# THU_NGN at SemEval-2018 Task 2: Residual CNN-LSTM Network with Attention for English Emoji Prediction

**Chuhan Wu**[1], **Fangzhao Wu**[2], **Sixing Wu**[1], **Zhigang Yuan**[1],
**Junxin Liu**[1] **and Yongfeng Huang**[1]
[1]Tsinghua National Laboratory for Information Science and Technology,
Department of Electronic Engineering, Tsinghua University Beijing 100084, China
[2]Microsoft Research Asia
{wuch15,wu-sx15,yuanzg14,ljx16,yfhuang}@mails.tsinghua.edu.cn
wufangzhao@gmail.com

## Abstract

Emojis are widely used by social media and social network users when posting their messages. It is important to study the relationships between messages and emojis. Thus, in SemEval-2018 Task 2 an interesting and challenging task is proposed, i.e., predicting which emojis are evoked by text-based tweets. We propose a residual CNN-LSTM with attention (**RCLA**) model for this task. Our model combines CNN and LSTM layers to capture both local and long-range contextual information for tweet representation. In addition, attention mechanism is used to select important components. Besides, residual connection is applied to CNN layers to facilitate the training of neural networks. We also incorporated additional features such as POS tags and sentiment features extracted from lexicons. Our model achieved 30.25% macro-averaged F-score in the first subtask (i.e., emoji prediction in English), ranking $7_{th}$ out of 48 participants.

## 1 Introduction

Emojis such as ❤️ and 😂 are widely used in social media and social network messages such as tweets. They are frequently combined with plain texts to visually complement the meaning of a message and convey various opinions and emotions (Novak et al., 2015; Barbieri et al., 2017). Social media platforms such as Twitter has accumulated a large number of emoji-incorporated messages. Analyzing the relationships between the textual message and emojis has many potential applications, such as emoji recommendation, automatic emoji-enriched message generation, and accurate sentiment analysis of social media messages (Barbieri et al., 2017).

However, the research on the relationships between textual message and emojis is limited. Existing studies on emojis mainly focus on analyzing the semantics, usage or sentiment of emojis (Aoki and Uchida, 2011; Barbieri et al., 2016a,b,c; Ljubešić and Fišer, 2016; Novak et al., 2015). For example, Barbieri et al. (2016b) explored the meaning and usage of emojis across different languages. Wijeratneet al. (2017) proposed to utilize the emoji sense definitions to improve the performance of emoji embedding model. However, these approaches cannot reveal the interplay between plain texts and emojis. In order to fill this gap, Barbieri et al. (2017) proposed a novel task to predict which emojis are evoked by text-based tweets. For example, given a tweet message *"Love my coworkers ! @user"*, a system is required to predict that emoji ❤️ is associated with this tweet.

As an extension, the SemEval-2018 Task 2[1] aims to predict emojis for English and Spanish tweets (Barbieri et al., 2018). Given a plain tweet message without emoji, systems are required to predict which emoji is evoked by this message. We proposed a residual CNN-LSTM with attention model (**RCLA**) for this task.[2] Our model combines LSTM and multi-level CNN layers to capture both long-range and local information to learn tweet representation. In addition, attention mechanism (Yang et al., 2016) is incorporated into our approach to select important components. Besides, we applied residual connection technique (He et al., 2016) to CNN layers in our model to facilitate the training of neural networks. We also incorporated additional features such as POS tags and sentiment features extracted from sentiment lexicons. Our model achieved 30.25% macro-averaged F-score on the test data of the first subtask (i.e., emoji prediction in English), and ranked $7_{th}$ out of 48 participants.

[1]https://competitions.codalab.org/competitions/17344
[2]The codes of our **RCLA** model are publicly available at https://github.com/wuch15/SemEval-2018-task2-THU_NGN



*Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018)*, pages 410–414
New Orleans, Louisiana, June 5–6, 2018. 

