Comparison of Machine Learning Techniques for Classification of Information Types on Twitter

Michal Ptaszynski †* Fumito Masui † Yoko Nakajima § Hiroshi Hayakawa ‡
Takehiko Saito ‡ Yasunori Miyamori‡

† Department of Computer Science, Kitami Institute of Technology
‡ Department of Civil and Environmental Engineering, Kitami Institute of Technology
§ Department of Information Engineering, Kushiro National College of Technology

* Corresponding author, Email: ptaszynski@cs.kitami-it.ac.jp

Abstract

In this paper, we compare various machine learning techniques to classify different types of information appearing on Twitter, with the aim of implementation in information triage during disasters. We assume that people seek different types of information right after a disaster event and sometime later. We base our approach on a preliminary study classifying the information on Twitter into three general types: Primary Information (first-hand reports), Secondary Information (second-hand reports, such as retweets), and Sesquiary Information (opinions, etc.). We compare a number of classifiers, including the proposed one based on Deep Convolutional Neural Networks.

1 Introduction

Twitter[1] is one of the most popular Social Networking Services (SNS). It specializes in information dissemination in the form of short messages. Additionally, by utilizing unique features such as "retweets," or "hashtags," Twitter allows easy transmission of information to an unspecified number of users. Usefulness of such functions made Twitter an important source of information in daily life, influencing the decision making process of many people.

The popularity of Twitter has also made it an effective instrument for tracking world-wide tendencies. Therefore much of research has currently been actively conducted using data obtained from Twitter. For example, Kuwano et al. [1] have extracted tourist information from Twitter, or Umejima et al. [2] has made an attempt to prevent the spread of false rumors by analyzing the phenomenon of Twitter hoaxes. Aramaki et al. used Twitter to predict the spread of influenza [3]. In addition, Twitter has been considered an effective tool in information transmission during emergencies, such as the Great East Japan Earthquake which occurred on March 11, 2011.

With regards to the above, appropriate selection of information is important especially when it comes to obtaining information in times of emergency and making decisions based on such information. Much of information appearing on Twitter contains private opinions about a variety of topics. This also includes various hoax tweets and false rumors unrelated to the general topic and mixed into the main thread. Therefore, a method for extracting only valid and useful tweets from a jumble of information on Twitter becomes essential. It is important to ensure the accuracy and the uniformity of the extracted information.

Moreover, when making decisions or when evaluating something, people are always subject to "cognitive bias" – psychological effects caused by external information, which hinders the perception of pure facts [4]. In situations of decision making on the basis of ambiguous information, the existence of cognitive bias causes a person's background to affect the final judgment through the "anchoring effect" (taking our background for granted). This causes the person to collect or remember only the information that is convenient for them, or to reinforce the prejudicial information, which is also called the "confirmation bias." The existence of the cognitive bias and related effects becomes a problem in situations of emergency or events of great importance, when obtaining the accurate and unbiased information is crucial for making appropriate judgments.

In this study, we perform automatic classification of tweets into three general types of information defined to appear on Twitter: primary, sequiary and secondary [5]. We compare the performance of the classification of multiple feature sets and a number of machine learning classifiers.

The outline of this paper is as follows. In Section 2 we present the core idea of information triage and the reasoning for its implementation on Twitter. In Section 3, we describe the proposed approach, including different data, preprocessing methods, the proposed classifier as well as other classifiers applied in experiment for comparison. In Section 4 we present the analysis and classification results for the analyzed tweet logs and confirm the validity of the proposed method. Finally, we conclude the paper in Section 5.

2 Information Triage on Twitter

The task of classification of information according to its importance and urgency is called *information triage* [6]. In cases when a task cannot be fully completed due to the limitations in time and resources, information triage becomes an important task helping determine the priority of information according to certain criteria.

In the previous study, Fukushima, et al. [5], performed a preliminary study using a sample of tweet logs from the time of the Great East Japan Earthquake. The basic tweet classification rules defined in the preliminary study were further used to classify other tweets by dividing them into representing either primary or secondary information. Primary information refers to the kind of information that a person directly saw, heard or personally did. Secondary information refers to indirect information such as re-posting or re-telling

[1] https://twitter.com/

Table 1: Definition of classification criteria for the three types of information on Twitter.

