ANA at SemEval-2019 Task 3: Contextual Emotion detection in Conversations through hierarchical LSTMs and BERT

Chenyang Huang, Amine Trabelsi, Osmar R. Zaïane
Department of Computing Science, University of Alberta
{chuang8,atrabels,zaiane}@ualberta.ca

Abstract

This paper describes the system submitted by ANA Team for the SemEval-2019 Task 3: EmoContext. We propose a novel Hierarchical LSTMs for Contextual Emotion Detection (HRLCE) model. It classifies the emotion of an utterance given its conversational context. The results show that, in this task, our HRCLE outperforms the most recent state-of-the-art text classification framework: BERT. We combine the results generated by BERT and HRCLE to achieve an overall score of 0.7709 which ranked 5th on the final leader board of the competition among 165 Teams.

1 Introduction

Social media has been a fertile environment for the expression of opinion and emotions via text. The manifestation of this expression differs from traditional or conventional opinion communication in text (e.g., essays). It is usually short (e.g. Twitter), containing new forms of constructs, including emojis, hashtags or slang words, etc. This constitutes a new challenge for the NLP community. Most of the studies in the literature focused on the detection of sentiments (i.e. positive, negative or neutral) (Mohammad and Turney, 2013; Kiritchenko et al., 2014).

Recently, emotion classification from social media text started receiving more attention (Mohammad et al., 2018; Yaddolahi et al., 2017). Emotions have been extensively studied in psychology (Ekman, 1992; Plutchik, 2001). Their automatic detection may reveal important information in social online environments, like online customer service. In such cases, a user is conversing with an automatic chatbot. Empowering the chatbot with the ability to detect the user's emotion is a step forward towards the construction of an emotionally intelligence agent. Giving the detected emotion, an emotionally intelligent agent would generate an empathetic response. Although its potential convenience, detecting emotion in textual conversation has seen limited attention so far. One of the main challenges is that one users utterance may be insufficient to recognize the emotion (Huang et al., 2018). The need to consider the context of the conversion is essential in this case, even for human, specifically given the lack of voice modulation and facial expressions. The usage of figurative language, like sarcasm, and the class size's imbalance adds up to this problematic (Chatterjee et al., 2019a).

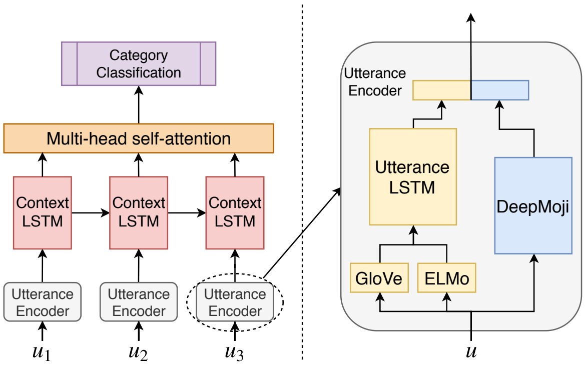

Figure 1: An illustration of the HRLCE model

In this paper, we describe our model, which was proposed for the SemEval 2019-Task 3 competition: Contextual Emotion Detection in Text (EmoContext). The competition consists in classifying the emotion of an utterance given its conversational context. More formally, given a textual user utterance along with 2 turns of context in a conversation, the task is to classify the emotion of user utterance as Happy, Sad, Angry or Others (Chatterjee et al., 2019b). The conversations are extracted from Twitter.

We propose an ensemble approach composed of two deep learning models, the *Hierarchical LSTMs for Contextual Emotion Detection* (HRLCE) model and the BERT model (Devlin et al., 2018). The BERT is a pre-trained language model that has shown great success in many NLP classification tasks. Our main contribution consists in devising the HRLCE model.

Figure 1 illustrates the main components of the HRLCE model. We examine a transfer learning approach with several pre-trained models in order to encode each user utterance semantically and emotionally at the word-level. The proposed model uses Hierarchical LSTMs (Sordoni et al., 2015) followed by a multi-head self attention mechanism (Vaswani et al., 2017) for a contextual encoding at the utterances level.

