CLP at SemEval-2019 Task 3: Multi-Encoder in Hierarchical Attention Networks for Contextual Emotion Detection

Changjie Li
Artificial Intelligence Group
Sogou
Beijing, China, 100000
lichangjie0308@gmail.com

Yun Xing
Artificial Intelligence Lab
Lenovo Research, Lenovo
Beijing, China, 100000
xingyun44@hotmail.com

Abstract

In this paper, we describe the participation of team "CLP" in SemEval-2019 Task 3 "Contextual Emotion Detection in Text" that aims to classify emotion of user utterance in textual conversation. The submitted system is a deep learning architecture based on Hierarchical Attention Networks (HAN) and Embedding from Language Model (ELMo). The core of the architecture contains two representation layers. The first one combines the outputs of ELMo, hand-craft features and Bidirectional Long Short-Term Memory with Attention (Bi-LSTM-Attention) to represent user utterance. The second layer use a Bi-LSTM-Attention encoder to represent the conversation. Our system achieved F1 score of 0.7524 which outperformed the baseline model of the organizers by 0.1656.

1 Introduction

Emotion detection has been widely researched in psychology, sociology and computer science. Being able to recognize the emotion of text is of vital importance in the human-computer interaction (Cowie et al., 2001). However, detecting emotion in text is generally considered very challenging in absence of facial expression or voice modulation. In domain of natural language processing, emotion detection is a task of associating words, phrases or documents with emotions using psychological models (Duppada et al., 2018). Traditional rule-based approaches (Balahur et al., 2011; Chaumartin, 2007) and machine learning approaches (Alm et al., 2005; Balabantaray et al., 2012) rely on extracting word-level features to classify emotion. These methods suffer from low recall as many texts do not contain emotion words. To tackle the problem, recent deep learning approaches (Mundra et al., 2017) take the advantage of Word2Vec representation (Mikolov et al., 2013a) to extract semantic features and achieve remarkable performance. However, limited researches have been done in classifying textual conversation emotions, which is further compounded by difficulty in the context understanding.

Task 3 "Contextual Emotion Detection in Text" in SemEval-2019 aims to find better solutions for those difficulties in contextual emotion detection (Chatterjee et al., 2019). The task considers textual emotion classification on four-point scale(Happy, Sad, Angry along with an Others category). It classifies emotion of user utterance along with 2 turns of context in conversation.

This paper describes the components and results of our emotion recognition system. The proposed system is a deep learning model based on HAN, which combines multiple encoding methods including ELMo, hand-craft features and Bi-LSTM-Attention encoder. Our system yields a micro-averaged F1 score of 0.7524 on test-set of Task 3 of SemEval 2019.

2 System Description

Figure 1 provides a overall architecture of our approach, which consists of three components: (1) preprocessing, where we use a specially designed text processing method to prepare inputs for our neural network, (2) utterance encoder, where we use ELMo, hand-craft features and Bi-LSTM-Attention encoder to represent user utterance, (3) conversation encoder, where we use a Bi-LSTM-Attention layer to represent the conversation.

2.1 Preprocessing

Twitter limits that a tweet should not exceed 140 characters, which makes users use informal ways to express themselves. Emotion detection for these kinds of tweets is very challenging. To ensure effective feature extraction, we use Ekphrasis (Baziotis et al., 2017) to normalize the utterance.

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 164–168
Minneapolis, Minnesota, USA, June 6–7, 2019.