Tweet type	Primary	Sesquiary	Secondary
- Factual Information - Description of an action - Decisive expressions - Interview contents - Policy	○ ○ ○ ○ ○		
- Expression of an intention - Emotional expressions - Opinions - A call to action - Introduction of an URL link		○ ○ ○ ○ ○	
- Official RT - Things seen on TV (incl. facts) - Expressions indicating a rumor - Written reproduction of original information - Citations			○ ○ ○ ○ ○

what was described by someone else (third party), such as describing friend's opinions about books, or what someone saw on TV.

Although the two original types of information (primary and secondary), appeared the most frequently during the earthquake, there was a large chunk of information for which it was impossible to apply the initial criteria for binary classification, and was classified as "other." To optimize the criteria, Fukushima, et al. [5] additionally analyzed tweets from a different major event requiring a decision making process, namely, tweets that appeared during the general elections, on December 2012. They used the tweets about the general elections because they differed in the required type of information considered as important. In the disaster tweets, the factual information was the most important. In the election tweets, users often write about their political preferences, thus it was also important to take into consideration information from the borderline of pure fact and rumor, such as opinions, or attitudes. This kind of information in the Earthquake tweets is mostly considered as noise. However, in election tweets, private opinions and emotional comments could become useful as referential information. Therefore it is important to distinguish this kind of information from the rest and annotate it separately. To do this, Fukushima et al. defined a third kind of information which was neither primary nor secondary, though keeping a structure of its own, namely, "sesquiary" (from Latin "sesqui-" = 1.5) information.

Automatic classification and dynamic switching through the above three types of information could help effectively provide information needed by users at the moment, which could be helpful in emergency situations such as disasters.

The detailed criteria for classification of primary, sesquiary and secondary information were represented in Table 1. In this research, we applied these criteria to prepare the dataset used further in developing a Deep Learning-based model for automatic classification of information appearing on Twitter.

3 Approach Description

We applied the criteria described in section 2 to collect the datasets containing tweets representing each type of information. We applied these datasets to train and test a classifier for the optimal performance in distinguishing the three types of information. In the experimental phase, we used seven different classifiers with additional parameter modifications for comparison and to chose the best performing classifier. Moreover, we tested eleven ways of data preprocessing to further optimize the classifier performance.

3.1 Data Preprocessing

In this research, we focus on preprocessing datasets in the Japanese language. We used MeCab[2], a Japanese morphological analyzer, CaboCha[3], a Japanese dependency structure analyzer, and ASA, an argument structure analyzer[4], to preprocess the dataset in the following ways:

- **Tokenization:** Words, punctuation marks, etc. separated by spaces (later: TOK).
- **Lemmatization:** Like above but words represented in generic (dictionary) forms, or "lemmas" (LEM).
- **Parts of speech:** Words replaced with parts of speech (POS).
- **Tokens with POS:** Words and POS information integrated in each element (TOK+POS).
- **Lemmas with POS:** Like above but lemmas instead of words (LEM+POS).
- **Tokens with Named Entity Recognition:** Words integrated with named entities (private name of a person, organization, numerals, etc.). The NER information annotated by CaboCha (TOK+NER).
- **Lemmas with NER:** Like above but with lemmas (LEM+NER).
- **Chunking:** Larger sub-parts of sentences separated syntactically (CHNK).
- **Dependency structure:** Chunks with added information on syntactical relations between them (DEP).
- **Chunking with NER:** NER information integrated in chunks (CHNK+NER).
- **Dependency structure with NER:** Dependency relations and NER integrated in each element (DEP+NER).
- **Semantic Roles:** Words and phrases are replaced with their semantic role representations within sentence context. (SEM).
- **Morphosemantic Structure:** The sentences are preprocessed using combined morphological and semantic information. (MS).

From each of the thirteen dataset versions, a Bag-of-Words language model was generated, producing eleven different models (Bag-of-Words/Tokens, Bag-of-Lemmas, Bag-of-POS, Bag-of-Chunks, etc.). Weights of the features were calculated with traditional term frequency multiplied by inverse document frequency (tf*idf).

