Neural Networks and Support Vector Machine based Approach for Classifying Tweets by Information Types at TREC 2018 Incident Streams Task

Abu Nowshed Chy, Umme Aymun Siddiqua, and Masaki Aono
Department of Computer Science and Engineering
Toyohashi University of Technology
Toyohashi, Aichi, Japan
[nowshed, aymun]@kde.cs.tut.ac.jp and aono@tut.jp

Abstract: Microblog, especially twitter, is treated as an important source to serve the situational information needs during a disaster period. Monitoring and producing the curated tweets based on different information types from massive twitter posts provide enormous opportunities to different public safety personnel or used for post-incident analysis. In this paper, we present our approach to addressing the problem defined in the TREC 2018 incident streams (TREC-IS) task. The task is to classify the tweets in each event/incident's stream into different high-level information types within the incident ontology. In our approach, we employ different deep neural network (DNN) classifiers in combination with a multi-class support vector machine (SVM) classifier and a rule-based classifier. We consider a rich set of hand-crafted features to train our multi-class SVM classifier, whereas a pre-trained word2vec model is used for the DNN based classifiers. Moreover, we introduce a set of rules based on the language of tweets, exploiting indicator terms, and WH-orientation of tweets for our rule-based classifier. Experimental results showed that our proposed KDEIS4_DM method obtained the second position among the participants in TREC-IS task and outperforms the participant median by more than 8% and 5% in terms of F1 Score and accuracy, respectively.

Keywords: TREC incident streams, microblog retrieval, disasters, TREC-IS, deep neural network (DNN), DeepMoji, attention mechanism, SVM, supervised learning, hand-crafted features, information types classification.

1. Introduction

Microblog sites such as twitter, tumblr, sina weibo, etc. are rapidly moving towards a platform for mass collaboration in user-generated information production. Twitter has become the most popular among the microblog services. Everyday lots of people turning to this online platform to share their views, opinions, breaking news as well as fulfill their diverse information needs. The real-time nature of twitter plays an important role during a disaster period, such as earthquake, floods, wildfires, and typhoons. Because the user-generated twitter posts during such events might be useful to serve the situational information needs [1]. However, due to the brevity of the tweets and noisy tweet contents, information retrieval in twitter is regarded as a challenging IR problem. To address the general real-time information seeking behaviors, TREC was introduced the microblog ad-hoc search task in 2011 [2]. In contrast, this year TREC-2018 introduces an incident streams (TREC-IS) task designed specifically to tackle the microblog retrieval during a disaster period. The main task for the 2018 TREC-IS track was to categorize the tweets in each event/incident's stream into different high-level information types defined in the TREC-IS incident ontology.

In this paper, we present our participation in the 2018 TREC-IS task. We combine different types of classifiers in our proposed approach. We define a set of rules for the rule-based classifier by focusing on the language of tweets, exploiting indicator terms from the training corpus, and WH-orientation of tweets. We consider lexical and content relevance features, incident and event related features, sentiment aware, and twitter specific features to train our multi-class SVM classifier, whereas a pre-trained *word2vec* model is used for the deep neural network (DNN) classifiers.

The rest of the paper is structured as follows: We will introduce our proposed TREC-IS framework in **Section 2**. **Section 3** includes experiments and evaluation to show the effectiveness of our proposed methods. Some concluded remarks and future directions of our work described in **Section 4**.

