Social Media based Real-time Event Detection by using Deep Learning Methods

Van Quan Nguyen*, Hyung-Jeong Yang**, Young-chul Kim**,
Soo-hyung Kim**, Kyungbaek Kim**

Abstract

Event detection using social media has been widespread since social network services have been an active communication channel for connecting with others, diffusing news message. Especially, the real-time characteristic of social media has created the opportunity for supporting ~~for~~ real-time applications/systems. Social network such as Twitter is the potential data source to explore useful information by mining messages posted by the user community. This paper proposed a novel system for temporal event detection by analyzing social data. As a result, this information can be used by first responders, decision makers, or news agents to gain insight of the situation. The proposed approach takes advantages of deep learning methods that play core techniques on the main tasks including informative data identifying from a noisy environment and temporal event detection. The former is the responsibility of Convolutional Neural Network model trained from labeled Twitter data. The latter is for event detection supported by Recurrent Neural Network module. We demonstrated our approach and experimental results on the case study of earthquake situations. Our system is more adaptive than other systems used traditional methods since deep learning enables to extract the features of data without spending lots of time constructing feature by hand. This benefit makes our approach adaptive to extend to a new context of practice. Moreover, the proposed system promised to respond to acceptable delay within several minutes that will helpful mean for supporting news channel agents or belief plan in case of disaster events.

■ keywords : Social Data|Deep Learning|Convolutional Neural Network|Long Short Term Memory|Event Detection

I. Introduction

Social networks have become the effective mean of communicating information. Millions of users can join the social network and share information what they witness, experience, or hear from other sources about different aspects of daily life. Those can range from personal status (i.e., opinions, emotions, thoughts.) to social event they witness (i.e., election, traffic jam) even disaster event. Thanks to naturally real-time property, users play a role as a social sensor, while posted messages are working as the response signals; therefore, social data can cover an event faster than the traditional news media. From such motivation, we proposed a novel system that can automatically detect the occurrence of a disaster event based on the social data stream and take advantages of deep learning techniques.

We address the disaster event detection problem in the social media context. Since we directly work with a noisy environment as the social network, our approach will present how to filter noisy data to remain

*Student Member, **Member: Dept. of Electronics and Computer Engineering, Chonnam National University

Manuscript : 2017 07 25
Revised : 2017.08.31, 2017.09.18
Confirmation of publication : 2017.09.25
Corresponding Author : Hyung-Jeong Yang, e-mail: hjyang@jnu.ac.kr

informative data for detection. In the scope of event detection, there are many traditional approaches to this task, but feature-based methods always require sophisticated feature engineering. Therefore, we proposed deep learning based approach with Convolutional Neural Network (CNN) model for filtering and Long Short Term Memory (LSTM) network model for event detection.

The rest of the paper is organized as follows: Section 2 provides a brief discussion on a background and related works. Section 3 mainly focus on proposed approach using neural network including CNN and LSTM architecture for temporal event detection. Then, we conducted the experiments with each module and performances are presented in Section 4. Finally, Section 5 is the conclusion and future works.

II. Related work

Most system for social data processing always relate to detecting a new event, and previous studies approach to find the significant increase in the number of signals at a given moment. Because there exist messages which are not relevant for a given event, we have to preprocess to filter irrelevant messages out as [1, 3] for obtaining informative data.

The social network also played as a communication and interactive platform during a disaster, this trend has been raised as the hot topic for warning disaster or event detection system using social data [1, 3, 4, 8]. The data source of [4] is Indonesia' s Tweet related to three earthquakes occurred at Sumatra coast. Text processing techniques are used to prove that viability of the use of twitters by government agencies as a complementary source when compared with traditional sources.

The authors in [3] have mined Twitter data for real-time earthquake detection and warned earlier than that of Japan Meteorological Agency. This system used Support Vector Machine (SVM) classifier to remove the irrelevant tweet. The probabilistic model is performed for temporal analysis. Poison process is the core of temporal estimation model. Similarly, [1] implemented a system for earthquake data on the Italy. This system initially considered both tweets and replied tweet from Twitter. To eliminate irrelevant information a classification based sophisticated filter is made with the features learned from experts. For temporal analysis, they created a burst detection method which observes the number of message in the time window.