3.2 Classification Methods

In the comparison of classification methods, we applied the following seven classifiers.

Naïve Bayes classifier, traditionally used as a baseline in text classification tasks, applies Bayes theorem with the assumption of a strong (naive) independence between features.

k-Nearest Neighbors (kNN) classifier takes as input k-closest training samples with assigned classes and classifies input sample by a majority vote. Here, we used k1.

JRip also known as Repeated Incremental Pruning to Produce Error Reduction (RIPPER) [7], which learns rules incrementally to further optimize them. It has been especially

[2] http://taku910.github.io/mecab/

[3] http://taku910.github.io/cabocha/

[4] http://www.cl.cs.okayama-u.ac.jp/study/project/asa/

effective in the classification of noisy text [8]

J48 is a decision tree algorithm [9], which firstly builds decision trees from a labeled dataset and each tree node selects the optimal splitting criterion further chosen to make the decision.

Random Forest in training phase creates multiple decision trees to output the optimal class (mode of classes) in classification phase [10]. An improvement of RF to standard decision trees is their ability to correct over-fitting to the training set [11].

SVM or support-vector machines [12] represent data, belonging to specified categories, as points in space, and find an optimal hyperplane to separate the examples from each category. We used four types of kernel functions, namely, linear, polynomial, radial basis function, and sigmoid, in particular, hyperbolic tangent function [13].

CNN or Convolutional Neural Networks are an improved type of a feed-forward artificial neural network. Although originally designed for image recognition, CNNs proved their performance in many tasks, including NLP [14] and sentence classification [15].

We applied implementation of Convolutional Neural Networks with Rectified Linear Units (ReLU) as the neuron activation function [16], and max-pooling [17], which applies a max filter to non-overlying sub-parts of the input to reduce dimensionality and in effect correct over-fitting by down-sampling input representation. Moreover, we applied dropout regularization on penultimate layer, which prevents co-adaptation of hidden units by randomly omitting (dropping out) some of the hidden units during training [18].

We applied two versions of CNN. First, with one hidden convolutional layer containing 100 units was applied as a proposed baseline. Second, the final proposed method consisted of two hidden convolutional layers, containing 20 and 100 feature maps, respectively, both layers with a 5x5 size of the patch and 2x2 max-pooling, and Stochastic Gradient Descent [19] for weight optimization.

4 Evaluation Experiment

4.1 Datasets

In the experiment, we applied the data collected in previous research for manual analysis of information types appearing on Twitter [5], where the authors analyzed two types of situations: the time of and after a natural disaster (earthquake) and the period before elections. On this basis, they specified the criteria for classifying tweets as containing each type of information (primary, sesquiary, and secondary). In the experiment, we included samples representing each of the three types of information. However, each type of situation (disasters and elections) could have in reality different ratio of tweets representing the specified criteria. Thus, to make the experiment reveal how a classifier deals with the data in an objective and unbiased way, we randomly extracted 100 tweet samples of each kind of information for each analyzed situation. We decided to normalize the number of samples to eliminate any bias in the data. This provided 600 samples. Moreover, we prepared additional new 300 samples (100 samples per each class type) from another disaster, namely, the eruption of a volcano on Mt. Ontake, on September 27th, 2014, which provided us with overall 900 tweet samples.

Table 2: Results of all applied classifiers (Scores averaged for *primary*, *sesquiary*, and *secondary* prediction calculated separately; best classifier for each dataset in **bold type fond**; best dataset generalization for each classifier – underlined).