The model evaluation on the competition's test set resulted in a 0.7709 harmonic mean of the macro-F1 scores across the categories *Happy*, *Angry*, and *Sad*. This result ranked 5th in the final leader board of the competition among 142 teams with a score above the organizers' baseline.

2 Overview

2.1 Embeddings for semantics and emotion

We use different kinds of embeddings that have been deemed effective in the literature in capturing not only the syntactic or semantic information of the words, but also their emotional content. We breifly describe them in this section.

GloVe, (Pennington et al., 2014) is a widely used pre-trained vector representation that captures fine-grained syntactic and semantic regularities. It has shown great success in word similarity tasks and Named Entity Recognition benchmarks.

ELMo, or Embeddings from Language Models, (Peters et al., 2018) are deep contextualized word representations. These representations enclose a polysemy encoding, i.e., they capture the variation in the meaning of a word depending on its context. The representations are learned functions of the input, pre-trained with deep bi-directional LSTM model. It has been shown to work well in practice on multiple language understanding tasks like question answering, entailment and sentiment analysis. In this work, our objective is to detect emotion accurately giving the context. Hence, employing such contextual embedding can be crucial.

DeepMoji (Felbo et al., 2017) is a pre-trained model containing rich representations of emotional content. It has been pre-trained on the task of predicting the emoji contained in the text using Bi-directional LSTM layers combined with an attention layer. A distant supervision approach was deployed to collect a massive (1.2 billion Tweets) dataset with diverse set of noisy emoji labels on which DeepMoji is pre-trained. This led to state-of-the art performance when fine-tuning DeepMoji on a range of target tasks related to sentiment, emotion and sarcasm.

2.2 Hierarchical RNN for context

One of the building component of our proposed model (see Figure 1) is the Hierarchical or Context recurrent encoder-decoder (HRED) (Sordoni et al., 2015). HRED architecture is used for encoding dialogue context in the task of multi-turn dialogue generation task (Serban et al., 2016). It has been proven to be effective in capturing the context information of dialogue exchanges. It contains two types of recurrent neural net (RNN) units: *encoder* RNN which maps each utterance to an utterance vector; *context* RNN which further processes the utterance vectors. HRED is expected to produce a better representation of the context in dialogues because the *context* RNN allows the model to represent the information exchanges between the two speakers.

2.3 BERT

BERT, the Bidirectional Encoder Representations for Transformers, (Devlin et al., 2018) is a pre-trained model producing context representations that can be very convenient and effective. BERT representations can be fine-tuned to many downstream NLP tasks by adding just one additional output layer for the target task, eliminating the need for engineering a specific architecture for a task. Using this setting, it has advanced the state-of-the-art performances in 11 NLP tasks. Using BERT in this work has slightly improved the final result, when we combine it with our HRLCE in an ensemble setting.

2.4 Importance Weighting

Importance Weighting (Sugiyama and Kawanabe, 2012) is used when label distributions between the training and test sets are generally different, which is the case of the competition datasets (Table 2). It corresponds to weighting the samples according to their importance when calculating the loss.

A supervised deep learning model can be regarded as a parameterized function $f(\boldsymbol{x}; \boldsymbol{\theta})$. The backpropagation learning algorithm through a differentiable loss is a method of *empirical risk minimization* (ERM). Denote $(\boldsymbol{x}_i^{tr}, y_i^{tr})$, $i \in [1 \ldots n_{tr}]$

are pairs of training samples, testing samples are $(\boldsymbol{x}_i^{te}, y_i^{te})$, $i \in [1 \dots n_{te}]$.

