Emotion Classification by Jointly Learning to Lexiconize and Classify

Deyu Zhou [*]
South China University of Technology
chawdoe@163.com

Shuangzhi Wu
Tencent / Beijing, China
frostwu@tencent.com

Qing Wang [*]
University of Illinois at Urbana Champaign
qwang55@illinois.edu

Jun Xie
Tencent / Beijing, China
stiffxie@tencent.com

Zhaopeng Tu
Tencent AI Lab
tuzhaopeng@gmail.com

Mu Li
Tencent / Beijing, China
limugx@qq.com

Abstract

Emotion lexicons have been shown effective for emotion classification (Baziotis et al., 2018). Previous studies handle emotion lexicon construction and emotion classification separately. In this paper, we propose an emotional network (EmNet) to jointly learn sentence emotions and construct emotion lexicons which are dynamically adapted to a given context. The dynamic emotion lexicons are useful for handling words with multiple emotions based on different context, which can effectively improve the classification accuracy. We validate the approach on two representative architectures – LSTM and BERT, demonstrating its superiority on identifying emotions in English tweets. Our model outperforms several approaches proposed in previous studies and achieves new state-of-the-art on the benchmark Twitter dataset.

1 Introduction

Tweet	Emotion
This is a **joke** really how long will he keep diving and ducking.	*disgust*
That's the **joke**. I know it's incense.	joy

Table 1: Example sentences and their emotions.

The last several years have seen a land rush in research on identification of emotions in short text such as Twitter or product reviews due to its greatly commercial value. For example, the emotions (e.g., anger or joy) expressed in product reviews can be a major factor in deciding the marketing strategy for a company (Meisheri and Dey, 2018). The SOTA approaches to this task (Baziotis et al., 2018; Meisheri and Dey, 2018) generally employ pre-defined emotion lexicons, which have two major limitations:

1. Most established emotion lexicons were created for a general domain, and suffer from limited coverage and inaccuracies when applied to the highly informal short text.
2. The pre-defined lexicons suffer from the ambiguity problem: the emotion of a word is highly influenced by the context. Table 1 shows an example. The word "joke" carries different emotions according to different context.

In this work, we tackle these challenges by jointly learning to construct the emotion lexicons and classify the emotions of short texts. Specifically, we propose a novel emotional network (EmNet), which consists of three main components:

1. *Sentence encoder* encodes the input sentence into semantic hidden states, which can be implemented as either LSTM or BERT.

[*] Work done while the author was an intern at Tencent.

Proceedings of the 28th International Conference on Computational Linguistics, pages 3235–3245
Barcelona, Spain (Online), December 8-13, 2020

2. *Emotion generator* leverages both the semantic hidden states and word embeddings to construct word emotions, which dynamically adapt to the sentence context.
3. *Emotion classifier* classifies the sentence emotions based on both the encoded sentence representations and generated word emotions.

With the newly introduced emotion generator, our EmNet can alleviate the domain mismatch and emotion ambiguity problems of using external lexicons. For example, the contextual words “how long”, “keeps diving and ducking” can help disambiguate the emotion of the word “joke”, thus improve the accuracy of emotion classification.

We validate the proposed approach on the Twitter dataset of SemEval-2018 task (Mohammad et al., 2018) on top of both the LSTM (Hochreiter and Schmidhuber, 1997) and BERT (Devlin et al., 2019) architectures. Our approach consistently outperforms the baseline models across model architectures, demonstrating the effectiveness and universality of the proposed approach. In addition, our model also outperforms the SOTA method of leveraging external emotion lexicon. Further analyses reveal that the proposed EmNet can learn reasonable emotion lexicons as expected, which avoids the mismatch problem of using external resource.

Contributions. The main contributions of this paper are listed as follows:

- We propose a novel emotional network for multi-label emotion classification which jointly learns emotion lexicons and conducts classification. We apply EmNet to both LSTM and BERT architectures to verify its effectiveness.
- The proposed model can generate context-aware word emotions, which are effective to improve the classification accuracy. We also give a qualitative analysis to help to understand how EmNet works.
- Experimental results show that our method outperforms the baselines on the public benchmark. Further analysis demonstrates the effectiveness of the proposed methods on correlation representation.