Although approaches in [1, 3] are the feature-based approach, they still have some limitations in which features need to be picked by the expert. To avoid this limitation we intend to use deep learning way to learn the features of data automatically. CNN [9, 11] and LSTM [2] have become as a popular technology with advantages of deep architecture. They can learn a better representation of input data without feature engineers.

Several studies have attempted to use the discrete signal to find high frequency or "burst" feature in data stream [15]. For time series data, event detection refers to models of normal network behavior to detect new data point that significantly deviates from learning model. Long Short-Term Memory network is a recurrent neural network which is trained with Backpropagation through time and overcome the vanishing gradient problem which is a limitation in Recurrent Neural Network. We used LSTM model for event detection.

III. Proposed Method

1. Data representation

In this section, we consider how to transform a tweet into an information record. The representation of choice for the textual form is typically a numerical vector in which a sentence can be converted to a numerical matrix using Word Embeddings. Word Embeddings is natural language modeling technique or feature learning techniques. Such model takes a large corpus of text and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space [6].

We realize these vectors are as useful feature for our problems, therefore we also use Glove pre-trained in Twitter corpus [17] to generate the features for tweet data in determining informative ones (binary-classifier). Furthermore, CNN model on NLP enables to sort raw data into many classes (multi-classifier). For instance, classify input data into earthquake, flood or noise data.

Also, social data always associate to timestamp or temporal characteristic; we also transformed considered textual data into discrete signals [15] since we need to accumulate frequency of the keywords of a specific event in the regular interval to support for temporal event detection. Keywords terms are predefined to limit the scope of the tweets gathered from Twitter. A keyword may be certain hashtags (#), or terms (tokens) about event contained in tweet sentence. For example, we start with "earthquake" keys for earthquake disaster detection. Time interval refers to a specific period in which the data collected from Twitter. All tweets in the interval will be analyzed, and summation of the informative tweets is assigned as the frequency to the corresponding interval. Suppose that each time interval works as a slot and those are not overlap but be successive. In our study, the time interval is 10 seconds.

For real-time event detection, this work shows how we find abnormalities from collected data by using LSTM based prediction model.Distributing computation on the discrete signal is used as a new representation (time series data) comparing to the original form of text. The two above representations are the essential components of CNN based classification and event detection, respectively.

2. CNN based classifier

Tweets are very noisy and often contain misleading information. To enhance the influence of the informative tweets on event detection, we adopted a Convolution Neural Network (CNN) based model to determine informative information even categorizing the informative data into separated topic.

If we have a tweet message as equation (1) we obtain a feature vector matrix size of $A \in R^{S \times D}$.

$$ts = (word_1, word_2, ..., word_s) \quad (1),$$

The transformation of converting unstructured text data to readable data is look-up table layer under Word Embeddings [12].

The matrix representation A is then passed through a convolution layer, pooling layer to automatically learn features [7, 10]. Each filter function as feature extractor that is parameterized by $w \in R^{L.D}$ to perform convolution on the sub-sentence matrix. Convolution with non-linear active function generates the feature maps as equation (2) with each element determined by $h_t = f(w.x_{t:t+L-1} + b_t)$,

$$H_i = [h_1, h_1, ..., h_{T+L-1}] \quad (2),$$

where $x_{t:t+L-1}$ is the concatenation of L input vectors, b_t as bias term and f is a non-linear active function. With applying $N filters$ on the same region size for extracting leads to N different feature maps $H_1, H_2, ..., H_N$. Wide convolutional technique [7] was implemented to ensure the whole content of tweets must be filtered before passing max-pooling layer. We pooling map $H_i = [h_1, h_1, ..., h_{T+L-1}]$ to obtain

$$m = [\max_p(H_1), \max_p(H_2), ..., \max_p(H_N)] \quad (3),$$

$\max_p(H_i)$ is max-pooling function with a window of feature from each feature map H_i. The pooling operation aims to reduce the output dimensionality for cheap cost of computation and preserving the most important information of the feature map.

