International Journal of Service and Knowledge Management
International Institute of Applied Informatics
2019, Vol. 3, No. 1, 51 – 65

A Sentiment Polarity Classifier for Regional Event Reputation Analysis

Tatsuya Ohbe [*], Tadachika Ozono [*], Toramatsu Shintani [*]

Abstract

It is important to analyze reputation or demands for regional events, such as school festivals. In our previous works, we proposed sentiment polarity classification based on bag-of-words models and found that the traditional models were poor at classifying negative tweets. To improve the performance, we employed several classifier models based on deep learning models. In this paper, we described how to improve the performance of the sentiment polarity classification using deep learning models. We compared the performance of four models in terms of the classification accuracy and the training speed. As a result, we found that the CNN-based model, three words convolutions, was best among the four models. As the application, we also described the overview of a system based on the classifiers, which supports to analyze regional event reputation. We showed a case of a regional event analysis of a school festival by using our system.

Keywords: convolutional neural networks, recurrent neural networks, sentiment polarity classification, sentiment visualization

1 Introduction

There is great demand for automated tools that help the collection and coordination of regional event reputation. Examples of regional events are school festivals, fireworks displays, Comiket and so on. The participants and operators of these regional events often want to know their reputation from others as this can help them prepare for the next sessions. Recently, microblogs such as Twitter are widely used as they enable people to post and read short messages from anywhere. Information on regional events, including their reputation, can be found via social media. In this paper, the data of "regional event reputation" consists of (1) sentences of the reputation on a regional event in tweets and (2) the time series and location distribution of the tweets.

The implementation of an automated system requires expertise in natural language pro-cessing (NLP), machine learning (ML) and information retrieval (IR). Therefore, in this study, we developed a system that supports the collection and coordination of tweets about regional events and its analysis. Along with other social networking services (SNSs) such as Facebook, the content on Twitter is in real-time. Therefore, we

* Nagoya Institute of Technology, Aichi, Japan

have used Twitter as the data resource in the proposed system. In this study, we used sentiment polarity classification in our system to support and coordinate the reputation. This classification of positive and negative sentences helps to coordinate reputation. Our system classified tweets by their sentiment polarity for the purpose of coordinating the reputation. Our system visualized the result of the sentiment polarity classification for a macroscopic analysis, which could not be found from each tweet. Our system enables users to perform an exploratory data analysis by users, with a flexible comparable interface. The performance of the sentiment polarity classification is important to coordinate tweets and analyze regional event reputation accurately. Therefore, we evaluated its performance by comparing the classification by the system to that by people. We also discussed an example of a regional event analysis using our system and showed the efficiency of our system. In the previous works, we used bag-of-words (BoW) based models for the classification[1]. This paper describes how to improve the classifier using deep learning models.

In addition to this introduction, the paper is organized as follows. In Section 2, we describe the related works. We describe our sentiment polarity classification in 3. In Section 4, we describe the overview of how our system analyzes regional events and show an example of the analysis of a regional event using our system. In Section 5, we describe experiments for evaluating the performance of our sentiment polarity classification and discuss the result of the experiments in Section 6. Finally, we conclude this paper in Section 7.

2 Rerated Works

It is often difficult to evaluate the reputation of regional events due to the lack of sufficient data of reputation on such events. Someone usually organizes the reputation in the case of massive country-wide events such as New Year's visit to a shrine. However, the reputation in the case of regional events is often left undetermined. Those who want to know the reputation of a regional event must collect them manually. Forley et al.[2] proposed a method for extracting and retrieving regional events from the Web. They extracted regional events without any additional human supervision by using texts, meta data, and structural features.

Deep learning models have achieved remarkable results in many fields. In the field of NLP, deep learning models are frequently used to make word vectors. The deep learning methods to make word vectors are called neural word embedding. Simple BoW word vectors have vocabulary size dimension. Neural word embedding reduces dimensions of word vectors and makes distributed representation. Mikolov et al.[3] proposed *Word2vec*, which is one of the neural word embedding. Joulin et al.[4][5] proposed another method for make word vectors called *fastText* . Le et al.[6] proposed *Paragraph Vector*, which is one of the methods to make a vector from a sentence. Deep Learning models are used not only for word representations but also for other NLP tasks. Convolutional Neural Networks (CNN) is used for computer graphics such as image generation or image processing. There are several approaches for text classification with CNN. Kim[7] proposed CNN for sentence classification. Severyn et al.[8] proposed sentiment polarity classification for Twitter using deep CNN . Recurrent Neural Networks (RNN) is used to make sentence representation from word representation. RNN is often used for a sequence to sequence (Seq2seq) model. In Seq2Seq, sequential vectors are compressed into one context vector. In NLP field, an Encoder-Decoder language model is often used for machine translation. Classical RNN, which simply use the hidden state of the last unit, cannot treat long sequences. There are several methods to adapt to long sequences such as Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU)[9].

Sentiment polarity classification is one of the tasks in NLP that aims to understand human sentiment. Sentiment polarity classification is often used to extract the reputation. In this classification, data are typically classified into three categories: positive, negative, or others. Wang et al.[10] focused on hashtags in Twitter and applied sentiment polarity classification to them . They made a graph of hashtags and showed sentiment polarity as a direction in the graph. Paula et al.[11] proposed a method to classify blog posts as objective, positive or negative and spawned feature vectors of posts focusing on part-of-speech (POS) . They classified feature vectors of posts with Support Vector Machine (SVM). Wei et al.[12] proposed a method learning sentiment on product reviews using Hierarchical Learning and Sentiment Ontology Tree. Terazawa et al.[13] extracted positive and negative sentences in web sites and extracted sentences by sentiment polarity and set snippet of searching result . They used Paragraph Vector as the feature vectors of sentences and multinomial logistic regression analysis (MLRA) as the classifier.

