Haghighi, A. and Barzilay, R.: Event Discovery in Social Media Feeds, *ACL-HLT '11 Proc. 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Vol.1, pp.389–398 (2011).

[5] Chakrabarti, D. and Punera, K.: Event Summarization Using Tweets, *ICWSM '11 Proc. 5th Int. AAAI Conf. Weblogs and Social Media*, pp.66–73 (2011).

[6] Bollen, J., Mao, H. and Zeng, X.J.: Twitter Mood Predicts the Stock Market, *J. Comput. Sci.*, Vol.2, No.1, pp.1–8 (2011).

[7] Asur, S. and Hubrman A.H.: Predicting the Future with Social Media, *WI-IAT '10 Proc. IEEE/WIC/ACM Intl. Conf. Web Intelligence and Intelligent Agent Technology*, Vol.1, pp.492–499 (2010).

[8] Sakaki, T., Okazaki, M. and Matsuo, Y.: Earthquake Shakes Twitter Users: Real–Time Event Detection by Social Sensors, *WWW '10 Proc. 19th Intl. Conf. World Wide Web*, pp.851–860 (2010).

[9] Chunra, R., Andrews, J. and Brownstein, J.: Social and News Media Enable Estimation of Epidemiological Patterns Early in the 2010 Haitian Cholera Outbreak, *American Journal of Tropical Medicine and Hygiene*, Vol.86, No.1, pp.39–45 (2012).

[10] Aramaki, E., Maskawa, S. and Morita, M.: Twitter Catches the Flu: Detecting Influenza Epidemics Using Twitter, *EMNLP '11 Proc. Conf. Empirical Methods in Natural Language Processing*, pp.1568–1576 (2011).

[11] Lampos, V., Tijl, D.B. and Nello C.: Flu Detector – Tracking Epidemics on Twitter, *ECML PKDD '10 Proc. European Conference on Machine Learning and Knowledge Discovery in Databases*, Part III, pp.599–602 (2010).

[12] Abel, F., Celik, I., Houben, G. and Siehndel, P.: Leveraging the Semantics of Tweets for Adaptive Faceted Search on Twitter, *ISWC '11 Proc. 10th Intl. Conf. The Semantic Web*, Vol.Part I, pp.1–17 (2011).

[13] Sankaranarayanan, J., Samet, H., Teitler, B., Lieberman, M. and Sperling, J.: Twitterstand: News in Tweets, *SIGSPATIAL '09 Proc. 17th Intl. Conf. Advances in Geographic Information System*, pp.42–51 (2009).

[14] Culotta, A.: Towards Detecting Influenza Epidemics by Analyzing Twitter Messages, *SOMA '10 Proc. 1st Workshop on Social Media Analytics*, pp.115–122 (2010).

[15] Pang, B., Lee, L. and Vaithyanathan, S.: Thumb up? Sentiment Classification Using Machine Learning Techniques, *EMNLP '02 Proc. Conf. Empirical Methods in Natural Language Processing*, pp.79–86 (2002).

[16] Phan, X.H., Nguyen, L. and Horiguchi, S.: Learning to classify short and sparse text and web with hidden topics from large-scale data collections, *WWW '08 Proc. 17th Intl. Conf. World Wide Web*, pp.91–100 (2008).

[17] Ramage, D., Dumais, S. and Liebling, D.: Characterizing microblogs with topic models, *Intl. AAAI Conf. Weblogs and Social Media*, Vol.5, No.4, pp.130–137 (2010).

[18] Blei, D.M., Ng, A.Y. and Jordan, M.I.: Latent dirichlet allocation, *J. Mach. Learn. Res.*, Vol.3, pp.993–1022 (2003).

[19] Ramage, D., Hall, D., Nallapati, R. and Manning, C.D.: Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora, *In Proc. 2009 Conference EMNLP*, Vol.1, pp.248–256 (2009).

[20] Acuña, E. and Rodríguez, C.: An empirical study of the effect of outliers on the misclassification error rate, *Submitted to Transactions on Knowledge and Data Engineering* (2005).

[21] Rose, S., Engel, D., Cramer, N. and Cowley, W.: Automatic keyword extraction from individual documents, *Text Mining*, pp.1–20 (2010).

[22] Smadja, F.: Retrieving Collocations from Text, *J. Computational Linguistics*, Vol.19, pp.143–177 (1993).

[23] Dunning, T.: Accurate Methods for the Statistics of Surprise and Coincidence, *J. Computational Linguistics*, Vol.19, pp.61–74 (1993).

[24] Manning, C.D. and Schutze, H.: Foundations of Statistical Natural Language Processing, *The MIT Press* (1999).

[25] Witten, I.H., Frank, E. and Hall, M.A.: Data Mining: Practical Machine Learning Tools and Techniques, Third Edition, *Morgan Kaufmann Publishers* (2011).

[26] Damerau, F.J.: Generating and Evaluating Domain–oriented Multi–word Terms from Texts, *J. Inf. Process. Manage.*, Vol.29, pp.272–278 (1993).

[27] Toutanova, K., Klein, D., Manning, C.D. and Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network, *Proc. NAACL-HLT '03*, Vol.1, pp.173–180 (2003).

[28] Weng, J., Lim, E.P., Jiang, J. and He, Q.: Twitterrank: Finding topic-sensitive influential twitterers, *Proc. 3rd ACM Intl. Conf. Web Search and Data Mining*, pp.261–270 (2010).

[29] Mihalcea, R. and Tarau, P.: TextRank: Bringing order into texts, *Proc. EMNLP '04*, Vol.4, pp.404–411 (2004).

[30] Hulth, A.: Improved automatic keyword extraction given more linguistic knowledge, *Proc. EMNLP '03*, pp.216–223 (2003).

Muhammad Asif Hossain Khan received his B.Sc. and M.S. degree in Computer science from University of Dhaka, Bangladesh in 1999 and 2001 respectively. He joined the same university as a faculty member in 2002. Currently, he is pursuing Ph.D. in Information Science and Technology from the University of Tokyo, Japan. His main research interest is Natural Language Processing, Microblog Analysis and Information Retrieval. His current research involves analyzing Twitter stream for predicting changes in different social indicators. He is a student member of IEEE.

Masayuki Iwai received his Ph.D. (Media and Governance, Keio University) in 2004. He was the lecturer at Keio University, Graduate School of Media and Governance until 2008. From 2008, he joined JST/CREST project in Tokyo Denki University. From 2009–2012, he has been an Assistant Professor at the University of Tokyo, Department of Informatics and Electronics, Institute of Industrial Science. He works as an Associate Professor of Tokyo Denki University from 2013. His interesting research fields are SNS Context Analyzing, Sensor Data Mining, Wireless Sensor Networks, Distributed/Mobile Computing, Ubiquitous Application, Human Probe System, and MediArt System. He is a member of IPSJ, IEICE and IEEE.

Kaoru Sezaki received his B.E., M.E. and Dr.E. degrees in Electrical Engineering from the University of Tokyo in 1984, 1986, and 1989, respectively. He joined the Institute of Industrial Science at the University of Tokyo in 1989. He is now a Professor at the Center for Spatial Information Science at the University of Tokyo. From 1994 to 2000, he was a guest Associate Professor at the National Center for Science Information Systems. He has been a special member of Telecommunications Business Dispute Settlement Commission from 2001 to 2008. From 1996 to 1997, he was a visiting scholar at the University of California, San Diego. His research interests include Communication Networks, Location- and Context-aware Network Services, Collaboration Systems with Haptics, GIS and Image Processing. He is a member of IPSJ and IEEE.