2 Emotional Network

2.1 Framework

Problem Formalization Given an input sentence, the goal of emotion analysis is to identify single or multiple emotions expressed by it. Formally, we define $S = w_1, w_2, ..., w_i, ..., w_n$ as the input sentence with n words. w_i is the i-th word and the corresponding word embedding $\mathbf{E}_{w_i}$ is retrieved from a lookup table $\mathbf{E} \in R^{d\times|V|}$. Moreover, let Φ be a set of pre-defined emotion labels with $|\Phi| = K$. Thus, for each S, the task is to predict whether it contains one or more emotion labels in Φ. We denote the output as $\mathbf{l} \in \{0, 1\}^K$, a vector with maximum dimension K, where the element $l_k \in \{0, 1\}$ refers to whether or not S contains the k-th emotion. The training data D contains a set of sentences together with their label vectors $D = \{S^{(i)}, \mathbf{l}^{(i)}\}$.

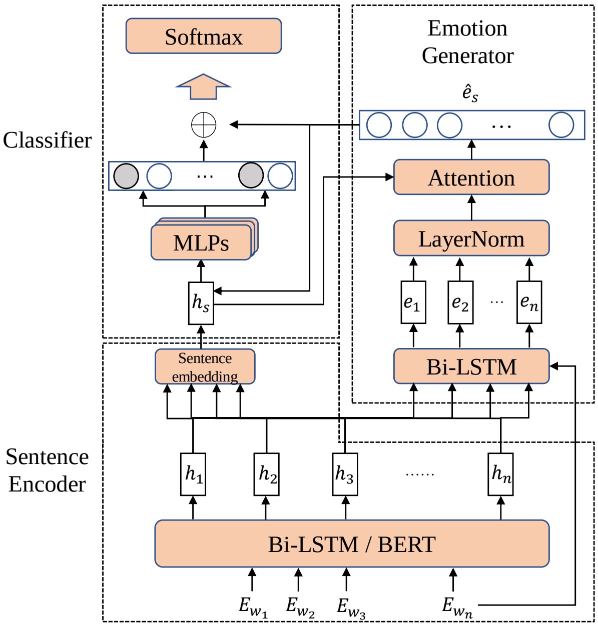


Figure 1: Overview of the Emotion Network.

Model Description Figure 1 illustrates the architecture of the proposed emotional network. There are three parts in the EmNet: sentence encoder, emotion generator, and emotion classifier. EmNet first encodes the input sentence S into semantic hidden states $\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_i, ..., \mathbf{h}_n$ via sentence encoder and generates sentence embedding $\mathbf{h}_s$. Then the hidden states $\mathbf{h}_i$ are used to generate context-aware emotion representations $\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_i, ..., \mathbf{e}_n$ for each word through emotion generator. e_i is a vector with K dimension, K = $|\Phi|$. Thus each element e_i^k in $\mathbf{e}_i$ represents the degree that word w_i expressed on the k-th emotion. Based on the word emotion $\mathbf{e}_i$, the emotion generator calculates the sentence emotion $\hat{\mathbf{e}}_s = \{\hat{e}_s^1, ..., \hat{e}_s^k, ..., \hat{e}_s^K\}$ [1] by an attention layer between $\mathbf{h}_s$ and each $\mathbf{e}_i$, where $\hat{e}_s^k$ is a scalar represents the degree on the k-th emotion of the input sentence. Finally, the classification layer takes $\mathbf{h}_s$ and $\hat{\mathbf{e}}_i$ as input and outputs the classification results. We will describe the components in detail in the following sections.

2.2 Sentence Encoder

We use two architectures as sentence encoder, one is the standard bi-directional Long Short Term Memory (Bi-LSTM) network (Hochreiter and Schmidhuber, 1997) and the other is the pretrained language model BERT (Devlin et al., 2019). The two kinds of encoders have some differences and we will introduce them respectively.

Bi-LSTM Encoder For input sentence S, a forward and a backward LSTMs are used to encode the sentence. We denote $\overrightarrow{\mathbf{h}}_i$ as the i-th hidden states from the forward LSTM and $\overleftarrow{\mathbf{h}}_i$ as the state from the backward LSTM. The final state is the concatenation of the two, $\mathbf{h}_i = [\overrightarrow{\mathbf{h}}_i; \overleftarrow{\mathbf{h}}_i]$. To acquire the sentence embedding, we average the hidden states $\mathbf{h}_s = \frac{1}{n}\sum_{i=0}^{n} \mathbf{h}_i$.

BERT Encoder We also introduce a BERT (Pre-training of Deep Bidirectional Transformers for Language Understanding) (Devlin et al., 2019) based encoder which is a powerful and effective pretrained model. We let the BERT model take S as the first segment, the second segment is set to null. Thus the input of BERT is defined as "$[CLS]$ $w_1, ..., w_i, ...w_n$ $[SEP][SEP]$". Position, segment, and token embeddings are added and fed into the self-attention layers. After encoding the segment, we use the contextual representations $\mathbf{h}_s = \mathbf{T}_{CLS}$ as the sentence embedding and collect the hidden states $\mathbf{h}_1, ..., \mathbf{h}_i, ..., \mathbf{h}_n$ for emotion generator.