		LEM +POS	TOK +POS	LEM	TOK	CHNK +NER	POS	DEP +NER	DEP	CHNK	LEM +NER	TOK +NER	MS	SEM
SVM linear	P	.725	.734	.731	.739	.636	.389	.571	.611	.643	.636	.668	.503	.452
	R	.728	.737	.733	.741	.564	.398	.498	.510	.557	.637	.670	.536	.521
	F1	.725	.734	.732	.739	.545	.370	.452	.465	.534	.634	.667	.482	.437
	A	.728	.737	.733	.741	.564	.398	.498	.510	.557	.637	.670	.536	.521
SVM polynomial	P	.446	.446	.446	.446	.111	.297	.111	.111	.111	.221	.111	.474	.124
	R	.346	.346	.347	.344	.333	.344	.333	.333	.333	.333	.333	.380	.352
	F1	.192	.192	.194	.190	.167	.192	.167	.167	.167	.213	.167	.261	.184
	A	.346	.346	.347	.344	.333	.344	.333	.333	.333	.333	.333	.380	.352
SVM radial	P	.745	.744	.747	.758	.599	.405	.546	.524	.584	.622	.714	.356	.476
	R	.608	.603	.606	.611	.421	.409	.408	.377	.427	.475	.514	.530	.526
	F1	.592	.587	.591	.597	.334	.390	.316	.258	.337	.387	.475	.423	.441
	A	.608	.603	.606	.611	.421	.409	.408	.377	.427	.475	.514	.530	.526
SVM sigmoid	P	.746	.749	.742	.746	.737	.399	.633	.671	.735	.542	.728	.356	.508
	R	.572	.577	.559	.566	.388	.402	.417	.399	.392	.352	.509	.530	.530
	F1	.549	.555	.533	.541	.274	.364	.324	.294	.282	.236	.466	.423	.455
	A	.572	.577	.559	.566	.388	.402	.417	.399	.392	.352	.509	.530	.530
Naïve Bayes	P	.680	.681	.666	.669	.608	.412	.606	.705	.623	.671	.659	.457	.459
	R	.681	.681	.670	.671	.567	.417	.507	.502	.541	.670	.660	.501	.511
	F1	.664	.665	.652	.651	.535	.405	.448	.419	.502	.662	.650	.468	.463
	A	.681	.681	.670	.671	.567	.417	.507	.502	.541	.670	.660	.501	.511
JRip	P	.721	.734	.757	.782	.737	.371	.620	.662	.648	.719	.721	.490	.523
	R	.707	.708	.724	.732	.388	.348	.423	.477	.442	.687	.689	.512	.526
	F1	.695	.697	.713	.718	.274	.310	.322	.382	.344	.669	.670	.479	.478
	A	.707	.708	.724	.732	.388	.348	.423	.477	.442	.687	.689	.512	.526
J48	P	.727	.730	.747	.741	.622	.415	.538	.353	.626	.735	.708	.492	.488
	R	.722	.728	.743	.737	.404	.414	.412	.481	.502	.735	.710	.504	.505
	F1	.723	.728	.744	.738	.295	.413	.310	.376	.430	.733	.708	.494	.482
	A	.722	.728	.743	.737	.404	.414	.412	.481	.502	.735	.710	.504	.505
kNN (k=1)	P	.620	.623	.623	.610	.610	.412	.586	.726	.683	.554	.540	**.502**	.503
	R	.610	.609	.613	.594	.503	.417	.473	.357	.458	.539	.527	**.511**	.516
	F1	.612	.611	.617	.597	.450	.412	.405	.218	.396	.525	.520	**.502**	.501
	A	.610	.609	.613	.594	.503	.417	.473	.357	.458	.539	.527	**.511**	.516
Random Forest	P	.760	.740	.764	.768	.622	**.421**	.557	.635	.652	.746	.781	.499	**.517**
	R	.756	.739	.758	.764	.560	**.422**	.487	.500	.570	.737	.774	.510	**.526**
	F1	.745	.725	.749	.757	.540	**.420**	.442	.417	.537	.731	.772	.502	**.519**
	A	.756	.739	.758	.764	.560	**.422**	.487	.500	.570	.737	.774	.510	**.526**
CNN (1 hidden)	P	.770	.787	.769	.781	.563	.418	.516	.550	.601	.762	.764	.458	.462
	R	.769	.787	.769	.782	.556	.420	.507	.526	.587	.761	.764	.519	.537
	F1	.766	.785	.767	.781	.558	.417	.504	.517	.585	.760	.763	.469	.457
	A	.769	.787	.769	.782	.556	.420	.507	.526	.587	.761	.764	.519	.537
CNN (2 hidden)	P	**.939**	**.893**	**.919**	**.862**	**.914**	.333	**.987**	**.818**	**.990**	**.913**	**.910**	.333	.352
	R	**.939**	**.884**	**.919**	**.840**	**.910**	.333	**.987**	**.781**	**.990**	**.910**	**.906**	.333	.352
	F1	**.939**	**.886**	**.919**	**.842**	**.910**	.306	**.987**	**.779**	**.990**	**.910**	**.906**	.322	.352
	A	**.939**	**.884**	**.919**	**.840**	**.910**	.333	**.987**	**.781**	**.990**	**.910**	**.906**	.333	.352

