TokyoTech_NLP at SemEval-2019 Task 3: Emotion-related Symbols in Emotion Detection

Zhishen Yang
Tokyo Institute
of Technology
zhishen.yang
@nlp.c.titech.ac.jp

Sam Vijlbrief
Delft University
of Technology
s.vijlbrief
@student.tudelft.nl

Naoaki Okazaki
Tokyo Institute
of Technology
okazaki
@c.titech.ac.jp

Abstract

This paper presents our contextual emotion detection system in approaching the SemEval-2019 shared task 3: *EmoContext: Contextual Emotion Detection in Text.* This system co-operates with an emotion detection neural network method (Poria et al., 2017), emoji2vec (Eisner et al., 2016) embedding, word2vec embedding (Mikolov et al., 2013), and our proposed emoticon and emoji preprocessing method. The experimental results demonstrate the usefulness of our emoticon and emoji prepossessing method, and representations of emoticons and emoji contribute model's emotion detection.

1 Introduction

Social media and online text applications have been gaining popularity in recent years. Users post videos, pictures, and text to share their daily life as well as to communicate with others. This vast amount of multimodal data greatly facilitates various user analysis tasks such as sentiment analysis.

Emotion detection as part of sentiment analysis can be conducted with user's multimodal data such as facial expression and voice data in addition to text data. Therefore, emotion detection becomes a challenging problem when only textual data is available for extracting the contextual and sentiment features (Chatterjee et al., 2019). For example, without proper visual and voice data, "Why did you not call me last night" may be classified as sad or angry without an appropriate understanding of context.

The SemEval-2019 shared task 3: *EmoContext: Contextual Emotion Detection in Text* is the task to detect an emotion of a three-turn conversation (Chatterjee et al., 2019). We can consider this task as contextual sentiment analysis, as it requires detecting the emotion of the third-turn conversation by comprehensively understanding the contextual

Emotion class	Distribution
others	50%
happy	14%
sad	18%
angry	18%

Table 1: Emotion class distribution in the training data set.

relationship and sentiment features from both language and emotion-related symbols.

The task organizers provided a training set that consisted of 30,160 three-turn conversations. Meanwhile, as Table 1 shows, the emotion class distribution in the training data set was unbalanced: 50% of samples were from "others" class. The task organizers also mentioned about the real-life distribution of emotion class: 88% of data is classified as "others" (Chatterjee et al., 2019). We also observed that over 28% of conversations in the training data set contained emoticons or emoji; users use them along with text to express emotion in a conversation. These two observations challenge us to have a method that can learn emotion and contextual features from unbalanced training data and the emotion-related symbols (emoticons and emoji).

The rest of this paper is organized as follows: Section 2 introduces the related work to the task; Section 3 explains our method in detail; Section 4 illustrates the experiments and analyzes the experimental results; Section 5 concludes the paper and presents the future work.

2 Related Work

Natural language processing in social media as an emergent area has attracted a lot of attention (Poria et al., 2017), especially from the recent advances in applying neural network methods with

pre-trained embeddings (Eisner et al., 2016).

To achieve generalization and robustness in social media sentiment analysis, pre-trained embeddings should contain the representations of not only words from natural language but also emotion-related symbols, such as emoticons and emoji (Eisner et al., 2016). Both pre-trained embeddings GloVe (Pennington et al., 2014) and word2vec (Mikolov et al., 2013) do not contain representations for emotion-related symbols, which restricts the performance of sentiment analysis in social media. Although pre-trained emoji2vec embedding contains Unicode emoji representation, not all emotion-related symbols are included, such as emoticons.

As emotion detection is a part of sentiment analysis, and the data from the task organizers contains emoticons and emoji for emotion expressions, we can utilize a neural network method with pre-trained embedding to solve this task. We also need to address the lack of representations of emotion-related symbols.

3 Method

We formalize the SemEval-2019 shared task 3 as an emotion classification problem. Our method performs as an emotion classifier that accepts a conversation containing three-turn textual utterances, and classifies the last utterance to one of four pre-defined emotion class (happy, sad, angry, and others).

Our method focuses on learning contextual relationships and extracting emotion features from three-turn conversations. Poria et al. (2017) presented an LSTM-based contextual sentiment analysis model: contextual LSTM network that captures inter-dependency and contextual relationship among utterances in a video. Their experimental results demonstrated that contextual features of utterances significantly boost the performance of sentiment analysis; therefore, we decided to use this model as the main component in our system.

Although pre-trained embeddings such as GloVe and word2vec are easy to access, both of them do not have representations for emoticons and emoji (Eisner et al., 2016). However, pre-trained emoji2vec embedding as a supplement to pre-trained Google news word2vec (Mikolov et al., 2013) contains 1,661 emoji symbols and is ready to be augmented in downstream natural language processing tasks for social media (Eis-

ner et al., 2016). Thus, we decided to concatenate word2vec and emoji2vec in our method as word embedding.