The ratio of $P_{te}(\boldsymbol{x}_i^{tr})/P_{tr}(\boldsymbol{x}_i^{tr})$ is referred as the *importance* of a traning sample $\boldsymbol{x}_i^{tr}$. When the probability of an input $\boldsymbol{x}_i^{tr}$ in training and testing sets are generally different: $P_{tr}(\boldsymbol{x}_i^{tr}) \neq P_{te}(\boldsymbol{x}_i^{tr})$, the training of the model $f_{\boldsymbol{\theta}}$ is then called under *covariate shift*. In such situation, the parameter $\hat{\boldsymbol{\theta}}$ should be estimated through *importance-weighted ERM*:

$$\arg\min_{\boldsymbol{\theta}} \left[\frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} \frac{P_{te}(\boldsymbol{x}_i^{tr})}{P_{tr}(\boldsymbol{x}_i^{tr})} \text{loss}(y_i^{tr}, f(\boldsymbol{x}_i^{tr}; \boldsymbol{\theta})) \right]. \tag{1}$$

3 Models

Denote the input $\boldsymbol{x} = [u_1, u_2, u_3]$, where u_i is the ith penultimate utterance in the dialogue. y is the emotion expressed in u_3 while giving u_1 and u_2 as context.

To justify the effectiveness of the modules in HRLCE, we propose two baseline models: SA-LSTM (SL) and SA-LSTM-DeepMoji (SLD). The SL model is part of the SLD model. The latter one corresponds to the utterance encoder of our HRLCE. Therefore, we illustrate the models consecutively in Sections 3.1, 3.2, and 3.3.

3.1 SA-LSTM (SL)

Let $\boldsymbol{x}$ be the concatenation of u_1 ,u_2, and u_3. Hereby, $\boldsymbol{x} = [x_1, x_2, \cdots, x_n]$, where x_i is the ith word in the combined sequence. Denote the pre-trained GloVe model as G. As GloVe model can be directly used by looking up the word x_i, we can use $G(x_i)$ to represent its output. On the contrary, ELMo embedding is not just dependent on the word x_i, but on all the words of the input sequence. When taking as input the entire sequence $\boldsymbol{x}$, n vectors can be extracted from the pre-trained ElMo model. Denote the vectors as $\boldsymbol{E} = [E_1, E_2, \cdots, E_n]$. E_i contains both contextual and semantic information of word x_i. We use a two-layer bidirectional LSTM as the encoder of the sequence $\boldsymbol{x}$. For simplicity, we denote it as $LSTM^e$. In order to better represent the information of x_i, we use the concatenation of $G(x_i)$ and E_i as the feature embedding of x_i. Therefore, we have the following recurrent progress:

$$h_t^e = LSTM^e([G(x_t); E_t], h_{t-1}^e). \tag{2}$$

h_t^e is the hidden state of encoder LSTM at time step t, and $h_0^e = \mathbf{0}$. Let $\boldsymbol{h}_{\boldsymbol{x}}^e = [h_1^e, h_2^e, \cdots, h_n^e]$ be the n hidden states of encoder given the input $\boldsymbol{x}$. Self-attention mechanism has been proven to be effective in helping RNN dealing with dependency problems (Lin et al., 2017). We use the multi-head version of the self-attention (Vaswani et al., 2017) and set the number of channels for each head as 1. Denote the self-attention module as SA, it takes as input all the hidden states of the LSTM and summarizes them into a single vector. This process is represented as $h_{\boldsymbol{x}}^{sa} = SA(\boldsymbol{h}_{\boldsymbol{x}}^e)$. To predict the model, we append a fully connected (FC) layer to project $h_{\boldsymbol{x}}^{sa}$ on to the space of emotions. Denote the FC layer as $output$. Let $o_{\boldsymbol{x}}^{SL} = output(h_{\boldsymbol{x}}^{sa})$, then the estimated label of $\boldsymbol{x}$ is the $\arg\max_i(o_{\boldsymbol{x}}^{SL})$, where i is ith value in the vector $o_{\boldsymbol{x}}^{SL}$.