2.3 Emotion Generator

The emotion generator is the same for Bi-LSTM and BERT. Its architecture is a Bi-LSTM network as shown in Figure 1. The input of the emotion generator is the concatenation of word embedding and hidden states from the sentence encoder $[\mathbf{h}_i; \mathbf{E}_{w_i}]$. Its output is the emotion representations $\mathbf{e}_1, ..., \mathbf{e}_i, ..., \mathbf{e}_n$, where $\mathbf{e}_i$ is a K dimension vector and K is the number of emotions which varies among tasks. To calculate $\mathbf{e}_i$, we first use a Bi-LSTM to encode $[\mathbf{h}_i; \mathbf{E}_{w_i}]$ as follows,

$$\overrightarrow{\mathbf{e}}_i = \text{LSTM}([\mathbf{h}_i; \mathbf{E}_{w_i}], \overrightarrow{\mathbf{e}}_{i-1}, \theta_f)$$
$$\overleftarrow{\mathbf{e}}_i = \text{LSTM}([\mathbf{h}_i; \mathbf{E}_{w_i}], \overleftarrow{\mathbf{e}}_{i+1}, \theta_b)$$

where θ_f and θ_b denote all the parameters in the forward and backward LSTM, both the dimensions of $\overrightarrow{\mathbf{e}}_i$ and $\overleftarrow{\mathbf{e}}_i$ are K. Then we compute final $\mathbf{e}$ as Equation 2,

$$\mathbf{e}_i = \text{LayerNorm}(\frac{1}{2}(\overrightarrow{\mathbf{e}}_i + \overleftarrow{\mathbf{e}}_i)) \tag{1}$$

The "LayerNorm" is the layer normalization proposed by Ba et al. (2016). We constrain that the dimension of emotion representations equals to the number of emotion types $|\Phi|$. The k-th element e_i^k in $\mathbf{e}_i$ where $0 \leq k < K$ corresponds to the j-th emotion in Φ. Thus $\mathbf{e}_i$ is similar with the human annotated emotion dictionaries where each dimension defines the emotion components in Φ. The difference is that the emotion in $\mathbf{e}_i$ is learned from the training corpus which avoids the mismatch problem. In addition, $\mathbf{e}_i$ of the word w_i is dynamically generated according to a certain context. The biggest challenge is how to align the K dimensions in $\mathbf{e}_i$ with the k-th emotion type in Φ. This will be explained in Section 2.5.

[1] In the rest of this paper, we use bold characters to represent vector and normal characters to represent scalar.

2.4 Emotion Classifier

For emotion classification, since emotion words are relatively more important for the model decision, we adopt the widely used attention mechanism (Bahdanau et al., 2014) to select the key words. Specifically, we use the sentence embedding $\mathbf{h}_s$ to obtain the attention weight a_i of $\mathbf{e}_i$ as follows:

$$a_i = \frac{\exp(u_i)}{\sum_{j=1}^{n} \exp(u_j)} \tag{2}$$

$$u_i = v^T \tanh(W\mathbf{h}_s + U\mathbf{e}_i) \tag{3}$$

where $W \in R^{K\times *}$, $U \in R^{K\times K}$ and $v \in R^{K}$ are weight matrices, $*$ denotes the hidden size which is decided by different encoders. The final sentence level emotion representation for S is calculated by

$$\hat{\mathbf{e}_s} = \sum_{i=1}^{n} a_i \mathbf{e}_i \tag{4}$$

$\hat{\mathbf{e}}_s = \{\hat{\mathrm{e}}_s^1, ..., \hat{\mathrm{e}}_s^k, ..., \hat{\mathrm{e}}_s^K\}$ is also a K dimension vector.

Then we apply a Multilayer Perceptron (MLP) with one hidden layer on the concatenation of $\mathbf{h}_s$ and $\hat{\mathbf{e}}_s$ for each emotion type l_k in Φ as the following equation,

$$\mathbf{o}_k = W_k([\mathbf{h}_s; \hat{\mathbf{e}}_s]) + b_k \tag{5}$$

where W_k and b_k are weight matrix. $\mathbf{o}_k = \{o_k^0, o_k^1\}$ is a two dimension vector which can be used by a softmax function to predict the probability.