Twitter, one of the most popular SNSs, is often used as a kind of social sensors. A social sensor is an idea using SNSs as a kind of sensors to detect events. For example, Twitter is used as a social sensor for cherry blossom forecast or for detecting train accidents. Rui et al.[14] implemented a system for detecting and analyzing events using Twitter. Focusing on Crime and Disaster related Events (CDE), they identified the importance of event and visualized tweets with maps and line charts. Hirata et al.[15] implemented a system that detects events using Twitter and collects related posts from news sites. They also collected relevant tweets using news posts. In these two works, they extracted an event from tweets. In the present work, we detected regional event reputation in a regional event.

Many studies have reported the use of automated tools for data mining. Sunayama et al.[16] developed Total Environment for Text Data Mining (TETDM), which is an automated text mining tool that targets a wide range of people from average web users to researchers. TETDM enables users to combine some text mining tools and links each tool. In addition, it also enables users to develop a text mining tool. In this work, we use sentiment polarity classification to extract the reputation or accident such as CDE. Our system visualizes tweets with maps and graphs. Our system links tweets with their time series and location data. Using sentiment polarity classification and visualization of tweets, our system supports the detection of reputation as social sensors.

3 Sentiment Polarity Classification

We propose several classifiers based on BoW models and deep learning models. First, each tweet is separated into words by morphological analysis. To classify tweets, the system spawns tweet vectors, which is feature vectors of tweets for the classification.

3.1 Classification Using BoW model

To spawn a BoW-based tweet vector, we directly use words in a tweet. From all part-of-speech (POS), we use nouns, verbs, and adjectives. We weight BoW vectors with tf-idf that makes the value of common words low. We also use Semantic Orientations of Words [17], which contains 55,125 annotated words. In Semantic Orientations of Words, each word is assigned real value in the range of -1 to +1. The positive words are assigned with a value

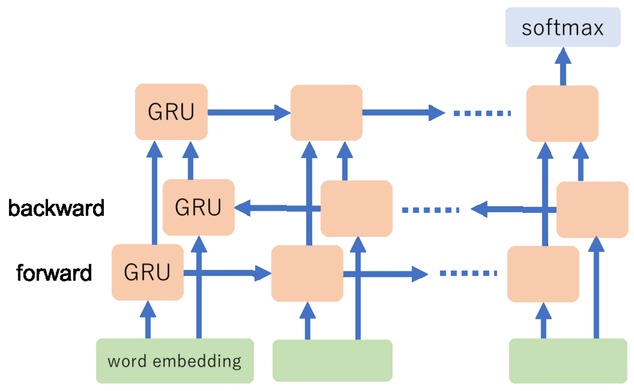


Figure 1: RNN Model Architecture.

close to +1 and the negative words are assigned with a value close to -1. We use sum total polarity values of positive words or negative words in a tweet.

We define a tweet vector $X(s)$ of tweet s as follows:

$$X(s) = [\omega_1, \omega_2, ..., \omega_i, ..., \omega_{n-1}, \omega_n, P, N]$$
$$P = \sum_{i=1}^{l} p_i \quad N = \sum_{j=1}^{m} n_j$$

$X(s)$ contains a BoW vector $[\omega_1, \omega_2, ..., \omega_i, ..., \omega_{n-1}, \omega_n]$ and the sum total polarity values $[P, N]$. Note that ω_i is a BoW value weighted by tf-idf of the i-th word of s. P and N are sum total polarity value of (positive/negative) words in the tweet, respectively. l and m are the number of (positive/negative) words in the tweet, respectively. p_i (or n_j) is the polarity value of a positive word i (or a negative word j). We use logistic regression for classification of tweet vectors. In sentiment polarity classification, tweets are classified into three categories; therefore, we use multinomial logistic regression analysis for classifier. For training the classifier, we use sentences annotated with labels: 1 (positive), -1 (negative), or 0 (others). From these annotated sentences, we get BoW vectors the same as tweet vectors.

3.2 Classification Using Recurrent Neural Networks

We use deep RNN to spawn a tweet vector from word vectors of each word in the tweet. As the unit of RNN, we use GRU. Figure 1 shows our RNN architecture. We use deep GRU-RNN with two layers. The first layer is a bi-directional layer, which consists of a forward RNN and a backward RNN. The input of the forward RNN is each word in a tweet. The input of the backward RNN is reverse ordered words of the tweet. Given a word at the time t, the hidden state of the forward RNN and the backward RNN are denoted as follows:

$$\overrightarrow{h}_t = G(\overrightarrow{W_x} x_t + \overrightarrow{W_h}\,\overrightarrow{h}_{t-1} + \overrightarrow{b_h})$$
$$\overleftarrow{h}_t = G(\overleftarrow{W_x} x_t + \overleftarrow{W_h}\,\overleftarrow{h}_{t+1} + \overleftarrow{b_h})$$

where the $\overrightarrow{W_x}$ and $\overleftarrow{W_x}$ denote the weight matrixes of the input, $\overrightarrow{W_h}$ and $\overleftarrow{W_h}$ denote the weight matrixes of the hidden layer, $\overrightarrow{b}$ *and* $\overleftarrow{b}$ denote the bias vectors, and a function G denotes a GRU function. The second layer is a uni-directional layer. The output of both the forward

.

