HiTrans: A Transformer-Based Context- and Speaker-Sensitive Model for Emotion Detection in Conversations

Jingye Li[1], **Donghong Ji**[1], **Fei Li**[1,*], **Meishan Zhang**[2], **Yijiang Liu**[1]
[1]Key Laboratory of Aerospace Information Security and Trusted Computing,
Ministry of Education, School of Cyber Science and Engineering, Wuhan University, China
[2]School of New Media and Communication, Tianjin University, China
{theodorelee, dhji, cslyj}@whu.edu.cn
{foxlf823, mason.zms}@gmail.com

Abstract

Emotion detection in conversations (EDC) is to detect the emotion for each utterance in conversations that have multiple speakers. Different from the traditional non-conversational emotion detection, the model for EDC should be context-sensitive (e.g., understanding the whole conversation rather than one utterance) and speaker-sensitive (e.g., understanding which utterance belongs to which speaker). In this paper, we propose a transformer-based context- and speaker-sensitive model for EDC, namely **HiTrans**, which consists of two hierarchical transformers. We utilize BERT as the low-level transformer to generate local utterance representations, and feed them into another high-level transformer so that utterance representations could be sensitive to the global context of the conversation. Moreover, we exploit an auxiliary task to make our model speaker-sensitive, called pairwise utterance speaker verification (PUSV), which aims to classify whether two utterances belong to the same speaker. We evaluate our model on three benchmark datasets, namely EmoryNLP, MELD and IEMOCAP. Results show that our model outperforms previous state-of-the-art models.

1 Introduction

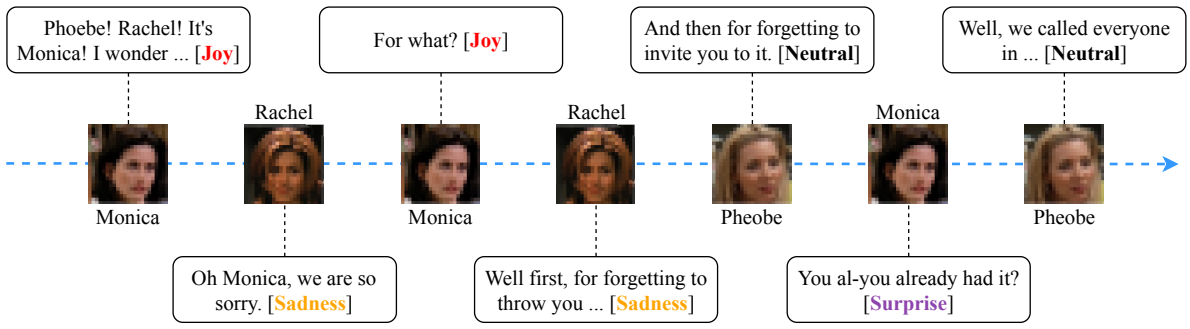


Figure 1: An example of emotion detection in conversations.

Emotion detection in conversations (EDC), whose objective is to detect the emotion for each utterance in conversations (Wen and Wan, 2014; Li et al., 2015), has received increasing attention in the natural language processing (NLP) community (Majumder et al., 2019; Zhong et al., 2019; Ghosal et al., 2019) due to its widely applications such as opinion mining (Cambria et al., 2017) and social media analysis (Majumder et al., 2019). As illustrated in Figure 1, there may be multiple speakers and utterances in the conversation and an EDC model needs to detect the emotion for each utterance from these speakers. Therefore, different from the traditional non-conversational emotion detection, the emotion of an utterance usually depends on the context of the whole conversation. In addition, the personality of a speaker

*Corresponding author.