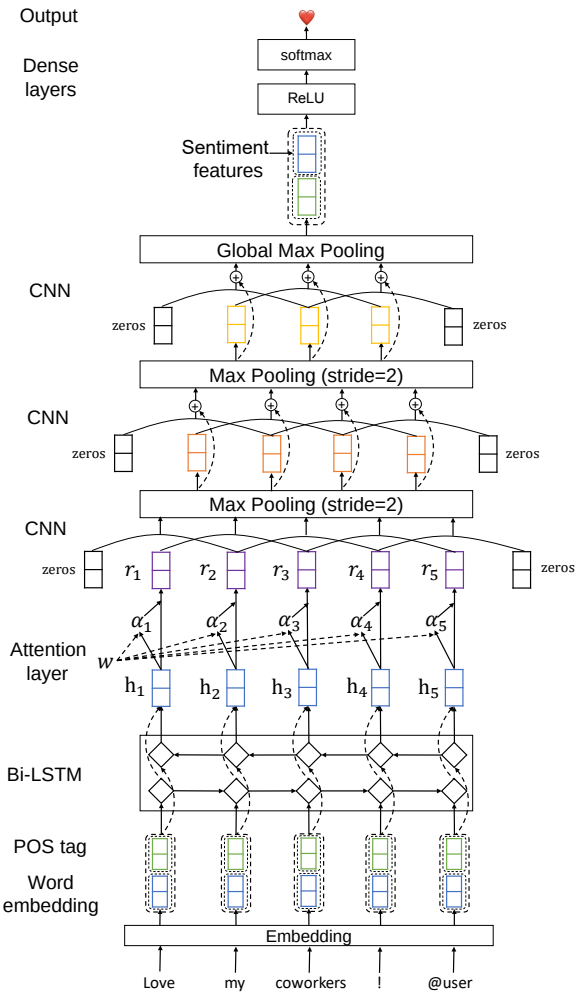


Figure 1: The architecture of our model. The dashed lines in CNN layers represent residual connections.

## 2 Residual CNN-LSTM with Attention

The framework of our residual CNN-LSTM with attention model (**RCLA**) is illustrated in Figure 1. Next, we will introduce each layer in our model from bottom to top in detail.

The first layer in our model is the embedding layer. This layer is used to convert a sentence from a sequence of words into a sequence of dense vectors. An embedding lookup table is used in this layer, whose parameters are obtained from pre-trained word embeddings and fine-tuned during training. POS tags have proven useful for many natural language processing tasks such as dimensional sentiment analysis (Wu et al., 2017). Motivated by existing studies, we also incorporate POS tags as additional features in our approach, and combining them with the word embeddings to form the final word features as the input of next layer. We use the Ark-Tweet-NLP[3] tool to obtain the POS tags of tweets.

The second layer in our model is bidirectional long short-term memory (Bi-LSTM) layer. This layer is used to capture long-range contextual information from tweets. At time step $i$, a hidden state $\mathbf{h}_i$ is generated which contains both previous and future context information. Since different words and phrases have different importance for emoji prediction, we incorporate an attention layer after the Bi-LSTM layer to help our model focus on important words and contexts. The input of the attention layer is the hidden state vector $\mathbf{h}_i$ at each time step. The attention weight $\alpha_i$ for this time step can be computed as:

$$\begin{aligned} \mathbf{m}_i &= \tanh(\mathbf{h}_i), \\ \hat{\alpha}_i &= \mathbf{w}^T \mathbf{m}_i + b, \\ \alpha_i &= \frac{\exp(\hat{\alpha}_i)}{\sum_j \exp(\hat{\alpha}_j)}, \end{aligned} \tag{1}$$

where $\mathbf{w}$ and $b$ are the parameters of the attention layer. The output of attention layer at the $i_{th}$ time step is formulated as follows:

$$\mathbf{r}_i = \alpha \mathbf{h}_i. \tag{2}$$

The third layer in our model is a 3-layer convolutional neural networks (CNN) to capture local context information. Each CNN layer has multiple kernels with different window sizes. In addition, we apply residual connections (He et al., 2016) to the CNN layers as shown in Figure 1, which have shown effectiveness in facilitating the training of deep neural networks. Max pooling is applied to the output of the last CNN layer to obtain the hidden representation of tweets.

Tweets with specific emojis such as ❤ usually convey strong sentiment information. Thus, sentiment information is helpful for emoji prediction. We incorporate sentiment features into our model to enhance its performance. These sentiment features are extracted using AffectiveTweets[4] (Mohammad and Bravo-Marquez, 2017) package in Weka[5]. Two filters are involved, i.e., TweetToLexiconFeatureVector (Bravo-Marquez et al., 2014) and TweetToSentiStrengthFeatureVector (Thelwall et al., 2012). These sentiment features are combined with the hidden tweet representations

[3]http://www.cs.cmu.edu/ ark/TweetNLP
[4]https://github.com/felipebravom/AffectiveTweets
[5]https://www.cs.waikato.ac.nz/ml/weka

generate by neural networks to form the final feature representation of tweets. Finally, a softmax layer is used to predict the emoji label.