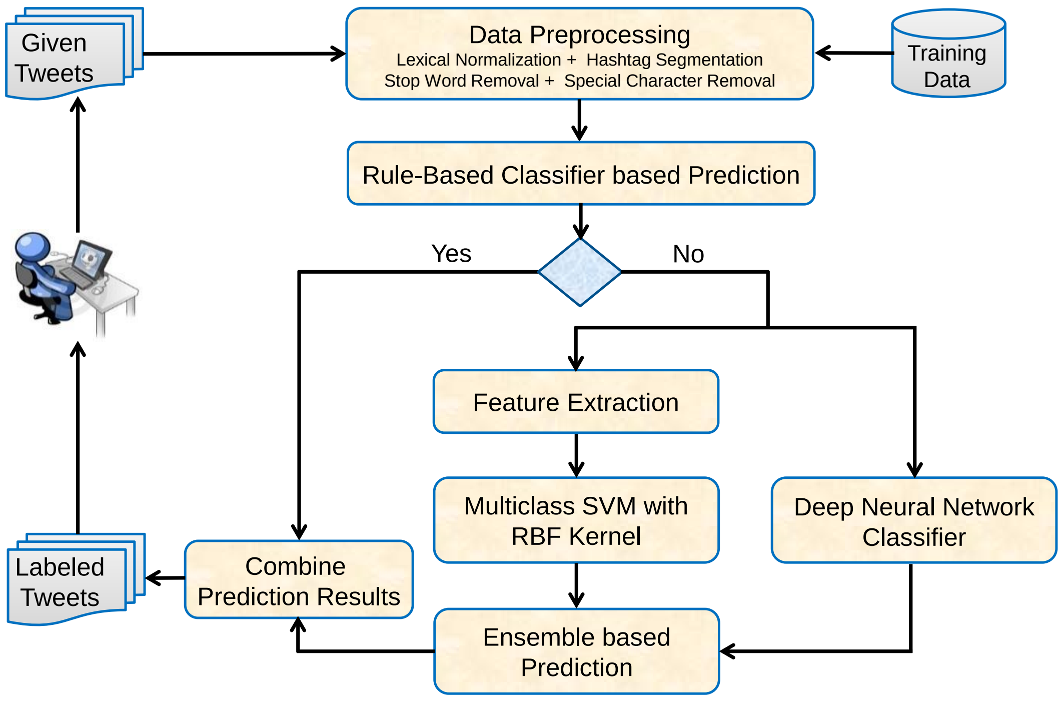

Given Tweets
Data Preprocessing
Lexical Normalization + Hashtag Segmentation
Stop Word Removal + Special Character Removal
Training Data
Rule-Based Classifier based Prediction
Yes
No
Feature Extraction
Multiclass SVM with RBF Kernel
Deep Neural Network Classifier
Labeled Tweets
Combine Prediction Results
Ensemble based Prediction

Fig. 1 Proposed TREC incident streams (TREC-IS) framework.

2. Proposed TREC-IS Framework

In this section, we describe the details of our proposed framework. Given a query related to an event/incident and a set of tweets, the goal of our proposed system is to categorize the tweet into the different high-level information types. The overview of our proposed framework depicted in Fig. 1.

At first, our system fetches a query and the corresponding tweet set as a single batch and indexed them for further processing. In the data preprocessing stage, we perform the tokenization, lexical normalization to the tokenized words, stop-word removal, special character removal, and hashtag segmentation. Next, our proposed rule-based classifier is applied to classify the tweets into the corresponding high-level information types. For the tweets that are not classified by the rule-based classifier, we consider the combined weighted prediction score from multi-class support vector machine (SVM) classifier and several deep neural network (DNN) classifiers. SVM classifier is trained with our extracted features. We extract several effective features broadly grouped into four different categories, including lexical and content relevance features, incident and event related features, sentiment aware features, and twitter specific features. For extracting sentiment aware features, we construct strong sentiment lexicons by combining several publicly available sentiment lexicons. To scale the feature values, we make use of the *Min-Max* normalization technique. For DNN based classifiers, a pre-trained *word2vec* model is applied. Tweets are labeled to the information type that gets the highest prediction score. Results of both the rule-based classifier and the ensemble of SVM and DNN classifiers are then combined and the set of labeled tweets return to the user.

2.1 Data Preprocessing

Data preprocessing stage is initiated with tokenization. As tweets are informal user generated contents, people use lots of non-English characters and symbols in tweets. Since meaningful English words do not contain these characters, we remove these characters from tweets. Moreover, the short length constraint of the tweet makes characters expensive. To overcome this constraint, people are utilizing twitter specific syntaxes such as #hashtag to express their thoughts concisely. For #hashtag removal, we segment each #hashtag by using a hashtag segmentation technique similar to Siddiqua et al. [3] and replaced the hashtag with the segmented words.