The features form the penultimate layer called fixed dense layer to naturally work with variable length of input. The fully connected soft-max layer is at the end to perform classification using probability distribution over output labels [10]. We trained CNN classifier models by optimizing the cross entropy using the gradient-based method along with dropout regularization. The rule Adadelta [13] is an aid of learning rate estimation in during training.

3. Event detection using LSTM

Algorithm 1: Temporal Event Detection

Input:

Size of *slide window* is fixed

Size of *time interval* $(\triangle T)$

Threshold for event candidate τ

Threshold for event detection Th

while streaming **do**

given $\widehat{i(t)}$ is prediction using LTSM model

$i(t)$ is accumulated frequency in interval $(\triangle T)$

caculate *Absolute Error* $e(t) = |i(t) - \widehat{i(t)}|$

check *anomally likelihood* of $e(t)$

if anomally likelihood $p(e(t)) < \tau$

Assign *event candidate* $C(t) = 1$

else

Assign *not event candidate* $C(t) = 0$

end

update μ, σ^2 for next check candidate

check *window score* for event detection

if sum $C(t) > Th \times size\,of\,slide\,window$

Event detected

end

end

To detect events in real-time, we proposed a temporal event detection method to deal with the streaming data. Input data of event detection is time series data underlying discrete signals which are accumulated from output of the classifier. Values are frequency of informative tweets in given interval. Our problem is similar to anomaly detection [14], that make sense for disaster event detection.

Because relative error is not suitable for our data, where frequency of events mentioned tweets could be zero, we just use absolute errors in detection algorithm 1 "Temporal Event Detection." The differences are between the real measurement and predicted value of itself. An observation is assigned as 'earthquake' candidate if the likelihood of error satisfies one threshold. Then, window scores are calculated as sum of number "earthquake" candidates in sliding windows and refers to an empirical condition. Sliding window is fixed as window sliding on time series data. This is different from the time interval since this window enables to run on time dimension with shift steps as one interval, while windows can be many intervals.

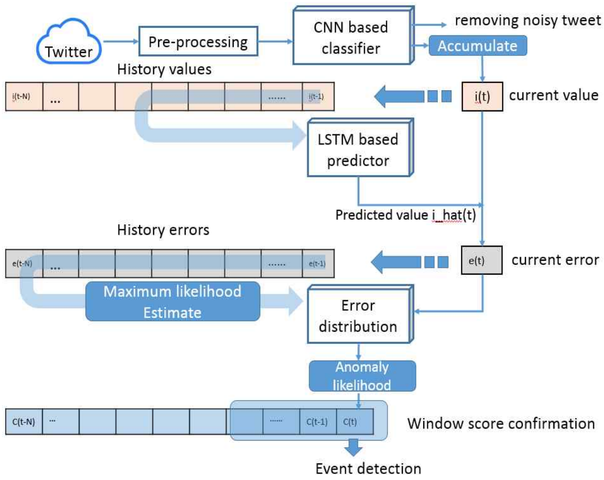


Fig. 1. Workflow for real-time event detection

The workflow of temporal event algorithm is illustrated in Figure 1, which shows that it is an infinite loop on streaming data from social platform Twitter.

The LSTM based predictor model is pre-trained to predict the current value using long-terms history data as input variables. Instead of neurons, LSTM network has memory blocks that are connected through layers. That makes itself more potential than classical neuron since the block has gates managing the block' s state and output. The LSTM based prediction model refers to observed history data to estimate incoming data point in the time-series. The prediction errors are modeled to fit parameters (mean, standard deviation) of a Gaussian error distribution using maximum likelihood estimation. Event detection will be verified via confirmation stage based on the number of event candidates in the slide window.