2.5 Joint Training and Inference

To guarantee the dimensions in $\mathbf{e}_i$ learn reasonable emotions in $|\Phi|$, we propose to align the emotion representations with the K emotion types in $|\Phi|$. One straightforward way is to add loss functions on each dimensions of $\mathbf{e}_i$ and use the emotion labels of the sentence to supervise its words. However, our experiment shows that this way is too hard and would force all words to learn the same emotion distribution which is unreasonable. In this section, we propose a soft strategy to jointly optimize the classification and emotion lexicon. To align the dimensions in $\hat{\mathbf{e}}_s$ with K emotion types, we add the k-th dimension $\hat{e}_s^k$ to o_k^1 from the k-th MLP layer and compute $\hat{\mathbf{o}}_k$ as Equation 6,

$$\hat{\mathbf{o}}_k = \{\hat{o}_k^0 = o_k^0,\ \hat{o}_k^1 = o_k^1 + \lambda \hat{e}_s^k\} \tag{6}$$

where λ is a pre-defined hyper-parameter. The reason we add $\hat{e}_s^k$ to o_k^1 is to measure how much contribution $\hat{e}_s^k$ makes to the final decision. In this way $\mathbf{e}_i$ is connected with the emotion types in Φ. We apply the softmax function on $\hat{o}_k$ for classification.

Formally, we train all the three components on a set of training examples $\{[S_k, \mathbf{l}_k]\}_{k=1}^{K}$. The training objective is:

$$J(\theta) = \arg\max_{\theta} \sum_{k=1}^{K} \log P(\mathbf{l}_k | S_k, \lambda\hat{\mathbf{e}}_s; \theta) \tag{7}$$

The objective consists of two parts: classification measures the accuracy of the ultimate classification task, and lexicon measures the accuracy of dynamic annotation of the emotion lexicons in the input. We use the Cross Entropy loss for classification. Once a model is trained, we select the emotion candidate with the highest classification scores. For each emotion type, we set the positive threshold to 0.4 and negative threshold to 0.6. Because we find that this setting performs slightly better than 0.5 and 0.5 on the development set. In the rest of this paper, we use **LSTM+EmNet** to represent the Bi-LSTM-based Emotion Network and **BERT+EmNet** for BERT-based Emotion Network.

3 Experiments

3.1 Setup

Dataset We use the English subset of Twitter dataset provided by SemEval 2018 (Mohammad et al., 2018). The dataset contains 11 emotions: *anger, anticipation, disgust, fear, joy, love, optimism, pessimism, sadness, surprise* and *trust*. The training data contains 6,838 tweets. The development and test sets have 886 and 3,259 tweets respectively.

The data prepropssing in the LSTM and BERT models are different. For LSTM models we preprocess the corpus following (Baziotis et al., 2018) where the ekphrasis2 (Baziotis et al., 2017) tool is used. The preprocessing steps included in ekphrasis are: Twitter-specific tokenization, spell correction, word normalization, word segmentation (for splitting hashtags) and word annotation.The BPE (Sennrich et al., 2015) is not applied and 800K unique words are collected. For BERT models, we just use the default preprocessing procedures in BERT including tokenization and BPE to preprocess the corpus.

Evaluation Metrics We use the official competition metric provided by SemEval 2018 for comparison that is the multi-label accuracy (or Jaccard index) (Mohammad et al., 2018). Multi-label accuracy is defined as the size of the intersection of the predicted and gold label sets divided by the size of their union. This measure is calculated for each tweet t, and then is averaged over all tweets T in the dataset:

$$\text{Accuracy(Jaccard)} = \frac{1}{|T|} \sum_{t \in T} \frac{G_t \cap P_t}{G_t \cup P_t} \tag{8}$$

where G_t is the set of the gold labels for tweet t, P_t is the set of the predicted labels for tweet t, and T is the set of tweets. Apart from Jaccard, following Mohammad et al. (2018), we also calculated F1-micro and F1-macro as secondary evaluations metrics, whose definitions are provided on the task webpage. [2]

Baselines We compare our proposed methods with the following baselines:

- **LSTM baseline** A baseline model based on the LSTM network. We remove the emotion generator of our Emotion Network in Figure 1 and directly use h_s for classification.

- **BERT baseline** A baseline model based on BERT. Similar to the LSTM baseline, we remove the emotion generator in Figure 1 and use the [CLS] embedding for classification.

- **NTUA-SLP** The Rank 1 method of SemEval-2018 Task 1 proposed by Baziotis et al. (2018). The model is a two-layer LSTM network where external emotion lexicons are used to provide word level affective knowledge.