Proceedings of the 28th International Conference on Computational Linguistics, pages 4190–4200
Barcelona, Spain (Online), December 8-13, 2020

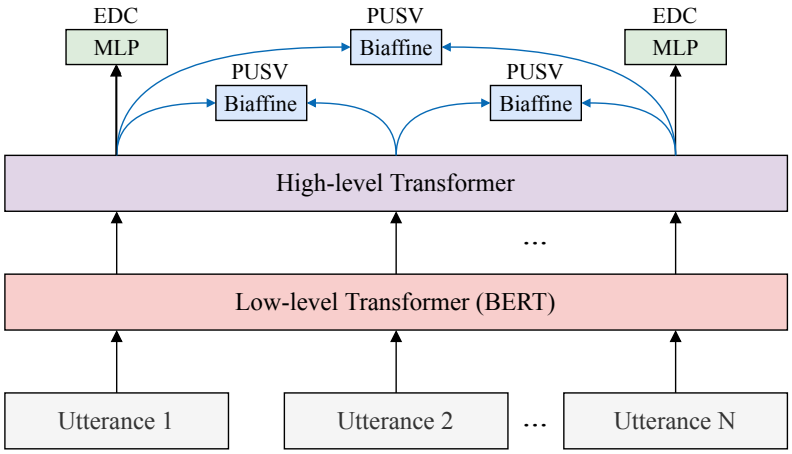


Figure 2: The high-level overview of our model HiTrans. EDC represents Emotion Detection in Conversations. PUSV represents Pairwise Utterance Speaker Verification.

may sometimes influence the emotion of an utterance. Follow previous work (Zhang et al., 2019), we call such characteristics as context sensitivity and speaker sensitivity.

Previous studies for EDC can be roughly divided into two categories, namely sequence-based methods and graph-based methods. Sequence-based methods treat the EDC task as a sequence labeling problem and assign an emotion label to each utterance (Poria et al., 2017; Majumder et al., 2019; Jiao et al., 2019). They usually leverage end-to-end neural sequence labeling models such as long short-term memories (LSTMs) (Poria et al., 2017) and gated recurrent units (GRUs) (Majumder et al., 2019; Jiao et al., 2019), which are capable of capturing long-distance context information from conversations. However, such methods usually neglect the relationships between utterances and speakers.

In contrast, graph-based methods model the context information and utterance-speaker relationships by constructing heterogeneous graphs that take utterances and speakers as vertices and their relationships as edges (Ghosal et al., 2019; Zhang et al., 2019). Then graph convolutional networks (Kipf and Welling, 2017; Zhang et al., 2018) are applied to propagate information among the utterance and speaker vertices. Therefore, the EDC task can be cast as a classification problem for the utterance vertices in the graph. Although graph-based methods have many advantages, it needs manually-defined graph structures and it may also suffer from the sparsity of graphs.

In this paper, we follow the line of sequence-based methods and propose a transformer-based context- and speaker-sensitive model for EDC, namely **HiTrans**, which consists of two hierarchical transformers (Vaswani et al., 2017), as shown in Figure 2. First, we utilize the pre-trained bidirectional transformer encoder (BERT) (Devlin et al., 2019) to generate local utterance representations. Then another high-level transformer is used to capture the global context information in conversations. On top of the model, a multi-layer perceptron (MLP) is used to determine the emotion of an utterance based on its representation. To make our model speaker-sensitive, we employ a biaffine classifier (Dozat and Manning, 2017) to classify whether two utterances belong to the same speaker, called pairwise utterance speaker verification (PUSV). Therefore, our model performs multi-task learning (Caruana, 1997; Liu et al., 2017) to learn from both the EDC and PUSV tasks.

We conduct experiments on three benchmark datasets to verify our models, including EmoryNLP (Zahiri and Choi, 2018), MELD (Poria et al., 2019) and IEMOCAP (Busso et al., 2008). The results show that our model outperforms the previous state-of-the-art models (Poria et al., 2017; Majumder et al., 2019; Ghosal et al., 2019; Zhang et al., 2019; Zhong et al., 2019), and improves the F1s for three benchmark datasets by about 2.4%, 2.5% and 0.3% absolutely. Empirical analysis shows than the PUSV task facilitates the EDC task significantly, improving the F1s by about 1.6%, 0.8% and 1.2% respectively. Experimental results demonstrate the effectiveness of our motivation that models the context sensitivity via transformers and models the speaker sensitivity via an auxiliary task, PUSV.