The tweets with different emojis in the training set are very imbalanced. For example, the ratio of ❤️ is higher than 20%, while the ratio of 😜 is only 2.4%. Motivated by the cost-sensitive cross-entropy method (Santos-Rodríguez et al., 2009), the objective function of our model is defined as:

$$\mathcal{L} = -\sum_{i=1}^{N} w_{y_i} y_i \log(\hat{y}_i), \quad (3)$$

where $N$ is the number of tweets, $y_i$ is the emoji label of the $i_{th}$ tweet, $\hat{y}_i$ is the prediction score, and $w_{y_i}$ is the loss weight of emoji label $y_i$. $w_{y_i}$ is defined as $\frac{\sum_{k=1}^{C}\sqrt{N_k}}{\sqrt{N_{y_i}}}$, where $C$ is the number of emoji labels and $N_j$ is the number of tweets with emoji label $j$. Thus, the infrequent emojis have relatively larger loss weights.

## 3 Experiment

### 3.1 Dataset and Experimental Settings

The dataset[6] for this task is collected from Twitter. There are 20 emojis in total. 489,277 tweets are used for model training. The number of tweets in the trial and test sets are both 50,000. We used the pre-trained word embeddings provided by Barbieri et al. (2016b). They were trained on 20 million geo-localized tweets and their dimension is 300. These word embedding were fine-tuned during model training.

The hyperparameters in our model were selected via cross-validation on the trail set. More specifically, the dimension of Bi-LSTM hidden states is 300, the window sizes of CNN filters are 2, 3, 4 respectively. The number of CNN filters is 200 and the number of sentiment features is 45. The dimension of dense layer is 300, and the dropout rate is 0.2 for each layer. The batch size is 500, and the maximal training epoch is set to 100. We use RMSProp as the optimizer for network training. The performance is evaluated by macro-averaged F-score.

### 3.2 Performance Evaluation

The performance of our model on the test set is shown in Table 1. According to Table 1, our model can achieve good performance on predicting frequent emojis, since their training data is sufficient.

[6]https://competitions.codalab.org/competitions/17344

In addition, the performance of our approach on some infrequent emojis is also satisfactory. For example, the F-score on emoji 🎄 is high. This is probably because specific words such as "Christmas" are frequently associated with this emoji, making it relatively easy to predict.

| Emo | P | R | F1 | % |
|---|---|---|---|---|
| ❤️ | 82.9 | 79.55 | 81.19 | 21.6 |
| 😍 | 27.52 | 41.61 | 33.13 | 9.66 |
| 😂 | 33.69 | 52.43 | 41.02 | 9.07 |
| 💕 | 20.94 | 20.54 | 20.74 | 5.21 |
| 🔥 | 51.74 | 45.67 | 48.51 | 7.43 |
| 😊 | 10.38 | 11.59 | 10.95 | 3.23 |
| 😎 | 16.16 | 18.44 | 17.22 | 3.99 |
| ✨ | 35.51 | 23.54 | 28.31 | 5.5 |
| 💙 | 22.73 | 14.4 | 17.63 | 3.1 |
| 😘 | 14.93 | 15.23 | 15.08 | 2.35 |
| 📷 | 22.0 | 25.63 | 23.68 | 2.86 |
| 🇺🇸 | 64.0 | 60.03 | 61.95 | 3.9 |
| ☀️ | 64.04 | 53.36 | 58.21 | 2.53 |
| 💜 | 20.6 | 9.78 | 13.27 | 2.23 |
| 😉 | 9.99 | 6.89 | 8.16 | 2.61 |
| 💯 | 26.58 | 22.03 | 24.09 | 2.49 |
| 😁 | 8.16 | 6.24 | 7.08 | 2.31 |
| 🎄 | 66.22 | 67.12 | 66.67 | 3.09 |
| 📸 | 31.84 | 18.58 | 23.46 | 4.83 |
| 😜 | 7.1 | 3.47 | 4.66 | 2.02 |

Table 1: Precision, Recall, F-score and percentage of occurrences of each emoji in the test set.

The visualization of the confusion matrix of our model is shown in Figure 2. From this figure, we find two pairs of emojis which are difficult for our model to discriminate between them. The emoji ❤️ is often wrongly identified as ☀️. This is probably because these two emojis are often used to express similar meaning and feelings. For example, they both can be used in tweets which convey happy emotion. Another pair of emojis is 📷 and 📸. These two emojis look quite similar, and discriminating them is quite difficult.