Moreover, tweets often contain non-standard word forms and domain-specific entities. For example, people usually use "earthquakeeeee" instead of "earthquake," "addiquate" instead of "adequate," "appt" instead of "appointment," etc. We utilized two lexical normalization dictionaries collected from [4] and [5] to normalize such non-standard words into their canonical forms. In incident streams task, stopwords play a negative role because they do not carry any incident-oriented information and may actually damage the performance of the classifiers. For stopword removal, we applied the Indri's standard stoplist[*1].

[*1] https://www.lemurproject.org/stopwords/stoplist.dft

Table 1 List of features used in this work.

Feature Type	Feature Name
Lexical and Content Relevance Features	(1) TF-IDF [6] similarity score between an incident query and a tweet. (2) Okapi BM25 [7] similarity score between an incident query and a tweet. (3) Language model with Dirichlet smoothing [8] score between an incident query and a tweet. (4) Tweet Length Feature: Number of words available in a tweet. (5) Average Word Length Feature: Average length of the words available in a tweet.
Incident and Event Related Features	(1) Location Count Feature: Number of locations name available in a tweet. (2) Organization Count Feature: Number of organizations name available in a tweet. (3) Person Count Feature: Number of person information available in a tweet. (4) Noun Count Feature: Number of noun POS available in a tweet. (5) Phone Number Count Feature: Number of phone number available in a tweet. (6) Known Already Count Feature: Number of previously posted tweets that are closely matched (based on Cosine Similarity) with the corresponding tweet.
Sentiment Aware Features	(1) Sentiment Polarity Feature: A binary feature that is assigned to 1 if a tweet has the positive or negative sentiment polarity and 0 otherwise. (2) Positive Word Count Feature: Number of positive words available in a tweet based on the lexicon. (3) Negative Word Count Feature: Number of negative words available in a tweet based on the lexicon. (4) Emoticon Count Feature: Number of emoticons available in a tweet.
Twitter Specific Features	(1) Hashtag Feature: A binary feature that is assigned to 1 if a tweet contains a hashtag and 0 otherwise. (2) Hashtag Count Feature: Number of hashtags available in a tweet. (3) URL Feature: A binary feature that is assigned to 1 if a tweet contains a URL and 0 otherwise. (4) Retweet Feature: A binary feature that is assigned to 1 if a tweet is a retweet of the other tweet and 0 otherwise.
Total	19 Features

2.2 Feature Extraction

In our proposed framework, we extract a set of 19 features broadly grouped into 4 different categories, including lexical and content relevance features, incident and event related features, sentiment aware features, and twitter specific features. The feature extraction processes are described in **Table 1**.

The first 3 lexical and content relevance features are used to estimate the similarity between a given incident query and a tweet. In this regard, we generate the incident query by combining the query title and narrative and perform the minimal preprocessing as described in Section 2.1. We also extract 6 incident event related features that seem to be important during the disaster situation. We utilize the Stanford named entity recognizer (NER) tool [9] to extract the location count, organization count, and person count features. Along with this direction, a publicly available library is utilized to estimate the phone number count feature. We also use the CMU ARK POS tagger [10] to identify the noun POS of each tokenized word which is required to extract the noun count feature. To estimate the sentiment polarity of a tweet, we use a publicly available package SentiStrength [11]. We construct the positive and negative sentiment bearing word lexicons as described in [12]. We utilize these lexicons to estimate the lexicon based sentiment aware features. For emoticon count feature, we use a publicly available library to identify the emoticon. Other features are extracted as described in Table 1.

2.3 Rule-Based Classifier

In a rule-based classifier, we usually construct a set of rules that determine a certain combination of patterns, which are most likely to be related to the different classes or information types. Each rule consists of an antecedent part and a consequent part. The antecedent part corresponds to the patterns and the consequent part corresponds to a class label. We can define a rule as follows:

$$R_j : \text{if } x_1 \text{ is } A_{j1} \text{ and } \ldots\ldots\ldots\ x_n \text{ is } A_{jn}$$
$$\text{then } Class = C_j, \quad j = 1, \ldots\ldots, N$$

where R_j is a rule label, j is a rule index, A_{j1} is an antecedent set, C_j is a consequent class, and N is the total number of rules.