Our contributions are summarized as follows:

- We propose a hierarchical transformer-based model for emotion detection in conversation, which consists of a low-level transformer for generating local utterance representations, and a high-level transformer for capturing the global context information in a conversation.
- We exploit an auxiliary task to classify whether two utterance belong to the same speaker to make our model speaker-sensitive, called pairwise utterance speaker verification (PUSV).
- Experimental results on three benchmark datasets show that our model outperforms the state-of-the-art models.

2 Related Work

Emotion detection has received increasing attention in recent years. NLP researchers has organized a number of competitions and published several datasets for emotion detection from different granularities of text such as documents (Alm et al., 2005), sentences (Li et al., 2015) and short text (Wang et al., 2012). Besides traditional emotion detection from the static text, conversational emotion detection has also become a research hotspot and many publicly available datasets have been released (Zahiri and Choi, 2018; Poria et al., 2019). Different from traditional emotion detection, there are two characteristics that play important roles in conversational emotion detection. First of all, the context information of the conversation is crucial since speaker's emotions may change during the conversation. Second, the emotions of utterances may be influenced by speakers' personalities sometimes. In the following, we will give an overview of the methods for EDC in previous work.

In early studies, emotion detection in textual conversations is often addressed via feature engineering such as lexicon and acoustic features (Forbes-Riley and Litman, 2004; Devillers and Vidrascu, 2006). As deep learning develops, recent studies begin to treat the EDC task as a sequence labeling problem (Lafferty et al., 2001; Ma and Hovy, 2016) and leverage deep recurrent neural networks (RNNs) to handle it (Poria et al., 2017; Tzirakis et al., 2017; Majumder et al., 2019; Jiao et al., 2019). Sequence-based approaches are able to effectively capture contextual utterance information and long-distance dependency in conversations. Our work also follows this line of work, but employs a more advanced model, namely the transformer (Vaswani et al., 2017). To our knowledge, there is only one prior work that uses the transformer for EDC (Zhong et al., 2019). They focus on knowledge enrichment for the transformer, while we focus on building a context- and speaker-sensitive transformer.

Another line of prior work paid attention to integrate speaker information into the model for EDC, as speaker information is able to affect EDC in a considerable extent (Hazarika et al., 2018a; Hazarika et al., 2018b). Majumder et al. (2019) propose an RNN-based EDC model to track both local and global state dynamically. In addition, graph-based approaches have also been employed for EDC, since they are capable of modeling context- and speaker-sensitivity (Zhang et al., 2019; Ghosal et al., 2019). For example, Zhang et al. (2019) build a conversational graph, where the nodes represent utterances or speakers and the edges represent the dependencies between the speakers and utterances. Then they leveraged a graph convolutional network (Kipf and Welling, 2017; Zhang et al., 2018) to propagate context and speaker information among the utterances. In this work, we explore speaker information for EDC in an alternative way, proposing an auxiliary task to utilize speaker information and improving our model by multi-task learning (Caruana, 1997; Liu et al., 2017).

3 Approach

First of all, we define the EDC task as below: given a conversation with N consecutive utterances $\{u_1, u_2, \cdots, u_N\}$ and M speakers $\{s_1, s_2, \cdots, s_M\}$, the objective of the EDC task is to predict the emotion label for each utterance, such as *Joy* and *Sadness*. Each utterance u_i is uttered by one speaker s_j. The high-level architecture of our model **HiTrans** is shown in Figure 2, which stacks a high-level transformer on a low-level transformer. The low-level transformer generates local utterance representations (Section 3.1) and the high-level one further embeds global context information into utterance representations (Section 3.2). On top of the model, a biaffine classifier performs pairwise utterance speaker verification (Section 3.4) as an auxiliary task to facilitate an MLP to do emotion detection (Section 3.3). We will introduce the details of each part in the following sections.