- **TCS Research** The Rank 2 method of SemEval-2018 Task 1 proposed by Meisheri and Dey (2018). The model uses two BiLSTM networks to encode tweets from different aspects. Then they concatenated the hidden states for the final classifications.

- **DATN-2** A transfer learning method proposed by Yu et al. (2018) for emotion classification in tweets. They used a shared-private architecture with the dual attention mechanism to encode tweets into features.

- **BERT$_{base}$+DK and BERT$_{large}$+DK** Ying et al. (2019) proposed to integrate domain knowledge into BERT for emotion classification. We compare with both their BERT$_{base}$ and BERT$_{large}$ models.

[2] https://competitions.codalab.org/competitions/17751

ID	Method	Accuracy	F1-micro	F1-macro	Average
1	NTUA-SLP (Baziotis et al., 2018)	58.8	70.1	52.8	60.6
2	TCS Research (Meisheri and Dey, 2018)	58.2	69.3	53.0	60.2
3	DATN-2 (Yu et al., 2018)	58.3	-	54.4	-
4	BERT$_{base}$+DK (Ying et al., 2019)	59.1	71.3	54.9	61.8
5	BERT$_{large}$+DK (Ying et al., 2019)	59.5	71.6	56.3	62.5
6	Bi-LSTM Baseline	56.6	68.3	49.2	58.0
7	6 + EmNet	$59.0^{\dagger}$	$70.1^{\dagger}$	$55.5^{\dagger}$	61.5
8	BERT$_{base}$ Baseline	58.0	70.1	53.0	60.3
9	8 + EmNet	**59.6**†	**71.6**†	**56.5**†	**62.6**

Table 2: Results of multi-label emotion classification on SemEval-2018. The results of "DATN-2" and 'BERT$_{base}$+DK " are taken from their papers. For "TCS Research" and "NTUA-SLP", we cite the number from the official lead-board. † means the results is statistically significant with $p < 0.01$ compared with the corresponding baseline (i.e. Bi-LSTM baseline or BERT baseline). The numbers in bold refers to the highest score and "-" means the number is not applicable.

Implementation Details The basic settings of LSTM-based models follow Baziotis et al. (2018), where the embedding size is set to 300 and the dimension of hidden states in sentence encoder is 618. The word embeddings are taken from Baziotis et al. (2018) which is Twitter-specific word embeddings pretrained on large-scale tweets by *word2vec* algorithm (Mikolov et al., 2013). The final vocabulary size is 800K. The out-of-vocabulary words are simply replaced by a "UNK" symbol. We set the max length to 128. The batch size is set to 128. For the emotion generator, the dimension of the hidden states is set to 11 —— the number of the emotion types.

For BERT-based models, the model implementation is based on the PyTorch version [3]. We use the BERT-base architecture for all experiments where the hidden size is 768. All the texts are tokenized by the BERT tokenizer. For word pieces of one word, we just treat them as individual word. The max length is 512. The hidden dimension of the Bi-LSTM in emotion generator is set to 11 as well. The batch size is set to 32.

All model parameters except the pretrained ones are initialized randomly with Gaussian distribution (Glorot and Bengio, 2010). The stochastic gradient descent (SGD) algorithm is used to tune parameters. In the update procedure, AdamW (Loshchilov and Hutter, 2017) algorithm is used with a learning rate of 5e-5 for BERT based models and Adam (Kingma and Ba, 2014) algorithm is used with a learning rate of 1e-3 for LSTM based models. The pretrained parameters are also updated during training. All of the models are trained on 4 NVIDIA GTX-1080 GPUs.

3.2 Results

Table 2 shows the comparison between our methods and the baselines. The first blocks are the state-of-the-art models on SemEval-2018 Task 1, where we directly cite the results from their paper or the lead-board.

We first consider the LSTM-base models. As expected, our Bi-LSTM baseline of ⑥ has the worst average performance compared with ①-③ as the three facilitate the LSTM model with either transfer learning or external emotion lexicons. When we add our EmNet to ⑥, the results of ⑦ achieves the best performance among the LSTM models. Comparing ⑦ with ① the top 1 model which leverages external emotion lexicons, our model gains 0.2 more score on Accuracy and shows more consistent improvement on F1-macro score (+2.7 points). In terms of average performance, ⑦ achieves 0.9 more scores. This is mainly because our EmNet can jointly model the emotion lexicon and classification, where the emotion lexicon is dynamically built. The results demonstrate that our context-aware emotion lexicons are more effective and can avoid the mismatch problem suffered by external resource.