IV. Experimental Results

1. Data for training CNN model

Taking advantages of Convolutional Neural Network, our system is adaptive with various context, however we just considered kind of earthquake and flood datasets which manually labeled from crowdsourcing website http://crisislex.org as described in Table 1. For purpose of training, the selected dataset from earthquake and flood are labeled as 3 classes: "informative earthquake", "informative flood" and "not-informative" which contains both "not-informative flood" and "not-informative earthquake".

Table 1. Labelled training data

Informative earthquake dataset	1922	Costa Rica(2012), Italy(2012), Bohol(2013)
Informative flood dataset	1831	Philipines (2012), Colorado (2013)
Not-informative dataset	1705	From both above

2. Performance of CNN model

The labeled dataset was divided into a train set (70%), validate set (10%) and test set (20%). We conducted training in every combination of the parameter and selected the combination with the best accuracy. Dropout rate is 0.5 (50%), minibatch size is 128, the number of filter for convolution is 150 with region size=3, and strike of pooling layer is 2.

Table 2. Performance of CNN based binary-classification (noise filtering)

SVM [3]	63.64%	87.50%	73.69%
SVM [1]	88.14%	92.60%	90.33%
CNN (proposed method)	94.66%	95.33%	95.00%

From Table 2, the performance of filtering using CNN is better than [1] and [3]. Moreover, [1, 3] require a rich feature set from experts or complicated feature engineering steps, while as CNN can automatically induce their features from word embedding. This feature makes our system more adaptive as extending to another context. For example, in case of multi-classification (flood, earthquake, noise), the training phase results are reported via the confusion matrix and performance as below Table 3.

Table 3. Performance of CNN based multi-classification (sorting data)

		Predicted		
		Informative flood	Informative earthquake	Not-informative
Actual	Informative flood	359	0	7
	Informative earthquake	0	357	27
	Not-informative	17	30	294
	Pre	Recall	F-score	Acc
CNN (this)	92.51%	92.58%	92.53%	92.66%

3. Data for training LSTM model

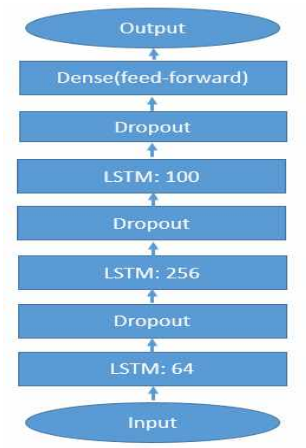


Fig. 2 :Stacked LSTM Network architecture

To train LSTM model in temporal event detection algorithm, collecting social data in time without disaster is required. In our case, earthquake detection was considered for performance so we used discrete data before or long time later specified earthquake. We should make sure that there is no disaster in collecting data for training. Stacked LSTM based prediction model was implemented with multiple recurrent LSTM layers as shown in Figure 4.

The architecture of LSTM is compose of nine layer: an input layer, six LSTM hidden layers with the number of LSTM cells {64, 256, 100}, respectively and dropout operations with 0.2 (20%), one feed−forward hidden layer and an output layer.