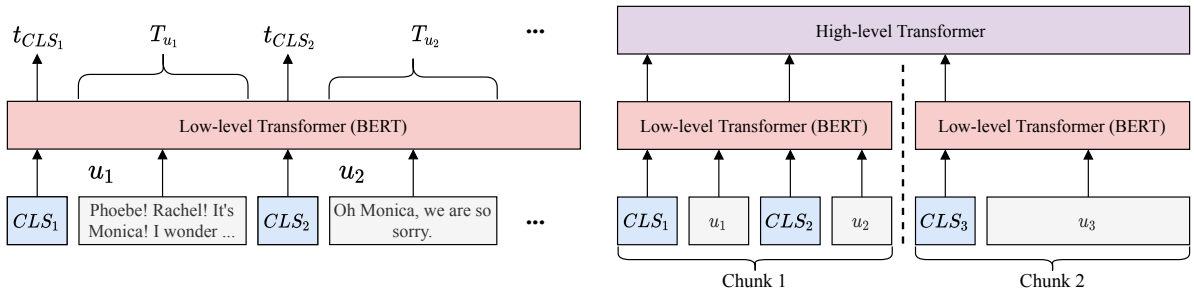


Figure 3: (a) Utterance encoding with the low-level transformer. (b) Context encoding with the high-level transformer.

3.1 Utterance Encoding

As shown in Figure 3(a), we leverage BERT (Devlin et al., 2019) as the low-level transformer to encode utterances since it has been demonstrated to be the-state-of-art model for representation learning in NLP (Wadden et al., 2019; Li et al., 2019). Inspired by Liu and Lapata (2019), we insert the token CLS at the beginning of each utterance and concatenate several utterances in a conversation together as the input of BERT. Concretely, given N utterances $\{u_1, u_2, \cdots, u_N\}$, we insert a token CLS_i before the utterance u_i and obtain a sequence $\{CLS_1, u_1, CLS_2, u_2, \cdots, CLS_N, u_N\}$. Then BERT takes the sequence as input and outputs a sequence $\{\boldsymbol{t}_{CLS_1}, \boldsymbol{T}_{u_1}, \boldsymbol{t}_{CLS_2}, \boldsymbol{T}_{u_2}, \cdots, \boldsymbol{t}_{CLS_N}, \boldsymbol{T}_{u_N}\}$, where $\boldsymbol{t}_{CLS_i} \in \mathbb{R}^h$ denotes the representation of the token CLS_i. $\boldsymbol{T}_{u_i} \in \mathbb{R}^{|u_i| \times h}$ denotes the representations of all the $|u_i|$ words in the utterance u_i. After that, we use the vector $\boldsymbol{t}_{CLS_i}$ as the representation of the utterance u_i.

Since BERT has a length limitation (512 tokens) for input, it is infeasible to handle all the utterances in a conversation simultaneously if the total length of all the utterances exceeds the limitation. To solve this problem, we split the utterances in an overlong conversation into chunks whose lengths are less than 512 tokens, as shown in Figure 3(b). Then each chunk is fed into BERT to obtain the representation for each utterance in this chunk. For example, assuming that there are 3 utterances, $\{u_1, u_2, u_3\}$, in a conversation. The length of u_1 and u_2 is no more than 512, but the length will exceed if adding u_3. Therefore, we will take u_1 and u_2 as a chunk and input it into BERT first, and then take u_3 as another chunk. In this way, we can obtain the representations $\boldsymbol{t}_{[CLS]_1}$, $\boldsymbol{t}_{[CLS]_2}$ and $\boldsymbol{t}_{[CLS]_3}$ for u_1, u_2 and u_3 respectively. However, the representation $\boldsymbol{t}_{[CLS]_3}$ of the utterance u_3 has no sense about $\boldsymbol{t}_{[CLS]_2}$ and $\boldsymbol{t}_{[CLS]_3}$ since they are input into BERT separately. Thus, the global context information in the conversation is not able to be captured. We will explain how to solve this problem in the next section.