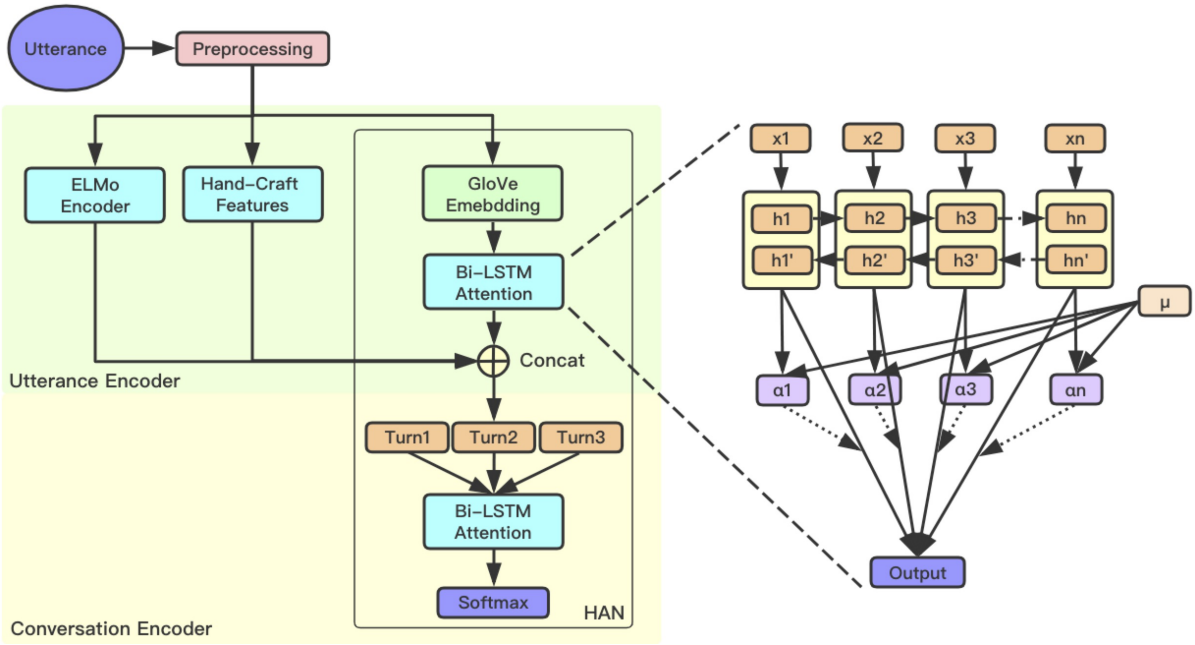

Figure 1: Overall architecture of our approach.

Ekphrasis contains a text processing pipeline that is specially designed for social network texts. The following steps are applied to utterances and lexicons in corpus:

1. **Tokenization.** We use the tokenizer in Ekphrasis to split utterance into word tokens and extract text emoticons from raw texts. The tokenizer is effective in splitting compounded words that are commonly used in Twitter. For example, the output of "#ifeelsad" will be "# i feel sad".

2. **Normalization.** We use regex regressions to detect and normalize categories, such as url/email/money/time/date. These categories are not sensitive features in the task.

3. **Annotation.** We annotate all uppercase words, repeated words and elongated words with corresponding tags, e.g. "hellooooooo" to "hello <elong>"; "yesyesyes" to "yes <repeated>"; "HELLO" to "hello <allcaps>". These informal words are vital features for prediction because they are rich in emotion.

4. **Spelling Correction.** We manually build a dictionary for out-of-vocabulary words (not in pre-trained word vectors) based on the provided datasets. 921 words are collected and corrected. The spelling correction reduces percentage of unknown words from 18% to 12%.

5. **Emoji and Emoticon Normalization.** We normalize emojis/emoticons because some of them have the same meaning. For example, "<3" and "<<33" both indicate heart, while ":(((" and ":(" both represent unhappy.

6. **Lowercase.** All characters in user utterance are converted to lowercase.

2.2 ELMo

ELMo (Peters et al., 2018) is an off-the-shelf pre-trained language model that produces deep contextualized word representation, which captures both syntax and semantic information. ELMo can be easily integrated into existing model and usually leads to performance improvement. For most state-of-the-art Natural Language Processing (NLP) tasks, pre-trained word representation is a key component (Mikolov et al., 2013b; Pennington et al., 2014). We assume that different word representations allow the model to benefit from diversified information, so we make ELMo a part of our utterance encoder. Specifically, to generate ELMo representation, we use pre-trained model provided by TensorFlow Hub [1], which outputs a mean-pooling vector of all contextualized

[1] https://tfhub.dev/google/elmo/2.

Dataset	Others	Happy	Sad	Angry	Total
original training	14948	4243	5463	5506	30160
cleaned training	14865	4231	5447	5476	30019
cleaned + augmented training	20351	14566	11240	8319	54476

Table 1: Emotion Distribution of Datasets.

word representations with 1024 dimensions in our model.