[3] https://github.com/huggingface/transformers

Emotion	LSTM	+EmNet	BERT	+EmNet
anger	76.4	**78.6 (+2.2)**	78.9	**79.4 (+0.5)**
anticipation	9.9	**27.4 (+17.5)**	11.1	**24.6 (+13.5)**
disgust	73.6	**74.3 (+0.7)**	75.7	**76.2 (+0.5)**
fear	69.9	**71.5 (+1.6)**	74.9	**77.0 (+2.1)**
joy	83.3	**83.9 (+0.6)**	83.5	**84.9 (+1.4)**
love	50.3	**63.2 (+12.9)**	57.3	**63.5 (+6.2)**
optimism	72.6	**72.6 (+0.0)**	74.3	**76.0 (+1.7)**
pessimism	22.6	**34.6 (+12.0)**	35.0	**45.5 (+10.5)**
sadness	**69.6**	69.4 (-0.2)	69.6	**70.1 (+0.5)**
surprise	11.5	**19.5 (+8.0)**	17.7	**19.8 (+2.1)**
trust	0.01	**15.7 (+15.7)**	**5.0**	4.9 (-0.1)

Table 3: F1 on binary classification performance on 11 emotion types.

In terms of BERT models, the baseline ⑧ is strong due to the abundant pre-training data and the deep structure. Based on BERT model ④ and ⑤ integrate domain knowledge into both BERT base and large models which achieve consistent improvements. After adding our EmNet to BERT (we only consider BERT base architecture), model ⑨ achieves the best performance on all the three metrics. Though we implement EmNet on the base settings, we achieve better results than the BERT large model ⑤ which shows the effectiveness of our EmNet.

3.3 Performance of Classification on Different Emotions

In this section, we test the performance on the emotion types separately in terms of testset. When considering a certain type, we use the F1-score as metric which is the harmonic mean of precision and recall. Table 3 shows the performance (F1-score) on 11 emotion types. Note that the score of "trust" is very low due to the low percentage of occurrence in both training data and testset. For both the LSTM and BERT model, EmNet achieves improvements in ten out of the eleven types. This is mainly due to the capacity of EmNet that can assign words with context-aware emtions, thus emotion ambiguity problem can be alleviated effectively.

3.4 Effect of λ

In this part, we discuss the effect of λ in Equation 7. Figure 2 shows the accuracy changes in terms of different λ. The blue lines refer to the BERT models and the red lines refer to the LSTM models. The x-axis is λ and y-axis is accuracy. We can see that the EmNet achieves the highest accuracy when λ is set to 1. Thus in all the experiments, we set $\lambda = 1$. It is interesting to find that when $\lambda = 0$, our EmNet still outperform the baseline models. This could be caused by that EmNet can learn emotion knowledge implicitly without $\lambda \hat{e}_s^k$ in Equation 6, which still helps the classification.

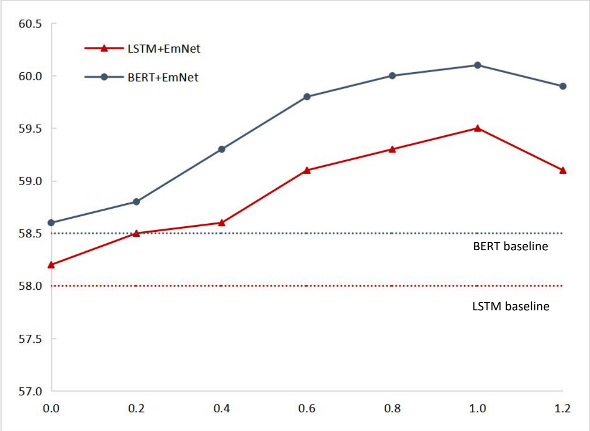


Figure 2: Performance changes in terms of λ on development set.

3.5 Analysis of Learned Emotion Lexicons

As aforementioned, one strong point of our approach over the external emotion lexicon methods is the possibility to dynamically adapt the lexicon to different domains. To validate our claim, we compare our learned emotion lexicons with the external dictionaries used in Baziotis et al. (2018) on the SemEval-2018 task 1, which contains 11 distinct emotion labels. Table 4 lists the results. As seen, the external lexicon dictionary only has 4 overlapped emotion labels, which covers 15.3% emotion vocabulary. These results confirm our claim that the external lexicon methods suffer from the domain mismatch problem. The proposed EmNet approach solves this problem by dynamically learning emotions for each word in different context, which can cover all the words and emotions in the given task.