3.2 Context Encoding

To capture the global context information in a conversation, we stack another transformer on top of BERT as shown in Figure 3(b). The high-level transformer takes the representations $\{\boldsymbol{t}_{[CLS]_1}, \boldsymbol{t}_{[CLS]_2}, \cdots, \boldsymbol{t}_{[CLS]_N}\}$ of all the utterances $\{u_1, u_2, \cdots, u_N\}$ as input. Following the standard transformer (Vaswani et al., 2017), we also add position embeddings to the utterance representations to model the relative positions of the utterances in the conversation. If the outputs of this transformer are denoted as $\{\boldsymbol{r}_1, \boldsymbol{r}_2, \cdots, \boldsymbol{r}_N\}$, they can be considered as the representations of the utterances $\{u_1, u_2, \cdots, u_N\}$ and meanwhile, they capture the global context information in the conversation and long-distance dependency among the utterances, due to the help of self-attention in the transformer.

3.3 Emotion Detection in Conversations (EDC)

When the final representation $\boldsymbol{r}_i$ for an utterance u_i are ready, we can calculate the probabilities $\boldsymbol{p}^{EDC}$ of all candidate emotion labels such as *Joy*, *Sadness* and *Neutral* by an MLP:

$$\boldsymbol{p}^{EDC} = \text{softmax}(\text{MLP}^{EDC}(\boldsymbol{r}_i)). \quad (1)$$

3.4 Pairwise Utterance Speaker Verification (PUSV)

The objective of PUSV is to classify whether two utterances u_i and u_j in a conversation are from the same speaker. The objective of PUSV is to classify whether two utterances u_i and u_j in a conversation are from the same speaker. Inspired by Dozat and Manning (2017), we apply a biaffine classifier to perform the PUSV task. In addition, we adopt an MLP before feeding the representations $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$ of u_i and u_j into the biaffine classifier, formalized as:

$$\begin{aligned} \boldsymbol{h}_i &= \mathrm{MLP}^{PUSV}(\boldsymbol{r}_i) \\ \boldsymbol{h}_j &= \mathrm{MLP}^{PUSV}(\boldsymbol{r}_j) \\ \boldsymbol{p}^{PUSV} &= \mathrm{softmax}(\boldsymbol{h}_i^{\top} \boldsymbol{W}_1 \boldsymbol{h}_j + \boldsymbol{W}_2(\boldsymbol{h}_i \oplus \boldsymbol{h}_j) + \boldsymbol{b}) \end{aligned} \tag{2}$$

where $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ respectively denote the weight matrix of the bi-linear and the linear terms and $\boldsymbol{b}$ is the bias item in the biaffine classifier, $\oplus$ denotes the concatenation operation, and $\boldsymbol{p}^{PUSV}$ indicates the probabilities that two utterances u_i and u_j belong to the same speaker or not.

3.5 Training

We optimize the model by minimizing the cross-entropy losses of both the EDC and PUSV tasks. For a single conversation, the objective function of the EDC task is defined as follows:

$$\mathcal{L}^{EDC} = -\frac{1}{N} \sum_{i=1}^{N} \log \boldsymbol{p}_{y_i}^{EDC}, \tag{3}$$

where N is the number of utterances in the conversation and y_i is the gold emotion label for the i-th utterance. For the PUSV task, the objective function is defined as below:

$$\mathcal{L}^{PUSV} = -\frac{1}{C_N^2} \sum_{i=1}^{N} \sum_{j=1, j<i}^{N} \log \boldsymbol{p}_{y_{i,j}}^{PUSV}, \tag{4}$$