Our unsupervised rule-based classifier casts the TREC incident streams (TREC-IS) task as a multi-class classification problem and labeled each tweet to the corresponding information types assigned by the rules. To achieve this, we define a set of rules based on the tweets language, indicator terms within tweets, and WH-orientation of the tweet. Descriptions of each set of rules are presented next.

2.3.1 Language Related Rule

Even though twitter is a multilingual microblog service, we only consider English tweets as relevant in this research. Therefore, we define a rule based on the language of a tweet that is if the language of a tweet is not English, we classify the tweet as *Irrelevant* information type. To identify the non-English tweets from the given tweet set, we apply a language detection library[*2] in our system.

2.3.2 Indicator Terms based Rule

A tweet may contain some highly influential indicator terms related to a high-level information type which may useful to categorize the tweet into that information type. In this regard, we exploit the indicator terms given in the training data. We prepare two curated indicator terms lexicon based on the given indicator terms of several information types. One for the *MultimediaShare* category and other for the *Donations* category. If a tweet contains words from these lexicons, it is classified to the corresponding information type. The priority of the information type is determined by the number of lexicon words available in the tweet.

2.3.3 WH-Orientation based Rule

Since people usually use WH sentence to know more about the incident, we use the regular expression to identify the WH-orientation of a tweet and categorize the tweet into the *InformationWanted* information type.

2.4 An Ensemble of Learning Approach

2.4.1 Support Vector Machine (SVM) Classifier

We use the $SVM^{multiclass}$ with RBF kernel from [13]. It uses the multi-class formulation described in [14]. For a training set $(x_1, y_1)...(x_n, y_n)$ with labels y_i in $[1..k]$, it finds the solution of the following optimization problem during training:

$$min 1/2 \sum_{i=1..k} w_i * w_i + C/n \sum_{i=1..n} \xi_i$$

s.t. for all y in $[1..k]$:

$$[x_1 * w_{yi}] >= [x_1 * w_y] + 100 * \triangle(y_i, y) - \xi_1$$

$$\ldots\ldots..$$

s.t. for all y in $[1..k]$:

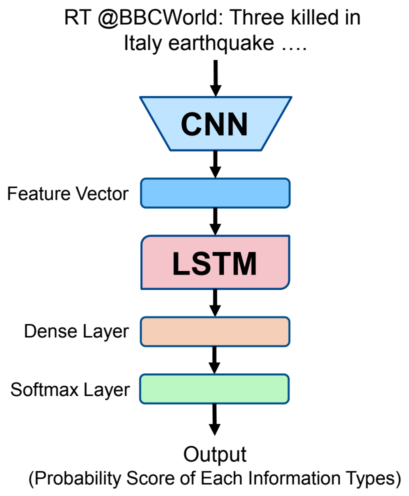
$$[x_n * w_{yn}] >= [x_n * w_n] + 100 * \triangle(y_n, y) - \xi_n$$

[*2] https://code.google.com/p/language-detection/

where C is the usual regularization parameter that trades off margin size and training error. We estimate the optimal value of C using cross-validation. $\triangle(y_n, y)$ is the loss function that returns 0 if y_n equals y, and 1 otherwise. To solve this optimization problem, $SVM^{multiclass}$ uses an algorithm based on structural SVMs. For the training, we use the features described in Section 2.2.

2.4.2 Deep Learning based Classifiers

Besides feature based multi-class SVM classifier, we employ the deep neural network (DNN) based classifier models because traditional bag-of-words based methods cannot perform well due to the curse of dimensionality and the loss of word order information. However, to train the deep learning models effectively, it is important to represent the tweets as meaningful features. To achieve this goal, we apply the CLSTM architecture inspired by the proposal of Zhou et al. [15].



RT @BBCWorld: Three killed in Italy earthquake ....
CNN
Feature Vector
LSTM
Dense Layer
Softmax Layer
Output
(Probability Score of Each Information Types)

Fig. 2 Convolutional long short-term memory (CLSTM) network.