4.2 Experiment Setup

The goal of the experiment was to select the best performing classifier with its optimal parameters, and the most adequate data preprocessing method. We applied a 10-fold cross validation on all of the balanced datasets used together. We compared in detail the top three performing classifiers and checked whether the differences between them are statistically significant. We also looked in detail at the errors made by the best classifier to see if these represent a structure unified enough to be systematically dealt with in the future. The results of the experiment were calculated using standard Precision, Recall, balanced F-score and Accuracy. As for the winning condition, we looked at which classifier achieved highest balanced F-score, with a confirming condition of higher Accuracy in case of two equally performing classifiers.

4.3 Results and Discussion

The classifiers can be divided into three groups, denning on by their results. The first, represented by simple classifiers, such as kNN or Naïve Bayes, obtained the lowest results. Also, SVMs using polynomial, radial, and sigmoid functions

Table 3: Contingency tables for top-three classifiers.

	2-layer CNN / shallow parsing (chunks)			
	classified as →	**primary**	**secondary**	**sesquiary**
correct	**primary**	298	1	1
	secondary	2	297	1
	sesquiary	4	0	296
	2-layer CNN / deep parsing with named entities			
	classified as →	**primary**	**secondary**	**sesquiary**
correct	**primary**	299	1	0
	secondary	6	292	2
	sesquiary	2	1	297
	2-layer CNN / lemmas with parts-of-speech			
	classified as →	**primary**	**secondary**	**sesquiary**
correct	**primary**	291	6	3
	secondary	10	275	15
	sesquiary	10	11	279

fit in this group, with polynomial SVMs scoring the lowest of all used classifiers.

The second group of classifiers contains linear SVM, JRip and Random Forest, as well as CNN with one hidden layer. Interestingly, from this mediocre scoring group, the simple CNNs usually scored highest, with Random Forest as the second best in this group.

Random Forest also scored highest of all for the dataset using only part-of-speech and semantic features, which were the most problematic for all classifiers. Unfortunately, although Random Forest scored for these dataset as highest, the score was still very low, below or around 50% of F-score.

Finally, he highest scoring classifier of all was the one based on Deep Convolutional Neural Networks with two hidden layers, which scored as the highest for all dataset preprocessing methods (except POS and semantic features). For most datasets, the 2-hidden layer CNN scored over 90% outperforming all other classifiers.

When it comes to the best performing feature set, simple tokenized dataset, tokens with either POS or NER, achieved the highest scores for most classifiers. Lemmatized dataset also scored highest twice for kNN and J48.

The highest combination of appropriate dataset preprocessing with classifier parameters belonged to the proposed 2-layer CNN with shallow parsing features. This version of the classifier obtained nearly perfect 99% for all used metrics. The second, and third best were respectively, also 2-layer CNN, but with feature sets based on dependency relations with named entities (F1=.987), and lemmas with POS (F1=.939). All results were summarized in Table 2.

As for statistical properties of the three best classifiers, at first we calculated Cohen's kappa statistic values for all three classifiers, based on their agreement with expected values, represented in contingency tables (see Table 3). Beginning from the worst, the kappa values were, κ=0.9083, κ=0.98, κ=0.985. For all classifiers, the strength of agreement was considered to be 'very good.'

As the final step of the analysis, we performed an analysis of most common types of errors the proposed classifier made. From Table 3 one can see that, when the classifier made a mistake, it most often annotated a tweet as "primary." This could suggest that the "primary" class has a tendency to be stronger in general, and even when a tweet expresses an opinion, it is mistakenly considered as primary information. Although six mistakes of this kind for 300 cases is not much (2%), we will focus on optimizing the criteria for primary information in the future.