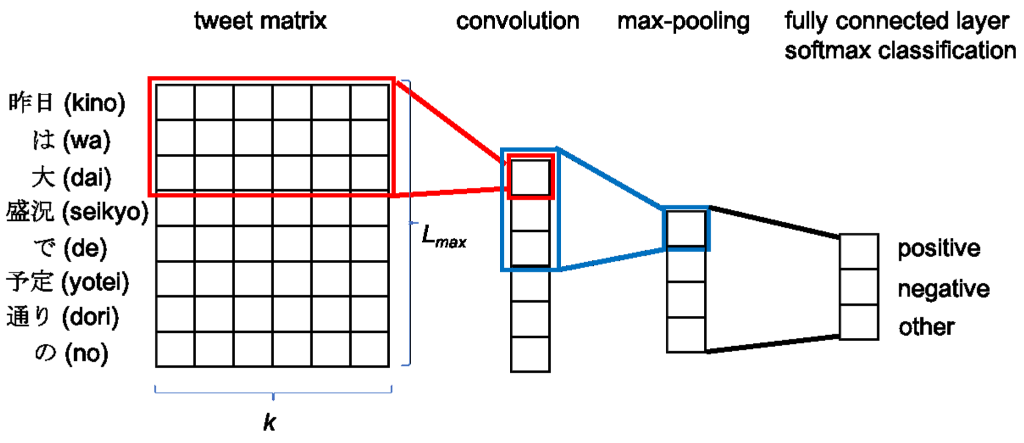


Figure 2: CNN Model Architecture.

and backward RNN are concatenated and inputted into the second layer. Therefore, the hidden state of the second layer at the time t is denoted as follows:

$$y_t = G(W_x(\overrightarrow{h}_t; \overleftarrow{h}_t) + W_h h_{t-1} + b_y)$$

where the W_x denotes the weight matrix of the input, W_h denotes the weight matrix of the hidden layer, b_y denotes the bias vector, and a function G denotes a GRU function. Note that operator ";" means a concatenate function. Word vectors are fine tuned during the training. Finally, the hidden layer of the last unit outputs a sentence vector. The output layer is a soft-max classification layer. The output layer classifies the sentence vector into the three categories. As the loss function, we use cross-entropy.

3.3 Classification Using Convolutional Neural Networks

We use word-level CNN to classify tweets. Figure 2 shows our CNN architecture. The input of our model is a tweet matrix. The tweet matrix is a matrix of word vectors of each word in the tweet. Given a tweet s with N words $\{\omega_1,\omega_2,...,\omega_N\}$, to justify the matrix size of each tweet, set dummy tokens for word ω_i for $L_N < i \leq L_{max}$. Note that L_N shows the length of a tweet s and L_{max} shows max sentence length. Next, we get a k-dimensional word vector for each word in a tweet. We use *fastText* as the pre-trained word vectors. The word vectors are fixed during the training. Given a word ω_i, we look up the word vector w_i. Finally, we get a tweet matrix $S \in \mathrm{R}^{L_{max} \times k}$ for each tweet. The i-th row of S represents a word vector w_i.

Our model consists of several convolution layers and pooling layers. The convolution layers are convolution sentence matrixes. The convolution filter size is $n \times k$. We stride the filter by one word. We apply an activation function for each convolution layer. The activation function is Rectified Linear Unit (ReLU). Then, we apply max-pooling. Finally, we use multi-layer perceptron (MLP) as fully connected layers. The output layer is a soft-max classification layer. As the loss function, we use cross-entropy.

4 Regional Event Information Analysis Support System

4.1 Regional Event Information Analysis

Using the proposed classification, we implemented a system visualize tweets sentiments. Our system supports the analysis of regional event reputation in three steps, (A) collection step, (B) classification step, and (C) visualization step. In the collection step, the system collects tweets by keywords given by the user. In the classification step, the system extracts positive tweets and negative tweets by sentiment polarity classification. In the visualization step, the system visualizes the tweets using the results of sentiment polarity classification.

In the collecting step, the system collects data sets to analyze. In this work, two types of data sets are collected. One is a static data set and the other is a dynamic data set. The static data set is the snapshot of dynamic data set. The dynamic data set required in the search qualification, in which the contents changes continuously. Dynamic data sets are automatically collected by the system. The system collects tweets by keywords inputted by the user. The static data set refers to the data set that was collected in the past. The user can bring out and save collected tweets as a static data set.

In the classification step, the system extracts positive tweets and negative tweets by sentiment polarity classification. In sentiment polarity classification, tweets are classified into three categories: positive, negative, and other. By extracting positive tweets and negative tweets, we support the user coordinates for the reputation of a regional event. Since Twitter is simple and convenient, there are a lot of noisy tweets in collected tweets. Thus, the system excluded noisy tweets by extracting other tweets. The system classifies feature vectors of tweets with a supervised classifier.