In order to further validate the effectiveness of our model, we compare the performance of our model with several baseline methods. The methods to be compared include: 1) LSTM, using Bi-LSTM for tweet presentation; 2) CNN, 3-layer CNN without residual connections; 3) CNN-LSTM (denoted as *CL*), using the combination of LSTM and CNN; 4) Residual CNN-LSTM (denoted as *RCL*), CNN-LSTM with residual connections; 5) Residual CNN-LSTM with attention (denoted as *RCLA*). The results are shown in Table 2. According to Table 2, the combination of

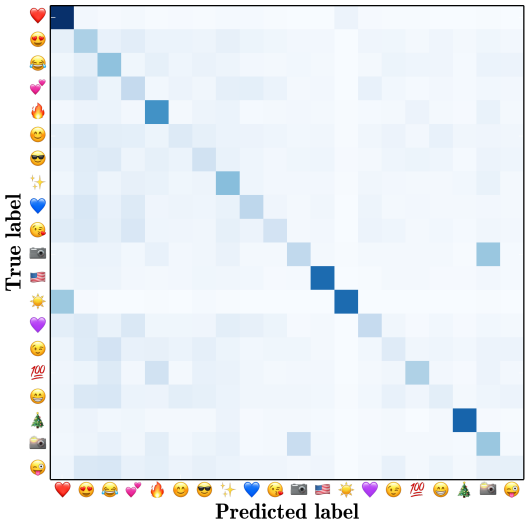


Figure 2: The confusion matrix of our model.

| Emo | LSTM | CNN | CL | RCL | RCLA |
|---|---|---|---|---|---|
| ❤️ | 79.61 | 81.24 | 81.26 | **82.79** | 81.19 |
| 😍 | 28.81 | 31.39 | 30.73 | 32.72 | **33.13** |
| 😂 | 37.34 | **41.16** | 38.35 | 38.72 | 41.02 |
| 💕 | **20.88** | 16.55 | 18.16 | 17.53 | 20.74 |
| 🔥 | 45.03 | 47.57 | 46.34 | 44.92 | **48.51** |
| 😊 | 9.71 | 9.59 | 7.99 | 10.03 | **10.95** |
| 😎 | 15.49 | **18.43** | 14.28 | 17.77 | 17.22 |
| ✨ | **28.40** | 27.43 | 27.62 | 27.72 | 28.31 |
| 💙 | **18.24** | 16.82 | 17.70 | 17.21 | 17.63 |
| 😘 | 12.13 | 14.83 | **15.87** | 13.86 | 15.08 |
| 📷 | 21.34 | 22.06 | 21.35 | 21.31 | **23.68** |
| 🇺🇸 | 58.14 | 52.76 | 59.36 | 59.58 | **61.95** |
| ☀️ | 56.37 | 57.50 | **59.03** | 58.65 | 58.21 |
| 💜 | 11.75 | 10.06 | 11.95 | **14.89** | 13.27 |
| 😉 | 9.17 | 4.00 | **10.43** | 8.72 | 8.16 |
| 💯 | 20.61 | 22.13 | 21.34 | 21.76 | **24.09** |
| 😁 | 6.50 | 6.83 | **8.73** | 6.26 | 7.08 |
| 🎄 | 63.88 | 64.73 | 64.65 | 64.50 | **66.67** |
| 📸 | 24.62 | 25.38 | 25.80 | **27.21** | 23.46 |
| 😜 | 6.31 | **6.77** | 6.11 | 5.97 | 4.66 |
| Avg. | 28.72 | 28.86 | 29.35 | 29.61 | **30.25** |

Table 2: The F-score of each emoji and the macro-F of different methods.

| Feature | Macro-F |
|---|---|
| *None* | 29.08 |
| *+POS* | 29.76 |
| *+Sentiment* | 29.55 |
| *+POS+Sentiment* | **30.25** |

Table 3: Influence of POS tags and sentiment features.

LSTM and CNN (*CL*) usually outperforms the single CNN and LSTM. It indicates that combining CNN and LSTM to capture both local and long-range context information is beneficial for tweet emoji prediction. In addition, by comparing *RCL* with *CL*, we find that the residual connections can improve the performance of *CL* model. It shows that the residual connections can facilitate the training of neural networks. Besides, the attention mechanism can also significantly improve the performance. It validates that employing attention mechanism to capture the important contexts for emoji prediction is useful.