5 Conclusions

In this paper, we presented our study in comparison of various machine learning classifiers for classification of three information types on Twitter: primary, sesquiary and secondary. To distinguish the information types on Twitter, we applied classification criteria manually developed by Fukushima et al.[5]. On the basis of these criteria, we collected additional messages related to the eruption of the Mt. Ontake volcano.

For future development of a system for information triage on Twitter for major events, such as disasters, we compared eleven classifiers with thirteen different feature sets and found out that the optimal combination was the 2-layer CNN trained on a dataset containing shallow parsing features.

In the near future, we plan to perform a deeper study of change in time and changes according to situation (when users are in the need of different kinds of information). As the final goal, we will implement the optimized version of the classifier into the system, and apply it to an online real-time disaster monitoring platform.

References

[1] Kuwano, T., Mitamura, T., Watanabe, I., Suzuki, Y., Oobori, T. (2012). The Study of Tourism Informatics Using Twitter. *Tourism Information Society Journal*, Vol.8, No.1, pp.27-38.

[2] Ayana Umejima, Mai Miyabe, Eiji Aramaki, Akiyo Nadamoto. (2011). Tendency of Rumor and Correction Re-tweet on the Twitter During Disasters [in Japanese]. *IPSJ SIG Notes*, 2011-DBS-152(4), 1-6, 2011-07-26.

[3] Eiji Aramaki, Sachiko Maskawa, Mizuki Morita. (2011). Twitter Catches the Flu: Detecting Influenza Epidemics using Factuality Analysis [in Japanese]. *IPSJ SIG Notes*, 2011-SLP-86(1), 1-8.

[4] D. Kahneman and A. Tversky. (1972). Subjective probability: A judgment of representativeness. *Cogn. Psychol.*, pp. 430-454.

[5] Yuto Fukushima, Fumito Masui, Michal Ptaszynski. (2014). Classification of Tweet Logs Based on Directness Derived from Surface Expressions [in Japanese]. *In Proceedings of The 28th Annual Meeting of the Japanese Society for Artificial Intelligence*.

[6] Catherine C. Marshall, Frank M. Shipman, III. (1997). Spatial hypertext and the practice of information triage, *HYPERTEXT'97*, pp. 124-133.

[7] Cohen, W. W. (1995). Fast effective rule induction. In Machine Learning Proceedings 1995 (pp. 115-123).

[8] Sasaki, M., & Kita, K. (1998). Rule-based text categorization using hierarchical categories. *International Conference on Systems, Man, and Cybernetics*, Vol. 3, pp. 2827-2830.

[9] Quinlan,J.R.(2014).C4.5:programs for machine learning.Elsevier.

[10] Breiman, L.(2001).Random forests.*Machine learning*,45(1),5-32.

[11] Hastie, T. and Tibshirani, R. and Friedman, J. (2013). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Series in Statistics.

[12] Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine learning, 20(3), 273-297.

[13] Lin, H. T., & Lin, C. J. (2003). A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods. submitted to Neural Computation, 3, 1-32.

[14] Collobert, R., & Weston, J. (2008, July). A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th international conference on Machine learning (pp. 160-167). ACM.

[15] Yoon Kim. (2014). Convolutional Neural Networks for Sentence Classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pp. 1746-1751.

[16] Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th international conference on machine learning (ICML-10) (pp. 807-814).

[17] Scherer, D., Mller, A., & Behnke, S. (2010, September). Evaluation of pooling operations in convolutional architectures for object recognition. In International conference on artificial neural networks (pp. 92-101). Springer, Berlin, Heidelberg.

[18] Hinton, Geoffrey E. and Srivastava, Nitish and Krizhevsky, Alex and Sutskever, Ilya and Salakhutdinov, Ruslan. (2012). Improving neural networks by preventing co-adaptation of feature detectors. CoRR, abs/1207.0580.

[19] LeCun, Y., Bottou, L., Orr, G. B., & Mller, K. R. (1998). Efficient backprop. In Neural networks: Tricks of the trade (pp. 9-50). Springer, Berlin, Heidelberg.