4. Temporal event detection

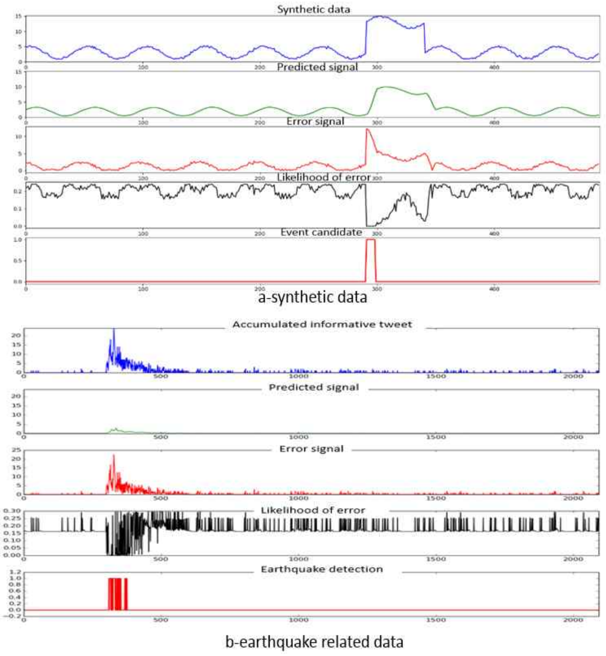


Fig. 3. Event detection on synthetic and actual data

Figure 3 illustrated flow of processing in an event detection algorithm using LSTM network on synthetic and earthquake data. From the top to down, each sub−figure presents actual time series $i(t)$ and predicted signal $\widehat{i(t)}$ that come from LSTM module, absolute error signal $e(t)$, an anomaly likelihood $p(e(t))$and earthquake candidate maker $C(t)$respectively.

We conducted the experiments in some tweet dataset we collected. For more details, these are represented as Table 4, in which the true information and time responses of our system are compared to prove the tolerant delay. Our reference comes from this site https://earthquake.usgs.gov/earthquakes. As mentioned above, our proposed approach is very potential to apply to the warning notification system [5, 16].

Table 4. Time response of event detection on the earthquake situation

	Earthquake	CNN+LSTM	Delay
Domestic earthquake (Korea)	2016/09/19 11:33:58 (UTC)	11:35:25	87sec
	2016/09/12 10:44:33 (UTC)	10:46:30	117sec
international earthquake	2016−12−27 23:20:55 (UTC)	23:22:52	117sec
	2016−12−28 08:18:00 (UTC)	08:19:16	76sec
	2016−12−28 12:38:49 (UTC)	12:39:57	68sec
	2017−06−06 18:12:28 (UTC)	18:19:14	406sec
	2017−06−08 17:01:19 (UTC)	17:02:49	90sec

V. Conclusions

In this study, we present a CNN based classifier as preprocessing raw data for event detection that utilized the capability of LSTM model on time series data. To demonstrate the potential power of proposed approach, two neural network based modules of our system also evaluated in two use cases (binary−classification, multi−classification) and 2 type of time series data (synthetic, real data) respectively. Our approach takes advantages of deep learning methods to adapt extended domain since features are learned during training. In the future, we will integrate more information such as geo−information into event detection and extend our research to many different events.

Acknowledgements

1. ITRC: “This research was supported by the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the ITRC(Information Technology Research Center) support program (IITP−2017−2016−0−00314)

supervised by the IITP (Institute for Information & communications Technology Promotion)”.

2. 지역혁신: “This research was supported by Ministry of Regional Innovation, creative business training Research Foundation of Korea, 2014 (NRF-2014H1C1A1066771)”.

REFERENCES

[1] M. Avvenuti, S. Cresci, M.N. La Polla, A. Marchetti and M. Tesconi, “Earthquake emergency management by Social Sensing,” *The Second IEEE International Workshop on Social and Community Intelligence*, pp. 587−592, 2014.

[2] S. Hochreiter, and J. Schmidhuber, “Long short−term memory,” *Neural Computation*. vol. 9, no. 8, pp. 1735-1780, 1997.

[3] T. Sakaki, M. Okazaki, and Y. Matsuo, “Earthquake shakes twitter users: real−time event detection by social sensors,” *Proceedings of the 19th international conference on World Wide Web*, pp. 851−860, 2010.

[4] A. T. Chatfield and U. Brajawidagda, “Twitter early tsunami warning system: A case study in Indonesia’s natural disaster management,” *System Science (HICSS)*, 2013 46th Hawaii International Conference on, pp. 2050−2060, 2013.

[5] T. V. Duc, N. V. Quyet, and Kyungbaek Kim, "Design and Implementation of Geo−Social information based Personalized Warning Notification system," *Smart Media Journal*, vol. 5, no. 2, pp. 42−50, 2016.

[6] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” *NIPS'13 Proceedings of the 26th International Conference on Neural Information Processing Systems*, pp. 3111-3119, 2013.

[7] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, “A convolutional neural network for modelling sentences,” *in Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics,* vol. 1, pp. 655-665, 2014.