In the visualization step, the system visualizes tweets using the results of the sentiment polarity classification. Our system includes several ways of visualization, mainly visualization of tweets time series and location distribution. The system visualizes time series with line charts that showed a horizontal axis time and a vertical axis in words of the count of tweets. With visualization of time series of tweets, the user can figure out some changes of counts of tweet. Therefore, the user can detect specific change of the reputation. Using the result of sentiment polarity classification, the graphs show each counts of tweets: all tweets in black color, positive tweets in red, negative tweets in blue, and other tweets in white. The system visualizes location distribution setting makers on a map by using Google Maps API to draw the map and markers. By setting markers on a map, the user can figure out what do the tweets describe. There are two types of location data we use, one is included in the original tweets and the other is gotten from geographical names in the tweets. Twitter officially supports the location data; however, there are only a few people who use the function. Therefore, we also use location data obtained from geographical names in the tweets. In morphological analysis, we first obtained geographical names, then determined the location data using Google Maps Geocoding API. For example, from a geographical name "Nagoya Institute of Technology", we can get a location data "latitude: 35.156512, longitude: 136.924872".

4.2 Implementation

We implemented the system as a web application based on Node.js, a server-side JavaScript environment, and Python, with the support of MongoDB, Twitter API, and Google Maps API. For morphological analysis, we used MeCab, a Japanese morph analyzer, and its Python bindings. To spawn the dictionary and tweet vectors, we used Gensim, an NLP library of Python. To learn and classify with LR, we used Scikit-learn, a ML library of Python. In addition, we used MeCab-IPADIC-Nelogd [1] as the dictionary of MeCab. Nel-ogd is a dictionary that contains a lot of new words and coined words. Nelogd is updated several times a month and contains over 2 million hot words. Since Twitter users often use new words and coined words, Nelogd is necessary to morph analyze tweets more accurately.

Figure 3 shows the architecture of our system. Our system contains two major parts: the client side and the server side. In the client side, the user gives input data through forms in the input page to collect tweets. The input data include keywords and start and end dates. The user can also input additional dictionaries and location data. These input data are sent to the server. The server consists of a collecting module and a classification module. In the collecting module, the tweets include the keywords are collected through Twitter API and stored into the database. The classification module takes the text of a tweet from the database and morph analyze them. Then the module spawns the feature vector of the tweet. At the same time, using extracted words as geographical names, the module gets location data through Google Maps Geocoding API. Then, the module stores location data into the database. The classification module classifies the feature vectors by their sentiment polarity. The classified labels are stored into the database. In the client side, the visualization module takes tweets and other data from the database and draws various graphs and the map.

We used the TSUKUBA Corpus as the training resources. The TSUKUBA Corpus is an annotated data set, included in Rakuten data sets [1] provided by Rakuten, Inc. It contains 4,309 sentences from review in Rakuten Travel [2]. Each sentence is assigned one or more labels, which include “p”, “k”, “y”, “e”, “Z”, and “o”. A sentence with a “p” label means positive sentence. A sentence with a ”k” label means a complaint sentence. A sentence with a “y” label means a claim sentence. A sentence with a “e” label is a neutral sentence that includes both positive and negative descriptions. A sentence with a “Z” label is a sentence without any reputation. A sentence with an “o” label is ‘others’. From the TSUKUBA Corpus, we used sentences with “p”, “k”, “y”, “Z”, and “o” labels, so the sentences with “e” labels were ignored. We treated the sentences with “p” labels as positive ones, those with “k” or “y” labels as negative ones and those with “o” or “Z” labels as other ones.

4.3 Exploratory Data Analysis

Our system enables users to perform an exploratory data analysis. First, the system collects tweets with the keyword inputted by the user. Second, the system classifies the collected tweets and visualizes the tweets using the results of classification. (1) The user observes the visualized tweets and inputs additional data as appropriate. (2) Then the system collects tweets with the additional data and visualize tweets again. The user analyzes regional event reputation exploratory with the cycle of (1) and (2). We also enable users to analyze regional event reputation exploratory with a flexible comparable interface and tweet extraction by keywords and NG words. We use transparent layers for comparing graphs. We draw each graph on a transparent window. The windows are draggable and resizable. The user moves the windows and compares an arbitrary set of