2.3 Hand-craft Features

Hand-craft features represent prior knowledge. We extract hand-craft features related to emoji and emoticon because they are frequently used as emotion indicators in Twitter and vital to textual emotion detection. We create a list that contains 300 emojis and emoticons based on this corpus. With the list, we build a Term Frequency–Inverse Document Frequency (TF-IDF) vectorizer. Finally, we convert the utterance to a 300 dimensions vector.

2.4 HAN

HAN (Yang et al., 2016) is designed to capture hierarchical structure in document. Conversation has the same hierarchical structure (words form sentence, sentences form conversation) as document, so we use HAN as the main structure of our system. Our HAN structure has two layers: utterance encoder and context encoder.

Utterance Encoder. We use pre-trained word vectors of GloVe (Pennington et al., 2014) for Twitter as our word embedding. The word embedding is put into a 1-layer Bi-LSTM followed by an attention layer (Vaswani et al., 2017). Figure 1 gives the architecture of Bi-LSTM-Attention. The Bi-LSTM summarizes utterance from both directions and incorporates the contextual information, while the attention mechanism extracts word importance. After the Bi-LSTM-Attention, we combine the output of Bi-LSTM-Attention, ELMo and hand-craft feature vector to represent user utterance.

Conversation Encoder. Given the utterance representation of each turn, we get the conversation representation in a similar way. We use another Bi-LSTM layer to summarize the contextual information in conversation, and we apply attention mechanism to capture the importance of each turn. The output vector of the conversation encoder is a high level representation of the conversation and can be used as features for classification, which is a final softmax layer that predict the emotion.

3 Experiments and Evaluation

3.1 Data Preparation

The organizers provide 30160 conversations for training, 2755 for development and 5509 for test. Before training, we remove conversations that might not be correctly labeled. Then we create more datasets by data augmentation.

Data Cleaning. We firstly train our models with five-fold cross validation. 500 false positive data points with high confidence are picked out. Among them, we manually filter and delete 141 wrong labeled conversations.

Data Augmentation. Data augmentation (DA) is frequently used in Computer Vision (Fawzi et al., 2016). However, this method is less powerful in NLP because NLP data is discrete. Even small perturbations may change the meaning of a whole sentence. In this task, we assume that the positions of emojis and emoticons do not influence the emotion of sentences, so DA can be considered reliable. Our DA includes two steps, 1) all emojis and emoticons in an utterance are extracted by using Ekphrasis, 2) we relocate the emojis and emoticons to the start and the end of the utterance, thus we create 2 additional utterances (not applied for utterances that contain emojis/emoticons only, or utterances begin or end with emojis/emoticons). In total, we get at most 3 utterances for each turn, which means 27 conversations for three turns.

Table 1 describes the emotion distribution of original, data cleaned, data cleaned and augmented training datasets. The proportion of each class in original training dataset is around 4:1:1:1 (Others:Happy:Sad:Angry) and it remains the same after data cleaning. However, DA changes the proportion to around 5:4:3:2 because Twitter users are more likely to use emojis/emoticons when they post happy, sad and angry tweets. In total, 24457 additional data points are created and the distribution of Angry, Sad and Happy classes is improved.

Model	$\mathbf{F1}_{Angry}$	$\mathbf{F1}_{Happy}$	$\mathbf{F1}_{Sad}$	$\mathbf{F1}_{Micro}$
Baseline of Organizers	0.5945	0.5461	0.6149	0.5868
HAN	0.6585	0.6716	0.7667	0.6935
HAN+ELMo	0.6922	0.6973	0.7462	0.7102
HAN+ELMo+HCF	0.7062	0.6997	0.7575	0.7199
HAN+ELMo+HCF+Preprocessing	0.7552	0.6935	0.7959	0.7459
HAN+ELMo+HCF+Preprocessing+DA	**0.7607**	**0.7013**	**0.7961**	**0.7524**

Table 2: Class-wise and micro-averaged F1 scores for models. Our best result comes from HAN + ELMo + Hand-craft Feature (HCF) + Preprocessing + DA.

3.2 Hyper-parameters

We minimize the cross-entropy loss function by using back-propagation with Adam (Kingma and Ba, 2015) and mini-batches of size 64. In order to optimize our results, we introduce class weights in loss function to reduce the impact of the unbalanced training set. The value of class weights is set based on the distribution of classes. The configuration of hyper-parameters includes as follows: the word embedding size is 200; the dimension of hidden layer size in LSTM is 200; the max length of the utterance in each turn is set 25, as nearly 99% of the utterances have less than 25 word tokens; the dropout rate is 0.2 to prevent over-fitting; the learning rate is 10e-3 and the learning rate decay is 10e-5 for each update.