	F1	Happy	Angry	Sad	Harm. Mean
SL	Dev	0.6430	0.7530	0.7180	0.7016
	Test	0.6400	0.7190	0.7300	0.6939
SLD	Dev	0.6470	0.7610	0.7360	0.7112
	Test	0.6350	0.7180	0.7360	0.6934
HRLCE	Dev	0.7460	0.7590	0.8100	**0.7706**
	Test	0.7220	0.766	0.8180	**0.7666**
BERT	Dev	0.7138	0.7736	0.8106	0.7638
	Test	0.7151	0.7654	0.8157	0.7631

Table 1: Macro-F1 scores and its harmonic means of the four models

3.2 SA-LSTM-DeepMoji (SLD)

SLD is the combination of SL and DeepMoji. An SLD model without the output layer and the self-attention layer is in fact the utterance encoder of the proposed HRLCE, which is illustrated in the right side of Figure 1. Denote the DeepMoji model as D, when taking as input $\boldsymbol{x}$, the output is represented as $h_{\boldsymbol{x}}^d = D(\boldsymbol{x})$. We concatenate $h_{\boldsymbol{x}}^d$ and $h_{\boldsymbol{x}}^{sa}$ as the feature representation of sequence of $\boldsymbol{x}$. Same as SL, an FC layer is added in order to predict the label: $o_{\boldsymbol{x}}^{SLD} = output([h_{\boldsymbol{x}}^{sa}; h_{\boldsymbol{x}}^d])$.

3.3 HRLCE

Unlike SL and SLD, the input of HRLCE is not the concatenation of u_1, u_2, and u_3.

Following the annotation in Section 3.1 and 3.2, an utterance u_i is firstly encoded as $h_{u_i}^e$ and $h_{u_i}^d$. We use another two layer bidirectional LSTM as the context RNN, denoted as $LSTM^c$. Its hidden states are iterated through:

$$h_t^c = LSTM^c([h_{u_t}^e; h_{u_t}^d], h_{t-1}^c), \tag{3}$$

where $h_0^c = \mathbf{0}$. The three hidden states $\boldsymbol{h^c} = [h_1^c, h_2^c, h_3^c]$, are fed as the input to a self-attention layer. The resulting vector $SA(\boldsymbol{h}^{\boldsymbol{c}})$ is also projected to the label space by an FC layer.

3.4 BERT

BERT (Section 2.3) can take as input either a single sentence or a pair of sentences. A "sentence" here corresponds to any arbitrary span of contiguous words. In this work, in order to fine-tune BERT, we concatenate utterances u_1 and u_2 to constitute the first sentence of the pair. u_3 is the second sentence of the pair. The reason behind such setting is that we assume that the target emotion y is directly related to u_3, while u_1 and u_2 are providing additional context information. This forces the model to consider u_3 differently.

4 Experiment

4.1 Data preprocessing

From the training data we notice that emojis are playing an important role in expressing emotions. We first use *ekphrasis* package (Baziotis et al., 2017) to clean up the utterances. *ekphrasis* corrects misspellings, handles textual emotions (e.g. ':)))'), and normalizes tokens (hashtags, numbers, user mentions etc.). In order to keep the semantic meanings of the emojis, we use the *emojis* package[1] to first convert them into their textual aliases and then replace the ":" and "_" with spaces.

4.2 Environment and hyper-parameters

We use PyTorch 1.0 for the deep learning framework, and our code in Python 3.6 can be accessed in GitHub[2]. For fair comparisons, we use the same parameter settings for the common modules that are shared by the SL, SLD, and HRLCE. The dimension of *encoder* LSTM is set to 1500 per direction; the dimension of *context* LSTM is set to 800 per direction. We use Adam optimizer with initial learning rate as 5e-4 and a decay ratio of 0.2 after each epoch. The parameters of DeepMoji are set to trainable. We use *BERT-Large* pre-trained model which contains 24 layers.