[1] https://github.com/neologd/mecab-ipadic-neologd/

[1] http://rit.rakuten.co.jp/opendataj.html

[2] http://travel.rakuten.co.jp/

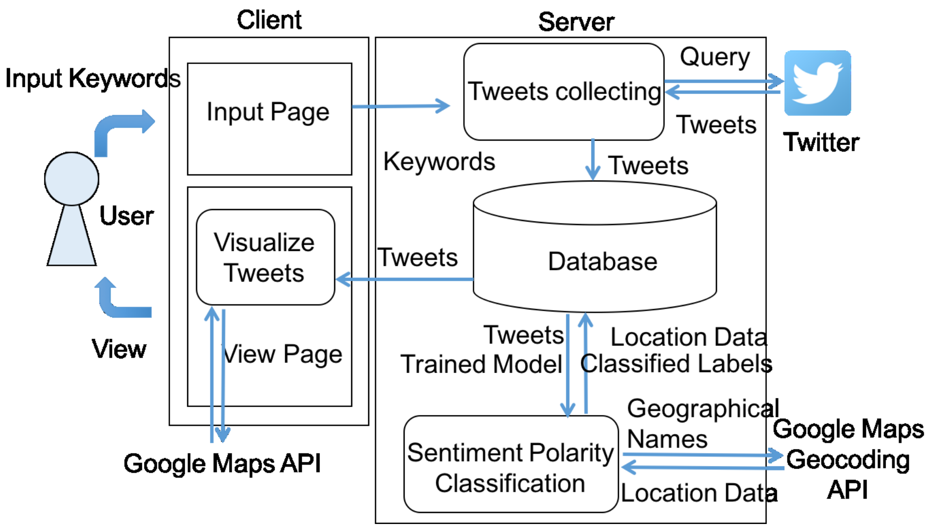
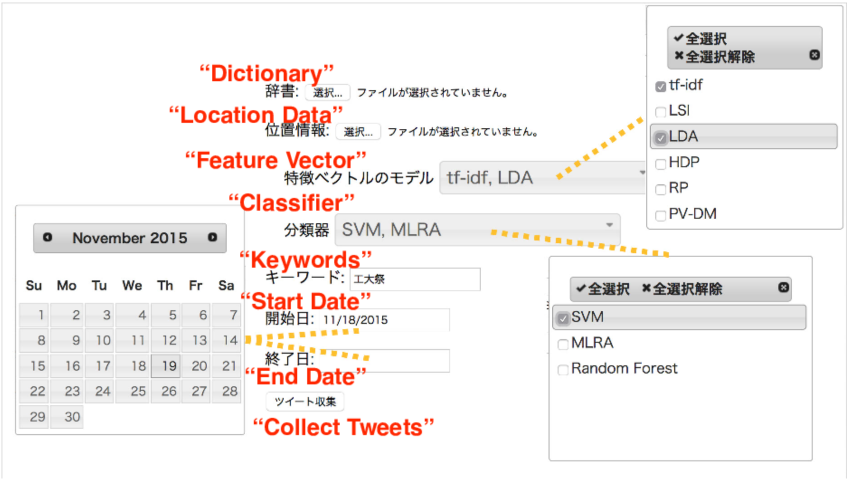


Figure 3: System Architecture.



辞書: 選択... ファイルが選択されていません。
位置情報: 選択... ファイルが選択されていません。
特徴ベクトルのモデル tf-idf, LDA
分類器 SVM, MLRA
キーワード: 工大祭
開始日: 11/18/2015
終了日:
ツイート収集

✔全選択 ✖全選択解除
[✓ tf-idf]
LSI
[✓ LDA]
HDP
RP
PV-DM

✔全選択 ✖全選択解除
[✓ SVM]
MLRA
Random Forest

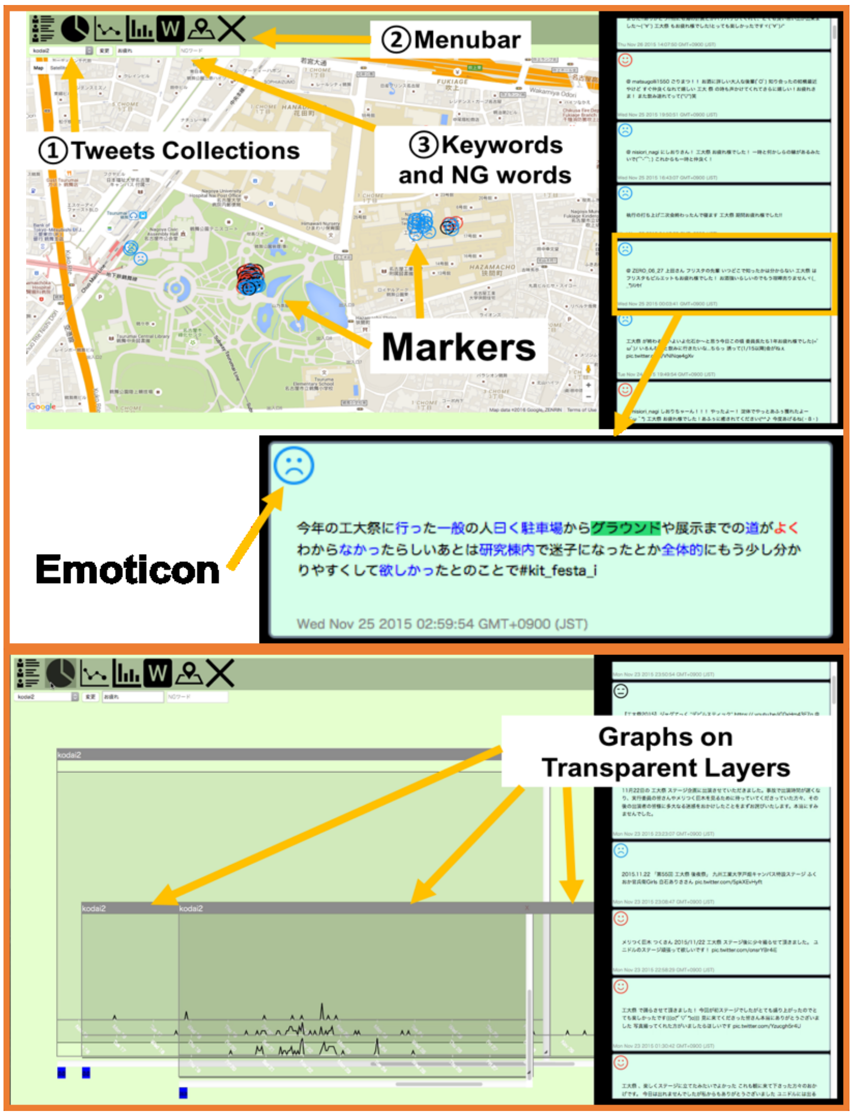
November 2015

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Figure 4: Input Page for Exploratory Data Analysis.