In our CLSTM architecture as depicted in **Fig. 2**, the higher level representations of CNN are fed into the LSTM to learn long-term dependencies. The CNN is constructed on top of the pre-trained word vectors from *fastText* [16] to learn higher-level representations of n-grams. The feature maps of CNN are then organized as sequential window features to serve as the input of LSTM to learn sequential correlations from higher-level sequence representations. The LSTM transition functions are defined as follows:

$$\begin{aligned}
i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\
u_t &= \phi(W_u \cdot [h_{t-1}, x_t] + b_u) \\
c_t &= f_t \odot c_{t-1} + i_t \odot u_t \\
o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}$$

where i_t, f_t, o_t, u_t, c_t, and h_t denote the input gate, forget gate, output gate, cell input activation, the cell state, and the current hidden state, respectively, at the current time step t. The symbol σ is the logistic sigmoid function to set the gating values in [0, 1]. ϕ is the hyperbolic tangent activation function that has an output in $[1, -1]$ and $\odot$ is the element-wise multiplication.

At the last time step of LSTM, the output of the hidden state is regarded as the final tweet representation and passed to a fully connected softmax layer. The output of the softmax layer is the probability distribution over all the information types. To learn the model parameter, we utilize the stochastic gradient descent (SGD) and adopt the Adam optimizer [17].

However, unidirectional LSTM only preserves information of the past context. To understand the context better, bidirectional LSTM is used which runs forward and backward LSTM along with each input sequence and captures both past and future context. The basic idea of bidirectional LSTM is that the output at each time depends on the previous elements and the next elements in the sequence. In a bidirectional LSTM, two LSTMs are stacked on the top of each other. The one that processes the input in its original order and the other that processes the reversed input sequence. The output is then computed based on the hidden state of both LSTMs.

More recently, the attention mechanism has been introduced in the neural network models to mimic the human visual attention characteristic that is focused on a certain region of an image and adjusting the focal point over time. Rather than encoding the full source text, the attention mechanism allows the model to learn what to attend based on the input text.

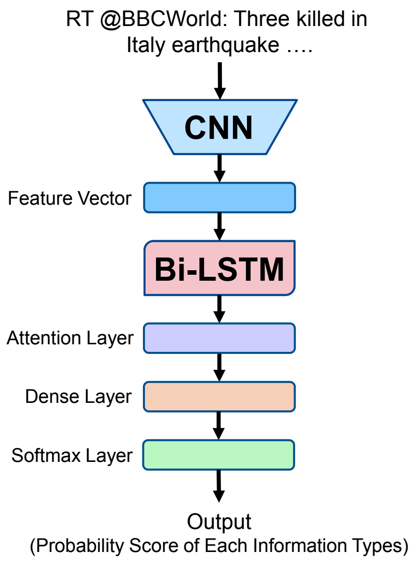

RT @BBCWorld: Three killed in Italy earthquake ....
CNN
Feature Vector
Bi-LSTM
Attention Layer
Dense Layer
Softmax Layer
Output
(Probability Score of Each Information Types)

Fig. 3 Attention based convolutional bidirectional LSTM (ACBLSTM) network.

In our ACBLSTM architecture as depicted in **Fig. 3**, the higher level representations of CNN are fed into the bidirectional LSTM to learn long-term dependencies. In order to amplify the contribution of important elements in the final representation of bidirectional LSTM, we employ a recently introduced attention mechanism [18], [19] to aggregate all the hidden states according to their relative importance.

In addition, we utilize the stacked bidirectional LSTM instead of a single bidirectional LSTM in our ACSBLSTM architecture. Our stacked bidirectional LSTM is comprised of $N = 15$ bidirectional LSTM layers, where each layer provides a sequence output to the next layer as depicted in **Fig. 4**.

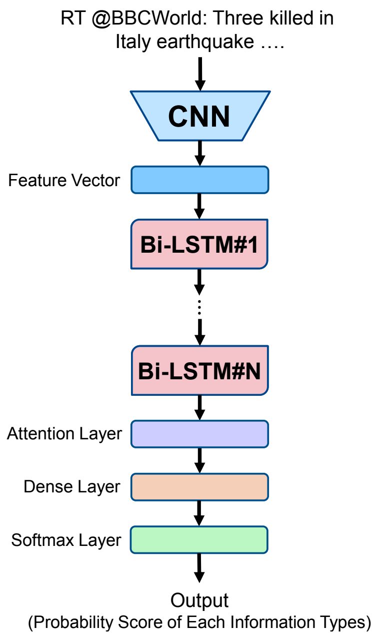

RT @BBCWorld: Three killed in Italy earthquake ....
CNN
Feature Vector
Bi-LSTM#1
Bi-LSTM#N
Attention Layer
Dense Layer
Softmax Layer
Output
(Probability Score of Each Information Types)

Fig. 4 Attention based convolutional stacked bidirectional LSTM (ACSBLSTM) network.