3.3 Results and Discussion

Table 2 presents the results of different approaches. The organizers provide a baseline model with 0.5878 F1 score. Our best model achieves F1 score of 0.7524 which outperforms the baseline by 0.1656. We present 5 models to show how different components (Preprocessing, ELMo, Hand-craft features and DA) affect the performance. The results indicate that HAN is well performed in this task, which alone increases F1 score from 0.5878 to 0.6935. With ELMo encoder and hand-craft features, the performance improves by 0.0264 and continues to rise to 0.7449 if we apply preprocessing to the utterances. DA improves the F1 score of Angry, Sad and Happy, suggesting that more data points of these three classes are beneficial for the task.

Attention weight in HAN reflects how utterances contribute to emotion classification. Therefore, we calculate the average attention weight of three turns in conversation of test-set. Figure 2 shows that the third turn contributes the most to emotion detection, complying with the objective

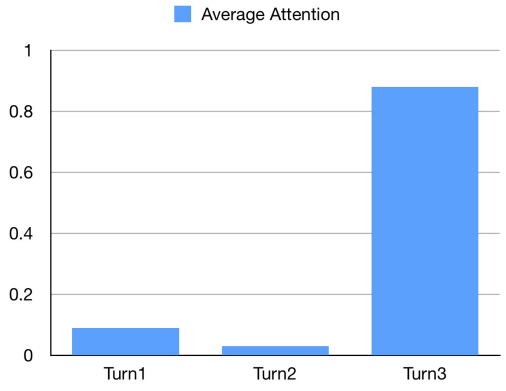

Figure 2: Attention weight of three turns.

of the task which is to predict third turn emotion. We also find that the weight of first turn is higher than the second, which can be explained by the fact that the first turn and the third turn come from the same user.

4 Conclusion

In this paper we describe our solution to SemEval 2019 Task 3. To classify contextual emotion in a conversation, we propose a HAN based deep learning model that combines multiple encoding methods including ELMo, hand-craft features and Bi-LSTM-Attention encoder. We also build a preprocessing method to improve inputs quality and we apply data augmentation to create more data points. With all these components, our system achieves micro-averaged F1 score of 0.7524 and ranks 17th out of 165 teams on Task 3 leaderboard.

References

Cecilia Alm, Dan Roth, and Richard Sproat. 2005. Emotions from text: Machine learning for text-based emotion prediction.

R C Balabantaray, Iiit Bhubaneswar, Mudasir Moham-

mad, and Nibha Sharma. 2012. N.: Multi-class twitter emotion classification: A new approach. *International Journal of Applied Information Systems*, pages 48–53.

Alexandra Balahur, Jesús M. Hermida, and Andrés Montoyo. 2011. Detecting implicit expressions of sentiment in text based on commonsense knowledge. In *Proceedings of the 2Nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, WASSA '11, pages 53–60, Stroudsburg, PA, USA. Association for Computational Linguistics.

Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017. Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754, Vancouver, Canada. Association for Computational Linguistics.

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-Régis Chaumartin. 2007. Upar7: A knowledge-based system for headline sentiment tagging. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 422–425, Prague, Czech Republic. Association for Computational Linguistics.

Roddy Cowie, Ellen Douglas-Cowie, Nicolas Tsapatsoulis, George Votsis, Stefanos Kollias, Winfried Fellenz, and J.G. Taylor. 2001. Emotion recognition in human-computer interaction. *Signal Processing Magazine, IEEE*, 18:32 – 80.

Venkatesh Duppada, Royal Jain, and Sushant Hiray. 2018. Seernet at semeval-2018 task 1: Domain adaptation for affect in tweets. pages 18–23.

A. Fawzi, H. Samulowitz, D. Turaga, and P. Frossard. 2016. Adaptive data augmentation for image classification. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3688–3692.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, USA. Curran Associates Inc.

Shreshtha Mundra, Anirban Sen, Manjira Sinha, Sandya Mannarswamy, Sandipan Dandapat, and Shourya Roy. 2017. Fine-grained emotion detection in contact center chat utterances. pages 337–349.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

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, San Diego, California. Association for Computational Linguistics.