graphs. Therefore, the user can move and overlay graphs to suit a gap between graphs and analyze correlation between data. The user can extract tweets by keywords or NG words. There are unintended tweets in collected tweets. For example, if there are some regional events that have the same name, it refers to tweets about both regional events collected at the same time. The user can eliminate unintended tweets using AND searching when the system collects tweets. However, it is difficult to predict the word that is efficient for eliminating unintended tweets or extracting only the required tweets. Therefore, we collect unintended tweets once, and then the user extracts the tweets. Figure 4 shows the input page. The user inputs a keyword and the start and end dates from this page. The user selects both dates from a calendar form. The user can upload additional dictionaries or location data, optionally. As an additional dictionary, a MeCab compatible file is available. For location data, a CSV file in the format of “location name, latitude, longitude” is also available.

Figure 5 shows an example of the system. Selecting a tweets collection from the drop-down list on the top ①, the system shows collected tweets on the right side. We show sentiment polarity of each tweet by setting emoticons: positive tweets with smiley faces, negative tweets with sad faces, and other tweets with neutral faces. We show the map on the left side and set icons same to tweets list, as marker. The user can show graphs by clicking the icons in the menu bar on the top②, like Figure 5. On selecting a range of tweet time series, the corresponding tweets in the list turns into orange. The user can show or hide each classified tweets by checking the checkboxes. In addition, the user can extract tweets by keywords or NG words inputting from the form on the top③.



今年の工大祭に行った一般の人曰く駐車場からグラウンドや展示までの道がよくわからなかったらしいあとは研究棟内で迷子になったとか全体的にもう少し分かりやすくして欲しかったとのことで#kit_festa_i

Wed Nov 25 2015 02:59:54 GMT+0900 (JST)

Figure 5: Visualization of Sentiment Polarity Classification.

This section explains a use case of the system. By using our system, we analyzed the reputation of “Kodaisai” (the school festival of Nagoya Institute of Technology). Figure 6 shows the line chart of the count of the tweets. Two lines in Figure 6 shows the count of all tweets and that of negative tweets. We detected a characteristic change about tweets time series. At the hours 1 in Figure 6, the count of negative tweets is soaring, while the count of all tweets is on the downside. This means the percentage of negative tweets to all tweets is increasing, indicating that something bad is about to occur.

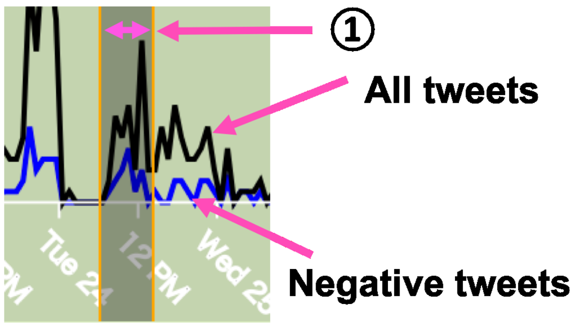


Figure 6: Negative tweets are soaring, while all tweets are on downside.

Then we extracted negative tweets at the hours. In the extracted tweets, there are criticisms against the regular examination or the deadline of the report right after the festival. Using this detection, the operators of the festival could change the date for the next year. From this result, we believe that our system contributes to the detection of some characteristic change as a social sensor.

5 Experiments

The performance of the sentiment polarity classification is important to coordinate tweets and analyze regional event reputation accurately. Therefore, we conducted evaluation experiments of the performance of sentiment polarity classification in our system. We conducted the experiments as follows.

A). SVM and MLRA
We changed the POS used, the use of Semantic Orientations of Words, classifier, and the model of feature vectors. We tested the classification accuracy with each combination.
We also experimented it by comparing the classification by people and by our system. We determined the accuracy of total categories and precision and recall of each category.

B). RNN and CNN
We experimented the classification using several deep learning models. We determined the accuracy of total categories and the precision and recall of each category.

5.1 SVM and MLRA

We determined the accuracy of the sentiment polarity classification to get the best combination of the factors of tweet vectors and the classifier. We used the following combinations: POS, use of Semantic Orientations of Words, the classifier and the model of feature vectors. For the classifiers, we tested SVM and MLRA. As the model of feature vectors, we tested tf-idf and LDA. We examined the classification of the sentences in the training data; Table 1 shows the result. As we see, we get the highest accuracy with the combination of surface, use, MLRA and tf-idf. Therefore, we use the combination as the default combination of the system.

Table 1: Classification Accuracy.

POS	SOW	SVM		MLRA	
		TF-IDF	LDA	TF-IDF	LDA
surface	non-use	63.9	60.4	63.8	59.2
surface	use	63.6	61.0	64.0	59.3
original form	non-use	65.0	62.3	65.5	60.5
original form	use	65.2	62.8	65.8	61.3

Table 2: Experimental Result of Tweet Classification.