Next, we employ the state-of-the-art deep learning architecture DeepMoji (DM), proposed by Felbo et al. [20]. We train the DeepMoji architecture without loading the pre-trained weights. As depicted in **Fig. 5**, DeepMoji uses an embedding layer of 256 dimensions to project each word into a vector space. Two bidirectional LSTM layers with 1024 hidden units in each (512 in each direction) are applied to capture the context of each word. Finally, an attention layer takes all of these layers as input using skip-connections. The representation vector obtained from the attention layer is sent to the softmax layer for classification.

2.5 Combining the Classifiers

After developing our proposed rule-based classifier and training the deep neural network (DNN) based classifiers and support vector machine (SVM) classifier, we combine them to classify the tweet into the high-level information type. At first, our rule-based classifier is applied to classify the tweet to the corresponding information type. Tweets that are not classified by the rule-based

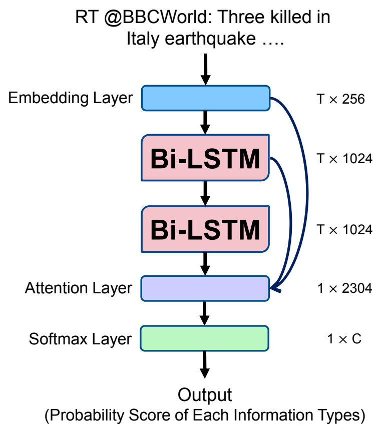


Fig. 5 DeepMoji (DM) network, where T is the tweet length and C is the number of classes.

classifier we consider the weighted ensemble based prediction from multi-class SVM classifier and several deep neural network models. The prediction score is computed using the following Equation 2.5.

$$P(C_i|T) = \alpha \cdot P(CL_{DNN}|T) + (1 - \alpha) \cdot P(CL_{SVM}|T)$$

where

$C_i \in$ {List of all high-level information types} and

$CL_{DNN} \in$ {CLSTM, ACBLSTM, ACSBLSTM, DM}

where, given a tweet T, the final relevance probability score $P(C_i|T)$ is estimated based on the prediction score from a deep neural network model denoted as $P(CL_{DNN}|T)$ and multi-class SVM classifier denoted as $P(CL_{SVM}|T)$. To select the optimal value for the anchoring parameter α, we swept the parameter between $\{0.1, ..., 0.9\}$. Information type that gets the highest probability score will be assigned to the label of the tweet.

3. Experiments and Evaluation

3.1 Dataset Collection

The TREC incident streams (TREC-IS) task at TREC-2018 provides a benchmark dataset to evaluate the performance of the proposed systems. The dataset contains 21 query topics along with the relevant tweets sampled from several disaster events such as earthquake, typhoon, shooting, etc. Among the 21 query topics, the training set contains 6 query topics and the test set contains 15 query topics. The number of tweets in the training set is around 1300, whereas the number of tweets in the test set is around 20,000. The organizer also provides an ontology of information types, which contains 25 information types or class label broadly grouped into Request, Report, CallToAction, and Other.

3.2 Evaluation Measures

To evaluate the performance of our proposed systems, we applied the evaluation measure used in the TREC-IS task. According to the benchmark of the 2018 TREC-IS task, participant systems were tasked to assign one most representative information type per-tweet. However, during the ground truth generation, the human assessors were allowed to select as many information types as appropriate for a single tweet. Therefore, to evaluate the performance of a TREC-IS system the organizer used two ways referred to as multi-type and any-type.