3.1 Contextual LSTM Network

Since bi-directional contextual LSTM (bc-LSTM) performed the best in the experiments of Poria et al. (2017), we selected this variant of contextual LSTM as the main component.

bc-LSTM (Poria et al., 2017) consists of five layers: 1) embedding layer; 2) input layer; 3) LSTM layer; 4) dense layer; and 5) softmax layer. The embedding layer (shared across three utterances) converts utterances into distributed representations. The input layer is a shared bidirectional LSTM, accepting the output of each utterance from the embedding layer in a sequence. The LSTM layer is an uni-directional LSTM that uses concatenation of outputs of utterances from the input layer. The extracted contextual features from LSTM layer feed into a dense layer. Finally, the softmax layer predicts an output from the dense layer.

3.2 Emoticon and Emoji Pre-possessing

The emoticon and emoji pre-processing method removes sentiment ambiguity that emoji bring and solves the lack of representations of emoticons in pre-trained emoji2vec (Eisner et al., 2016) embedding.

Emoji Normalization Since emoji2vec does not learn context-dependent definitions of emoji (Eisner et al., 2016), a mixture and duplication of emoji within textual data in an utterance will add the complexity and ambiguity in an emotion expression.

We also noticed that appending emoji to the end of an utterance did not change its sentiment. Instead, this process splits an utterance into two parts: text part and emoji part, which guarantees the smooth emotion expression in each part.

Our emoji normalization reduces multiple instances of an emoji into one instance and append it to the end of its belonging utterance.

Emoticon to Emoji Mapping In addition to emoji, emoticons also play a vital role in expressing emotions. Thus, representations for emoticons are also important to our method.

Although emoji2vec does not contain a representation of an emoticon, an emoticon can be treated as a "surface variation" of an emoji. Thus, we can use the same emoji2vec representation of

Emoticon	Emoji
:3	😊
D;	😫
:-#	🤐
>:/	😕
:-o	😲
:-D	😃
:-#	🤐
:/	😕
=/	😕
:-o	😳
;-)	😉
>_>	😏
:-þ	😜
:-)	😊
>:¥	😕

Figure 1: Emoticon to Emoji Dictionary.

an emoji only if the emoticon is associated with the emoji. We built a dictionary to map an emoticon to its corresponding emoji (Figure 1). Containing 150 common emoticons, this dictionary can be used to replace emoticons to emoji.

3.3 System Description

Figure 2 shows an overview of our system: TokyoTech_NLP contextual emotion detection system (TNCED); our system consists of two phases: training and test phases. Both phases use the same pre-processing method. In the test phase, we use the contextual LSTM network classifier from the training phase to predict the emotion class on the test data.

4 Experiments

4.1 Data

We used the training data provided by the task organizers to train our model. For evaluation, we used the SemEval 2019 task 3 test data and micro F1 score and F1 score as metrics.

4.2 Experiment Setup

We used Keras (Chollet et al., 2015) and the code from the Github repository of bc-LSTM[1] to implement our TNCED system with the following settings: 128 LSTM dimensions; adam (Kingma and Ba, 2014) as an optimizer; 0.003 learning rate, 0.2 dropout, and 75 epochs. We used pre-trained Google word2vec (Mikolov et al., 2013) and pre-trained emoji2vec (Eisner et al., 2016) as embeddings.

4.3 Experiment Design

To evaluate the performance of *emoji normalization* and *emoticon to emoji mapping*, we used them to create following data set:

Data Set 1: For both training and test data, we first conducted *emoticon to emoji mapping*, and then performed punctuation normalization to reduce the number of repeated and frequently-used punctuation '?', '!', ',', '.' into one. We put a white space around the punctuation. Then, we applied *emoji normalization*.

Data Set 2: In addition to Data Set 1, we applied punctuation normalization to two frequently-used punctuation '_' and ':'; we only conducted *emoji normalization* in this data set. This data set was used in our submitted system for SemEval 2019 task 3.

Data Set 3: We did not apply any pre-processing in both training and test data.

We used these three data sets to train and test our system (TNCED), and calculated micro F1 score on the three emotion classes (happy, sad and angry) and F1 scores of happy, sad and angry classes.

4.4 Experimental Result

Table 2 shows micro F1 score and F1 scores of happy, sad, and angry classes obtained from the TNCED systems trained with different settings. The system with setting 1 (TNCED+Data Set 1) had the highest micro F1 score (0.7004) among the settings; it also achieved the highest F1 scores (0.746 and 0.709) in both sad and angry classes.