	happy	angry	sad	others	size
Train	14.07%	18.26%	18.11%	49.56%	30160
Dev	5.15%	5.44%	4.54%	84.86%	2755
Test	4.28%	5.57%	4.45%	85.70%	5509

Table 2: Label distribution of train, dev, and test set

According to the description in (CodaLab, 2019), the label distribution for *dev* and *test* sets are roughly 4% for each of the emotions. However, from the *dev* set (Table 2) we know that the proportions of each of the emotion categories are better described as %5 each, thereby we use %5 as the empirical estimation of distribution $P_{te}(\boldsymbol{x}_i^{tr})$. We did not use the exact proportion of *dev* set as the estimation to prevent the overfitting towards *dev* set. The sample distribution of the *train* set is used as $P_{tr}(\boldsymbol{x}_i^{tr})$. We use *Cross Entropy* loss for all the aforementioned models, and the loss of the training samples are weighted according to Eq. 1.

4.3 Results and analysis

We run 9-fold cross validation on the training set. Each iteration, 1 fold is used to prevent the models from overfitting while the remaining folds are used for training. Therefore, every model is trained 9 times to ensure stability. The inferences over *dev* and *test* sets are performed on each iteration. We use the majority voting strategy to merge the results from the 9 iterations. The results are shown in Table 1. It shows that the proposed HRLCE model performs the best. The performance of SLD and SL are very close to each other, on the *dev* set, SLD performs better than SL but they have almost the same overall scores on the *test* set. The Macro-F1 scores of each emotion category are very different from each other: the classification accuracy for emotion *Sad* is the highest in most of the cases, while the emotion *Happy* is the least accurately classified by all the models. We also noticed that the performance on the *dev* set is generally slightly better than that on the *test* set.

5 Conclusions

Considering the competitive results generated by BERT, we combined BERT and our proposed model in an ensemble and obtained 0.7709 on the final test leaderboard. From a confusion matrix of our final submission, we notice that there are barely miss-classifications among the three categories (*Angry, Sad*, and *Happy*). For example, the emotion Sad is rarely miss-classified as "Happy" or "Angry". Most of the errors correspond to classifying the emotional utterances in the *Others* category. We think, as future improvement, the models need to first focus on the binary classification "Others" versus "Not-Others", then the "Not-Others" are classified in their respective emotion.

[1] https://pypi.org/project/emoji/

[2] https://github.com/chenyangh/SemEval2019Task3

References

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, Umang Gupta, Manoj Kumar Chinnakotla, Radhakrishnan Srikanth, Michel Galley, and Puneet Agrawal. 2019a. Understanding emotions in text using deep learning and big data. *Computers in Human Behavior*, 93:309–317.

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

CodaLab. 2019. Semeval19 task 3: Emocontext. https://competitions.codalab.org/competitions/19790#learn_the_details-data-set-format.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Paul Ekman. 1992. An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200.

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Chenyang Huang, Osmar R. Zaiane, Amine Trabelsi, and Nouha Dziri. 2018. Automatic dialogue generation with expressed emotions. In *16th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, New Orleans, USA.

Svetlana Kiritchenko, Xiaodan Zhu, and Saif M Mohammad. 2014. Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*, 50:723–762.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. Semeval-2018 task 1: Affect in tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 1–17. Association for Computational Linguistics.

Saif M Mohammad and Peter D Turney. 2013. Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 29(3):436–465.

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. *arXiv preprint arXiv:1802.05365*.

Robert Plutchik. 2001. The nature of emotions: Human emotions have deep evolutionary roots, a fact that may explain their complexity and provide tools for clinical practice. *American Scientist*, 89(4):344–350.

Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 553–562. ACM.

Masashi Sugiyama and Motoaki Kawanabe. 2012. *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT press.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Ali Yaddolahi, Ameneh Gholipour Shahraki, and Osmar R. Zaiane. 2017. Current state of text sentiment analysis from opinion to emotion mining. *ACM Computing Surveys*, 50(2):25:1–25:33.