We give a human evaluation of the learned word level emotions. 3 experienced annotators are invited. We select 100 tweets from the testset. For each sentence, our EmNet computes attention weights for its words and generate emotion values for each word. We simply select 2 words with the highest attention weights as the indicative words in the classification, together with their 2 emotions with the largest emotion values. Totally, we have 200 words to annotate. The annotators are asked whether the words are reasonably selected and whether their emotions are correctly predicted. The final accuracy is computed as follows:

Lexicons	**# Label**	**Coverage Ratio**	
		Label	*Word*
External	4	36.4%	15.3%
Ours	11	100%	100%

Table 4: Statistic of the external lexicon and our learned lexicon.

$$\text{Accuracy} = \frac{1}{200}\sum_{i=1}^{100}\sum_{j=0}^{1}\delta_{w_i^j} \tag{9}$$

where w is one of the 200 words, i is the word index and j is the index of the two emotions. $\delta_{w_i^j} = 1$ only when the w_i and its jth emotion are both annotated reasonable. The final accuracy is the average result of the 3 annotators and we get 72.9%. This illustrates that our EmNet can select reasonable words and predict high quality emotions.

3.6 Visualization of Word Emotion and Attention

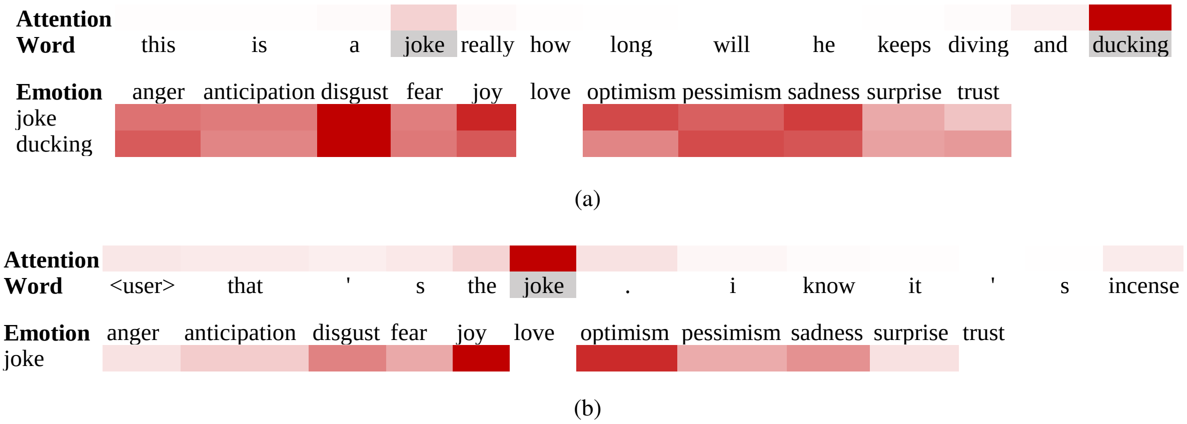
In this section, we give a qualitative analysis to help to understand how EmNet works. When classifying a given sentence, there exist differences in the contributions of different words. The EmNet model is able to generate dynamic word emotions and select the most informative words using attention mechanism. The visualization of the attention layer and learned emotion representations are shown in Figure 3. We focus on the tweets T1 and T2 in Table 1. We take the results from the LSTM+EmNet as examples. Both two cases are correctly predicted by our model.

For T1 in Figure 3 (a), it can be seen that the attention focuses on "joke" and "ducking", this means they are the most informative words in model decision. Then we select the two words and show their emotions in below. The two words in this case are more likely to represent *disgust* emotion. For T2 in Figure 3 (a), the attention is paid to word "joke". When we show its emotion in this context, we find that this time "joke" is more likely to represent *joy*. This illustrates that our model can effectively capture the word level emotions based on certain context, which help to facilitate the classification accuracy.

4 Related Work

Emotion classification has been extensively studied due to its wide applications in recent years. Different from sentiment classification which can be treated as either a single-label classification task (e.g., positive, negative), emotion classification or affect detection is a multi-label classification task which is to detect a discrete set of emotions present in a given sentence such as anger, joy, sadness etc (Dalgleish and Power, 2000; Plutchik, 2001).