Compared with setting 3 (without any pre-processing), both settings 1 and 2 gained better F1 scores of three emotion classes as well as micro F1 scores, especially the setting 1 improved nearly 0.131 in F1 score of sad class and 0.071 in
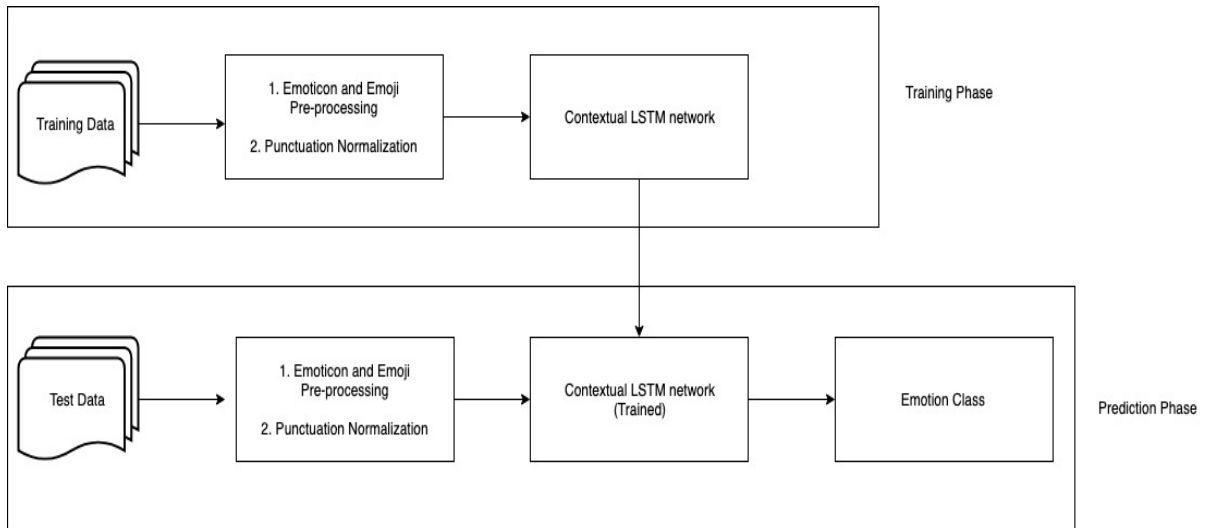
---

[1]https://github.com/SenticNet/conv-emotion/tree/master/bc-LSTM

Figure 2: TokyoTech_NLP contextual emotion detection system (TNCED).

Experiment	Micro F1	Happy F1	Sad F1	Angry F1
Setting 1 (TNCED + Data Set 1)	0.7004	0.650	0.746	0.709
Setting 2 (TNCED + Data Set 2) (Submitted)	0.6801	0.658	0.695	0.688
Setting 3 (TNCED + Data Set 3)	0.6294	0.606	0.615	0.663
emocontext_organizers	0.5868			

Table 2: Micro F1 score and F1 scores of happy, sad, and angry obtained from TokyoTech_NLP contextual emotion detection system with three different settings, and SemEval-2019 Task 3 baseline's micro F1 score.

micro F1 score. Compare to setting 2, *emoticon to emoji mapping* helped setting 1, improving 0.051 and 0.021 in F1 scores of sad and angry classes. These results indicate that our proposed emoticon and emoji pre-processing method helps the contextual LSTM network in understanding emotion expressions of emoticons and emoji.

5 Conclusion

In this paper, we described our system for SemEval-2019 shared task 3. This system consisted of the pre-processing method for emoticons and emoji, and bi-directional contextual LSTM network (Poria et al., 2017).

The proposed emoticon and emoji pre-processing method solves the lack of representations of emotion-related symbols (emoticons) in pre-trained embeddings, and removes the ambiguity in emotion expressions of emotion-related symbols. Experimental results demonstrate the usefulness of the emoticon and emoji pre-processing method by improving the representations of emoticons and emoji. It helped the neural network model to capture emotion expressions from emotion-related symbols.

A future direction of this work includes training embeddings to gather contextual definitions of all emotion-related symbols. Furthermore, we would like to explore other neural network architectures as well as retrieve more data to capture the nuance emotion in text.

6 Acknowledgement

References

Ankush Chatterjee, Kedhar Nath Narahari, Meghana Joshi, and Puneet Agrawal. 2019. Semeval-2019 task 3: Emocontext: Contextual emotion detection in text. In *Proceedings of The 13th International Workshop on Semantic Evaluation (SemEval-2019)*, Minneapolis, Minnesota.

François Chollet et al. 2015. Keras. https://keras.io.

Ben Eisner, Tim Rocktäschel, Isabelle Augenstein, Matko Bošnjak, and Sebastian Riedel. 2016. emoji2vec: Learning emoji representations from their description. *arXiv preprint arXiv:1609.08359*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Soujanya Poria, Erik Cambria, Devamanyu Hazarika, Navonil Majumder, Amir Zadeh, and Louis-Philippe Morency. 2017. Context-dependent sentiment analysis in user-generated videos. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 873–883.