In the multi-type evaluation, the categorization performance per information type is estimated in a 1 vs. All manner. If both the system and human assessor selected the corresponding category then the system is considered to correctly categorize a tweet. Whereas, in the any-type evaluation, a system is considered to correctly categorize a tweet if it assigned any of the categories that the human assessor selected for that tweet. However, any-type evaluation criteria is used to estimate the overall performance of a 2018 TREC-IS system. Four standard evaluation metrics including precision, recall, F1 score, and accuracy were used in both the multi-type and any-type evaluation criteria.

3.3 Results with Different Experimental Settings

We now evaluate the performance of our proposed methods in this section. We describe the experimental settings of each method and the summarized evaluation results were presented in **Table 2**.

At first, our rule-based classifier is applied to classify the tweet into corresponding information type and tweets that are not clas-

Table 2 Performance (Precision, Recall, F1 Score, and Accuracy; higher is better) and priority estimation error (mean squared error; lower is better) on TREC-IS 2018 test set for various experimental settings. The best results are highlighted in boldface.

Method	Multi-type (Macro)				Any-type (Micro)				Priority Estimation Error
	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score	Accuracy	
KDEIS1_CLSTM	0.1388	0.0607	0.0620	0.8929	0.2575	0.9783	0.4077	0.2580	0.0842
KDEIS2_ACBLSTM	0.1512	0.0689	0.0703	0.8890	0.2089	0.9734	0.3440	0.2098	0.0842
KDEIS3_ACSBLSTM	0.1209	0.0577	0.0482	0.8933	0.2630	0.9788	0.4147	0.2635	0.0842
KDEIS4_DM	0.1482	0.0708	0.0734	**0.9035**	0.3914	**0.9856**	**0.5603**	**0.3908**	0.0842
Participant Median	**0.1827**	**0.0784**	**0.0825**	0.8993	**0.3978**	0.6164	0.4775	0.3385	0.0933

sified by the rule-based classifier, we consider the weighted ensemble based prediction from multi-class SVM classifier and the CLSTM (described in Section 2.4.2) architecture. The prediction score is computed according to Equation 2.5. We denoted this setting as KDEIS1_CLSTM. Next, we used the ACBLSTM deep learning architecture instead of CLSTM in the above setting and referred this setting as KDEIS2_ACBLSTM. Similarly, in the KDEIS3_ACSBLSTM and KDEIS4_DM settings, we consider the ACSBLSTM and DM based deep neural network architectures, respectively. We also reported the participant median results for comparison.

Results showed that our KDEIS4_DM setting achieved the best performance in both the multi-type and any-type evaluation criteria in terms of primary evaluation measure F1 score. For the any-type evaluation criteria, our system outperformed the participant median by more than 8% in terms of F1 Score and by more than 5% in terms of Accuracy.

We also reported the results of the top 5 performing systems in TREC-IS 2018 in **Table 3**. It showed that our KDEIS4_DM achieved the second position among the participants in terms of primary evaluation measure F1 score for the any-type evaluation.

Table 3 Top 5 performing systems (Precision, Recall, F1 Score, and Accuracy; higher is better) in TREC-IS 2018. Boldfaced one is our proposed system.

Method	Any-type (Micro)			
	Precision	Recall	F1 Score	Accuracy
cbnuS2	0.4559	0.7780	0.5749	0.4213
KDEIS4_DM	0.3914	0.9856	0.5603	0.3908
umdhcilfasttext	0.4534	0.7260	0.5582	0.4022
cbnuS1	0.4472	0.7402	0.5575	0.4064
NHK_run2	0.4483	0.7143	0.5509	0.3997

4. Conclusion and Future Directions

In this paper, we presented our approach to the TREC 2018 incident streams (TREC-IS) task. We tackled the problem by employing an ensemble of classifiers. Along with a rule-based classifier and SVM classifier, four different deep neural network (DNN) models were employed in several experimental settings. Among our proposed methods, KDEIS4_DM achieved the second best performance (F1 Score = 0.5603 for any-type evaluation) among the participant systems.

There is much room left to further improve our methods. Shortage of training dataset for our deep learning approach is the main problem. In the future, we have a plan to overcome this limitation by incorporating more training samples collected in an unsupervised manner. We also have a plan to exploit the more sophisticated techniques in our deep learning approaches.