		A	B	C	D	E	major	unanimous
accuracy		63.0	58.7	58.7	55.7	50.7	67.3	71.0
positive	precision	57.5	49.3	53.4	64.4	64.4	67.9	67.7
	recall	61.8	53.7	55.7	46.5	45.6	65.5	72.4
negative	precision	21.5	27.7	23.1	41.5	33.8	20.9	15.4
	recall	53.8	54.5	50.0	52.9	40.0	65.3	66.7
other	precision	87.6	84.0	83.3	71.6	63.6	89.0	93.8
	recall	66.1	63.6	63.4	67.8	63.6	69.8	71.7

In this case, tf-idf is better than LDA about the tweet vectors, and MLRA is better than SVM about the classifier.

We randomly extracted 300 tweets from the collected 2,769 tweets with keyword “Kodaisai”, and classified them manually. We found out the total accuracy, precision of each class, and recall of each class. We solicited five 20-years-old men as volunteers, and each subject classified tweets. Since sentiments of each person are different from each other, so we treated “major” tweets and “unanimous” tweets. We defined “major” tweets as tweets labeled the same label by three or more of the subjects. We defined “unanimous” tweets as tweets labeled the same label by all the subjects. There were 223 “major” tweets and 158 “unanimous” tweets in the extracted 300 tweets. Table 2 shows the results. The columns “A” to “E” represent the results of each subject. The columns “major” and “unanimous” show the results of the system for “major” tweets and “unanimous” tweets, respectively. We obtained about 70% accuracy with “major” and “unanimous” tweets and obtained over 90% precision with the other tweets in “unanimous” tweets. However, the precision on the negative tweets was very low.

5.2 RNN and CNN

We splited the TSUKUBA Corpus into train data and test data. We treated 390 sentences as the test data and the remains as the train data. Both of our RNN and CNN models cannot treat words those are not in the dictionary. Therefore, the unknown words in the sentence were replaced to the dummy token. We compared RNN and CNN models. We describe the RNN classifier. The size of both the forward and backward hidden layer was

Table 3: Experimental Result of Classification using Deep Learning Models.

		RNN	BRNN	CNN	CNN-multi
accuracy		63.6	70.0	79.0	76.4
positive	precision	65.1	75.1	87.2	86.4
	recall	86.2	81.0	84.1	81.5
negative	precision	60.1	64.4	70.3	66.4
	recall	65.1	71.7	73.6	72.6
other	precision	61.1	63.5	72.5	68.9
	recall	12.4	44.9	74.2	70.0

Table 4: Training time (sec) of RNN and CNN.

	RNN	BRNN	CNN	CNN-multi
train	515.50	807.38	31.95	162.34

100 dimensions. The size of the second hidden layer was 200 dimensions. We used GRU-RNN with a uni-directional forward layer as the baseline. The column "RNN" in Table 3 shows the result of classification using the baseline RNN. The baseline RNN classification got 63.6% accuracy. In the case of the baseline RNN, we obtained stable precision about every classes. On the other hand, the recalls of the other tweets were low. The column "BRNN" in Table 3 shows the result of classification using our bi-directional RNN. The bi-directional RNN got higher accuracy than the baseline RNN and the recall of the other tweets were also high. We describe the CNN classifier. As the pre-trained word vectors, we used 300 dimensions *fastText* vectors. We made the *fastText* vectors from Japanese Wikipedia. In the training, we applied dropout before the output layer. We trained each model for 100 epochs. To optimize updating weights, we used *Adam* [18] because of its stability. The column "CNN" in Table 3 shows the result of classification using CNN with three words convolution. As we see, we got 79.0% accuracy. This accuracy is higher than the BoW models. In the case of the BoW model, we got very low precision. However, in the case of the CNN model, we got 79.0% accuracy and the precision of negative sentences were still high. The column "CNN-multi" in Table 3 shows the result of classification using CNN with multi convolution filters. We used filters that convoluted each three, five and seven words. We predicted that the model with multi filters overcame the model with single three words filter. However, this model with multi filters obtained worse performance. Table 4 shows the training time of each model during one epoch. Both the CNN models learned faster than the RNN models. Therefore, the CNN models are better than the RNN models in terms of the classification accuracy and the training time.

6 Discussion

6.1 SVM and MLRA

The experiments showed that the use of the original form is better than using the surface about POS. Moreover, the use of Semantic Orientations of Words is better than its non-use. Almost all words in Semantic Orientations of Words are contained as the original form. Therefore, we considered to get better performance when using the original form than the surface. In the experiment, the performance improved a bit with using Semantic Orientations of Words. On the other hand, the improvement was smaller when the original form was used. Therefore, we should check up how to use polarity values in tweet feature vectors. In this work, we use SVM and MLRA, which contain fixed parameters. Therefore, we should change the parameters first and then test. For example, we should use grid search to get the best parameter. In the tweet classification experiment, we obtained over 90% precision about other tweets. Due to this extraction, the elimination of noisy tweets was expected. On the other hand, we got very low precision about negative tweets. We believe this is mainly because of less appearance of negative words in negative sentences. In the cases of positive sentences, positive words were simply predisposed to appear. On the other hand, in cases of negative sentences, negative words do not appear in gentle negative sentences. To take care of this problem, we should use another method other than BoW. For example, we should use correlation of words or context of sentences. In addition, there are less negative tweets in the test data, because the regional event we treat this time does not gain so much negative reputation. If we would treat regional events that face bad weather or some accidents, the result might be change.