### 3.3 Influence of Additional Features

The influence of the POS tags and sentiment features is illustrated in Table 3. The results show that both POS tags and sentiment features can help improve the performance of tweet emoji prediction. It indicates that POS tags contain useful information for predicting emojis, since important emoji clues such as hashtags, emoticons and sentiment words usually have specific POS tags. Thus, incorporating POS tag features is beneficial. Incorporating sentiment features is also useful. This is because the sentiment features we extracted from sentiment lexicons can identify both formal and information sentiment signals such as hashtags and emoticons, and these sentiment signals usually have strong associations with specific emojis. Thus, incorporating the sentiment features is also beneficial to predict emojis.

## 4 Conclusion

In this paper, we introduce our residual CNN-LSTM model with attention model (**RCLA**) for SemEval-2018 Task 2, i.e., emoji prediction for tweets. Our model combines CNN and LSTM layers to capture both local and long-range context information for tweet representation, and incorporates an attention layer to select important information. Besides, we applied residual connections to CNN layers to facilitate the training of our model. In addition, we incorporated additional features such as POS tags and sentiment features to further improve the performance. The experimental results validate the effectiveness of our model on emoji prediction for English tweets.

## Acknowledgments

The authors thank the reviewers for their insightful comments and constructive suggestions on improving this work. This work was supported in part by the National Key Research and Development Program of China under Grant

2016YFB0800402 and in part by the National Natural Science Foundation of China under Grant U1705261, Grant U1536207, Grant U1536201 and U1636113.

## References

Sho Aoki and Osamu Uchida. 2011. A method for automatically generating the emotional vectors of emoticons using weblog articles. In *Proc. 10th WSEAS Int. Conf. on Applied Computer and Applied Computational Science, Stevens Point, Wisconsin, USA*, pages 132–136.

Francesco Barbieri, Luis Espinosa Anke, and Horacio Saggion. 2016a. Revealing patterns of twitter emoji usage in barcelona and madrid. In *CCIA*, pages 239–244.

Francesco Barbieri, Miguel Ballesteros, and Horacio Saggion. 2017. Are emojis predictable? In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 105–111. Association for Computational Linguistics.

Francesco Barbieri, Jose Camacho-Collados, Francesco Ronzano, Luis Espinosa-Anke, Miguel Ballesteros, Valerio Basile, Viviana Patti, and Horacio Saggion. 2018. SemEval-2018 Task 2: Multilingual Emoji Prediction. In *Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018)*, New Orleans, LA, United States. Association for Computational Linguistics.

Francesco Barbieri, German Kruszewski, Francesco Ronzano, and Horacio Saggion. 2016b. How cosmopolitan are emojis?: Exploring emojis usage and meaning over different languages with distributional semantics. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 531–535. ACM.

Francesco Barbieri, Francesco Ronzano, and Horacio Saggion. 2016c. What does this emoji mean? a vector space skip-gram model for twitter emojis. In *LREC*.

Felipe Bravo-Marquez, Marcelo Mendoza, and Barbara Poblete. 2014. Meta-level sentiment models for big social data analysis. *Knowledge-Based Systems*, 69:86–99.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Nikola Ljubešić and Darja Fišer. 2016. A global analysis of emoji usage. In *Proceedings of the 10th Web as Corpus Workshop*, pages 82–89.

Saif Mohammad and Felipe Bravo-Marquez. 2017. Wassa-2017 shared task on emotion intensity. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 34–49, Copenhagen, Denmark. Association for Computational Linguistics.

Petra Kralj Novak, Jasmina Smailovic, Borut Sluban, and Igor Mozetic. 2015. Sentiment of emojis. *PLOS ONE*, 10(12).

Raúl Santos-Rodríguez, Darío García-García, and Jesús Cid-Sueiro. 2009. Cost-sensitive classification based on bregman divergences for medical diagnosis. In *Machine Learning and Applications, 2009. ICMLA'09. International Conference on*, pages 551–556. IEEE.

Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. 2012. Sentiment strength detection for the social web. *Journal of the Association for Information Science and Technology*, 63(1):163–173.

Sanjaya Wijeratne, Lakshika Balasuriya, Amit Sheth, and Derek Doran. 2017. A semantics-based measure of emoji similarity. In *2017 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, Leipzig, Germany. ACM, ACM.

Chuhan Wu, Fangzhao Wu, Yongfeng Huang, Sixing Wu, and Zhigang Yuan. 2017. Thu_ngn at ijcnlp-2017 task 2: Dimensional sentiment analysis for chinese phrases with deep lstm. *Proceedings of the IJCNLP 2017, Shared Tasks*, pages 47–52.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.