Traditional methods such as lexicon, n-gram and graph models have been used. Xu et al. (2012) proposed a coarse-to-fine strategy for multi-label emotion classification. They dealt with the data sparseness problem by incorporating the transfer probabilities from the neighboring sentences to refine the emotion categories. Li et al. (2015) recast multi-label emotion classification as a factor graph inferring problem in which the label and context dependence are modeled as various factor functions. Yan and Turtle (2016)



(a)

(b)

Figure 3: Visualization of attention weights and word emotions for tweets in Table 1. (a) the case study of T1 in Table 1. (b) the case study of T2 in Table 1. The color in deep means more weights. We highlight the words with larger attention weights.

built a separate binary classifier for each emotion category to detect if an emotion category were present or absent in a tweet with traditional unigram features.

The neural network models have also been used in emotion classification. For example, Zhou et al. (2016) proposed an emotion distribution learning (EDL) method, which first used recursive auto-encoders (RAEs) to extract features and then conducted multi-label emotion classification by incorporating the label relations into the cost function. He and Xia (2018) provided an end-to-end learning framework by integrating representation learning and multi-label classification in one neural network. Recently, external knowledge has been widely employed for this task. One representative research line is the transfer learning. Yu et al. (2018) proposed a new transfer learning architecture to divide the sentence representation into two different feature spaces, which are expected to respectively capture the general sentiment words and the other important emotion-specific words via a dual attention mechanism. They transferred the sentiment classification knowledge to emotion classification tasks. Baziotis et al. (2018) proposed a Bi-LSTM architecture equipped with a multi-layer self attention mechanism. The attention mechanism can identify salient words in tweets, as well as gain insight from the models and make them easier to interpret. They leveraged a small scale annotated emotion dictionary and treated the annotated as fixed word affective features. They achieved the highest accuracy in SemEval 2018 workshop. Though the external dictionaries are effective, they are always task specific, different in granularity and limited in scale. Different from these work, we propose an emotion network to jointly learn emotion lexicons and classification. The learned emotion lexicons depend on a certain context, which are effective for emotion disambiguation and avoid the domain mismatch problem.

5 Conclusion and Future Work

In this paper, we propose a novel emotion network. Our model can jointly learn word emotions and conduct classification. We apply the emotion network to both LSTM and BERT models. Experimental results on the public Twitter dataset show that our model can learn reasonable word emotions, which can boost the classification and achieve significant improvements over several baseline models. Further analyses demonstrate the effectiveness of the proposed methods and intuitively interpret how our model works. In future work, along this research direction, we will try to apply our method to other tasks such as aspect-level classification to verify the effectiveness.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

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.

Christos Baziotis, Nikos Athanasiou, Alexandra Chronopoulou, Athanasia Kolovou, Georgios Paraskevopoulos, Nikolaos Ellinas, Shrikanth Narayanan, and Alexandros Potamianos. 2018. Ntua-slp at semeval-2018 task 1: Predicting affective content in tweets with deep attentive rnns and transfer learning. *arXiv preprint arXiv:1804.06658*.

Tim Dalgleish and Mick Power. 2000. *Handbook of cognition and emotion*. John Wiley & Sons.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Aistats*, volume 9, pages 249–256.

Huihui He and Rui Xia. 2018. Joint binary neural network for multi-label learning with applications to emotion classification. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 250–259. Springer.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

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

Shoushan Li, Lei Huang, Rong Wang, and Guodong Zhou. 2015. Sentence-level emotion classification with label and context dependence. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1045–1053.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*.

Hardik Meisheri and Lipika Dey. 2018. TCS research at SemEval-2018 task 1: Learning robust representations using multi-attention architecture. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 291–299, New Orleans, Louisiana, June. Association for Computational Linguistics.

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.

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, New Orleans, Louisiana, June. Association for Computational Linguistics.

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.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Jun Xu, Ruifeng Xu, Qin Lu, and Xiaolong Wang. 2012. Coarse-to-fine sentence-level emotion classification based on the intra-sentence features and sentential context. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 2455–2458. ACM.

Jasy Liew Suet Yan and Howard R Turtle. 2016. Exposing a set of fine-grained emotion categories from tweets. In *25th International Joint Conference on Artificial Intelligence*, page 8.

Wenhao Ying, Rong Xiang, and Qin Lu. 2019. Improving multi-label emotion classification by integrating both general and domain knowledge. *W-NUT 2019*, page 316.

Jianfei Yu, Luís Marujo, Jing Jiang, Pradeep Karuturi, and William Brendel. 2018. Improving multi-label emotion classification via sentiment classification with dual attention transfer network. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1097–1102, Brussels, Belgium, October-November. Association for Computational Linguistics.

Deyu Zhou, Xuan Zhang, Yin Zhou, Quan Zhao, and Xin Geng. 2016. Emotion distribution learning from texts. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 638–647.