[8] Q. Huang, G. Cervone, D. Jing, and C. Chang, “DisasterMapper: A CyberGIS framework for disaster management using social media data,” *BigSpatial'15Proceedings of the 4th International ACM SIGSPATIAL Workshop on Analytics for Big Geospatial Datan*,pp. 1−6, 2015.

[9] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436-444, 2015.

[10] Y. Kim, “Convolutional Neural Networks for Sentence Classification,” *in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pp. 1746-1751, 2014.

[11] R. Johnson and T. Zhang, “Effective use of word order for text categorization with convolutional neural networks,” arXiv:1412.1058v2, 2015.

[12] Y. Zhang and B. Wallace, “A Sensitivity Analysis of Convolutional Neural Networks for Sentence Classification,” arXiv:1510.03820v4, 2016.

[13] D. Z. Matthew, “ADADELTA: An Adaptive Learning Rate Method,” arXiv:1212.5701, 2012.

[14] R. Sebastian, “An overview of gradient descent optimization algorithms,” arXiv:1609.04747, 2016.

[15] D. T. Nguyen, and E. J. Jai, “Real−time event detection for online behavioral analysis of big

social data," *Future Generation Compter Systems*, vol. 66, pp.137-145, 2017.

[16] V. Q. Nguyen, H. J. Yang, Y. Kim, S. Kim, G. S. Lee, and M. Kwon, "Design and Implementation of Event Detection System on Hadoop." *in Proceedings 1st International. Conference Software & Smart Convergence (ICSSC 2017)*, vol. 6, no. 2, pp. 201-205, 2017.

[17] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," *in Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP)*, vol. 14, pp. 1532-1543, 2014.

Authors

Van Quan Nguyen

He received his B.E. degree in Electronics and Telecommunications from Hanoi University of Science Technology (HUST) in 2014. He is currently a M.S student at Dept. of Electronics and Computer Engineering, Chonnam National University, South Korea. His main research interests include multimedia data mining, artificial intelligence, distributed system.

Hyung-Jeong Yang

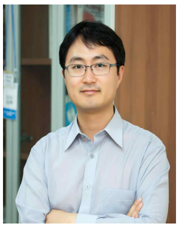
She received her B.S., M.S. and Ph. D. from Chonbuk National University, Korea. She is currently an associate professor at Dept. of Electronics and Computer Engineering, Chonnam National University, Gwangju, Korea. Her main research interests include multimedia data mining, pattern recognition, artificial intelligence, e-Learning, ande-Design.

Young Chul Kim

He received his PhD from Michigan State University, USA, the MS from the University of Detroit, USA, and BS in electronics engineering from Hanyang University, Korea. In 1993, he joined the Department of Electronics Engineering at Chonnam National University (CNU) where he is currently a professor. From 2000 to 2004, he was a director of IDEC at CNU. From 2004 to 2005, he was a Vice Dean of the College of Engineering in this university. From 2004 to 2014, he was the chief of the LG Innotek R&D center at CNU. His research interests are SoC design and smart interface and Natural Ussr Interface (NUI) and low power design.

Soo-Hyung Kim

He received his B.S. at Dept. of Computer Engineering, Seoul National University, and M.S. and Ph.D. at Dept.of Computer Science, Korea Advanced Institute of Science and Technology, Korea. He is currently a professor at Dept.of Electronics and Computer Engineering and a vice-Dean of the Engineering College, Chonnam National University, Gwangju, Korea.



Kyungbaek Kim

He received the BS, MS, and PhD degrees in electrical engineering and computer science from Korea Advanced Institute of Scienceand Technology (KAIST), Korea, in 1999,2001, and 2007, respectively. He is currently an Assistant Professor in the School of Electronics and Computer Engineering at the Chonnam National University, South Korea. Previously, he was a Postdoctoral Researcher in the Department of Computer Sciences, University of California Irvine, Irvine, CA, USA. His research interests are in the design and implementations of distributed systems including peer-to-peer systems, Grid/Clouding systems and social networking systems.