In this work, we treated and annotated data set of not tweets but reviews about travel. We believe that travel reviews have relevance with regional event reputation. However, travel reviews are less noisy than tweets and include biased words. Therefore, we should use annotated tweets and experiment performance of our system. More number of sentences should be used to train classifier. From 4,309 sentences in the TSUKUBA Corpus, we ignored sentences with an “e” label and used only 4,003 sentences, which are too few to be used as training data. However, there are little Japanese annotated datasets about sentiment polarity. Therefore, we must find another way to obtain training data. For example, we should automatically collect sentences about sentiment polarity.

6.2 RNN and CNN

In the experiment, the baseline RNN model classified negative sentences more accurately than BoW models. The BoW model treats a sentence as a set of words and does not consider the word order. We believe that is because the hidden states of RNN model can treat sentences as sequential data. On the other hand, we get low recall about other tweets. Our bi-directional RNN model obtained higher accuracy and enabled to classify other tweets accurately. Hidden states from next words as the context vector might help accurate classification. In the experiment, our CNN model classifies negative sentences more accurately than BoW models. We believe convolution computation enables to treat inversion of the polarity. For instance, a negative word followed by a positive word is a negative representation. The BoW models cannot treat this inversion of the polarity. CNN models can treat these adjacent words. A convolution layer extracts local inversion of the polarity and a pooling layer extracts global inversion of the polarity.

In future work we might consider the combination of the models that will improve the accuracy of classification. For instance, first, the BoW-based model extracts other tweets, which can classify other tweets most accurately in our models. Next, the RNN or CNN-based model classifies positive tweets and negative tweets.

7 Conclusion

In this paper, we proposed several deep learning models to classify tweets for regional reputation analysis. We evaluated the performance of the sentiment polarity classification using deep learning models. Using deep learning models, we achieved higher classification accuracy than the previous works. We showed the CNN-based classifier with three words convolutions was better than other models. As the application, we also implemented a support system for analyzing regional event reputation with sentiment polarity classification and visualization of tweets. We enabled users to analyze regional event reputation exploratory with a flexibly comparable interface and tweet extraction by keywords and NG words. We discussed an example of regional event analysis using our system and showed the efficiency of our system in analyzing regional event reputation. This system may contribute to the elimination of noisy tweets and coordination of regional event reputation.

Acknowledgments

This work was supported in part by JSPS KAKENHI Grant Number 15K00422, 16K00420.

References

[1] T. Ohbe, T. Ozono, and T. Shintani, "Developing a sentiment polarity visualization system for local event information analysis," in *Advanced Applied Informatics (IIAI-AAI), 2016 5th IIAI International Congress on*. IEEE, 2016, pp. 19–24.

[2] J. Foley, M. Bendersky, and V. Josifovski, "Learning to extract local events from the web," in *Proceedings of the 38th Annual ACM SIGIR Conference*. ACM, 2015, pp. 423–432.

[3] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, 2013, pp. 3111–3119.

[4] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," *arXiv preprint arXiv:1607.01759*, 2016.

[5] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *arXiv preprint arXiv:1607.04606*, 2016.

[6] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents." in *ICML*, vol. 14, 2014, pp. 1188–1196.

[7] Y. Kim, “Convolutional neural networks for sentence classification,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).* Association for Computational Linguistics, October 2014, pp. 1746–1751.

[8] A. Severyn and A. Moschitti, “Twitter sentiment analysis with deep convolutional neural networks,” in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR ’15. ACM, 2015, pp. 959–962.

[9] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder–decoder for statistical machine translation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).* Association for Computational Linguistics, 2014, pp. 1724–1734.

[10] X. Wang, F. Wei, X. Liu, M. Zhou, and M. Zhang, “Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach,” in *Proceedings of CIKM ’11*, 2011, pp. 1031–1040.

[11] B. Vincent, L. Xu, P. Chesley, and R. Srhari, “Using verbs and adjectives to automatically classify blog sentiment,” in *Proceedings of AAAI-CAAW-06*, 2006, pp. 27–29.

[12] W. Wei and J. A. Gulla, “Sentiment learning on product reviews via sentiment ontology tree,” in *Proceedings of ACL ’10*, 2010, pp. 404–413.

[13] Y. Terazawa, S. Shiramatsu, T. Ozono, and T. Shintani, “Sentiment polarity analysis for generating search result snippets based on paragraph vector,” in *Advanced Applied Informatics (IIAI-AAI), 2015 IIAI 4th International Congress on.* IEEE, 2015, pp. 109–114.

[14] R. Li, K. H. Lei, R. Khadiwala, and K. C.-C. Chang, “Tedas: A twitter-based event detection and analysis system,” in *Proceedings of ICDE 2012*, 2012, pp. 1273–1276.

[15] N. Hirata, S. Shiramatsu, T. Ozono, and T. Shintani, “A system for collecting tweets using event-based structuring of web contents,” *International Journal of Computer Science and Artificial Intelligence*, vol. 3, no. 2, pp. 50–58, 2013.

[16] W. Sunayama, Y. Takama, Y. Nishihara, T. Kajinami, M. Kushima, and H. Tokunaga, “Practical application in development and use of mining tools with total environment for text data mining,” *Transactions of the Japanese Society for Artificial Intelligence*, vol. 29, pp. 100–112, 2014.

[17] H. Takamura, T. Inui, and M. Okumura, “Extracting semantic orientations of words using spin model,” in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, 2005, pp. 133–140.

[18] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2014.