Acknowledgments This research was supported by MEXT KAKENHI, Grant-in-Aid for Scientific Research (B), Grant Number 17H01746.

References

[1] Sakaki, T., Okazaki, M. and Matsuo, Y.: Tweet analysis for real-time event detection and earthquake reporting system development, *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, Vol. 25, No. 4, pp. 919–931 (2013).

[2] Ounis, I., Macdonald, C., Lin, J. and Soboroff, I.: Overview of the trec-2011 microblog track, *Proceedings of the 20th Text REtrieval Conference (TREC)*, NIST (2011).

[3] Siddiqua, U. A., Chy, A. N. and Aono, M.: Stance detection on microblog focusing on syntactic tree representation, *International Conference on Data Mining and Big Data (DMBD)*, Springer, pp. 478–490 (2018).

[4] Han, B., Cook, P. and Baldwin, T.: Automatically constructing a normalisation dictionary for microblogs, *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Association for Computational Linguistics (ACL), pp. 421–432 (2012).

[5] Liu, F., Weng, F. and Jiang, X.: A broad-coverage normalization system for social media language, *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL): Long Papers-Volume 1*, ACL, pp. 1035–1044 (2012).

[6] Leskovec, J., Rajaraman, A. and Ullman, J. D.: *Mining of massive datasets*, Cambridge University Press (2014).

[7] Robertson, S. E., Walker, S., Beaulieu, M. and Willett, P.: Okapi at TREC-7: automatic ad hoc, filtering, VLC and interactive track, *NIST Special Publication SP*, No. 500, pp. 253–264 (1999).

[8] Ponte, J. M. and Croft, W. B.: A language modeling approach to information retrieval, *Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, pp. 275–281 (1998).

[9] Finkel, J. R., Grenager, T. and Manning, C.: Incorporating non-local information into information extraction systems by gibbs sampling, *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL)*, Association for Computational Linguistics, pp. 363–370 (2005).

[10] Owoputi, O., O'Connor, B., Dyer, C., Gimpel, K., Schneider, N. and Smith, N. A.: Improved part-of-speech tagging for online conversational text with word clusters, Association for Computational Linguistics (ACL) (2013).

[11] Thelwall, M., Buckley, K. and Paltoglou, G.: Sentiment strength detection for the social web, *Journal of the American Society for Information Science and Technology (JASIST)*, Vol. 63, No. 1, pp. 163–173 (2012).

[12] Siddiqua, U. A., Ahsan, T. and Chy, A. N.: Combining a rule-based classifier with ensemble of feature sets and machine learning techniques for sentiment analysis on microblog, *19th International Conference on Computer and Information Technology (ICCIT)*, IEEE, pp. 304–309 (2016).

[13] Tsochantaridis, I., Hofmann, T., Joachims, T. and Altun, Y.: Support vector machine learning for interdependent and structured output spaces, *Proceedings of the 21st International Conference on Machine Learning (ICML)*, ACM, p. 104 (2004).

[14] Krammer, K. and Singer, Y.: On the algorithmic implementation of multi-class SVMs, *Proceedings of JMLR*, pp. pages 265–292 (2001).

[15] Zhou, C., Sun, C., Liu, Z. and Lau, F. C. M.: A C-LSTM neural network for text classification, *CoRR*, Vol. abs/1511.08630 (online), available from ⟨http://arxiv.org/abs/1511.08630⟩ (2015).

[16] Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C. and Joulin, A.: Advances in Pre-Training Distributed Word Representations, *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)* (2018).

[17] Kingma, D. and Ba, J.: Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980* (2014).

[18] Raffel, C. and Ellis, D. P.: Feed-forward networks with attention can solve some long-term memory problems, *arXiv preprint arXiv:1512.08756* (2015).

[19] Wang, N., Wang, J. and Zhang, X.: YNU-HPCC at IJCNLP-2017 Task 4: Attention-based Bi-directional GRU Model for Customer Feedback Analysis Task of English, *Proceedings of the IJCNLP 2017, Shared Tasks*, pp. 174–179 (2017).

[20] Felbo, B., Mislove, A., Søgaard, A., Rahwan, I. and Lehmann, S.: Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm, *Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2017).