where C_N^2 is the pairwise-utterance combination number in the conversation, and $y_{i,j}$ is the ground-truth answer about whether two utterances u_i and u_j come from the same speaker. Finally, we employ multi-task learning (Caruana, 1997; Liu et al., 2017) between the EDC and PUSV tasks to train our model. Instead of using constant weights for the losses of these tasks, we employ dynamic weights during the training stage following the method of homoscedastic uncertainty (Kendall et al., 2018). The final loss is formalized as:

$$\mathcal{L} = \frac{1}{2\sigma_1^2} \mathcal{L}^{EDC} + \frac{1}{2\sigma_2^2} \mathcal{L}^{PUSV} + \log \sigma_1 \sigma_2, \tag{5}$$

where σ_1 and σ_2 are the standard deviations of the EDC and PUSV losses from the training instances in the conversation, respectively.

4 Experiments

4.1 Datasets

We evaluate our model on three benchmark datasets, namely MELD (Zahiri and Choi, 2018), EmoryNLP (Zahiri and Choi, 2018) and IEMOCAP (Busso et al., 2008), following previous work (Ghosal et al., 2019; Majumder et al., 2019; Zhong et al., 2019). Table 1 shows the statistics of the three datasets. MELD (Poria et al., 2019) is a multi-modal dataset collected from a famous TV show named "Friends". There are seven emotion labels in the dataset, including anger, sadness, disgust, surprise, fear, joy and neutral. EmoryNLP (Zahiri and Choi, 2018) is also collected from the TV show scripts of "Friends". The difference lies in emotion labels, which include neutral, happiness, sadness, anger, frustrated and excited. Different from the above datasets, IEMOCAP (Busso et al., 2008) consists of two-party conversations of ten speakers. Eight of them only appear in the train set, and the remaining

Dataset	Conversations			Utterances			Speakers/Conv			Utterances/Conv		
	train	val	test	train	val	test	train	val	test	train	val	test
EmoryNLP	659	89	79	7,551	954	984	3.5	3.1	3.4	11.5	10.7	12.5
MELD	1,028	114	280	9,989	1,109	2,610	2.7	3.0	2.7	9.7	9.7	9.3
IEMOCAP	120	-	31	5,810	-	1,623	2.0	-	2.0	48.4	-	52.4

Table 1: Statistics of the datasets. The last two columns denote the average speaker number and utterance number in a conversation.

Dataset	*b*	*lr*	*w*	Low-level Transformer				High-level Transformer			
				h	*hd*	*ff*	*l*	*h*	*hd*	*ff*	*l*
EmoryNLP	8	2e-5	1e-5	768	12	3072	12	768	4	768	1
MELD	8	2e-5	1e-5	768	12	3072	12	768	4	768	1
IEMOCAP	4	2e-5	1e-5	768	12	3072	12	768	6	1024	2

Table 2: Hyper-parameter settings. *b*: batch size, *lr*: learning rate, *w*: weight decay rate, *h*: hidden size, *hd*: the number of self-attention heads, *ff*: feed-forward size, *l*: the number of layers.

two speakers appear in the test set. IEMOCAP contains video, audio and text transcriptions. The utterances in the dataset are annotated with one of six emotion labels: happy, sad, excited, frustrated, angry and excited. Since there is no validation set in IEMOCAP, we split a subset from the training dialogues as the validation set.

4.2 Hyper-parameter Settings and Evaluation Metrics

We use PyTorch[1] to implement our model[2]. We tune the hyper-parameters using the validation sets of the datasets. The best value is listed in Table 2. We adopt the Adam optimizer with the batch size 8 or 4, the learning rate $2e-5$ and the weight decay rate $1e-5$ throughout all the experiments. Since we use the "bert-base-uncased" version[3] as our low-level transformer, all the settings are the same with BERT. For the high-level transformer, we set the hidden size, the number of self-attention heads, the feed-forward size and the number of layers as 768, 4, 768 and 1 for the model using in the EmoryNLP and MELD datasets, and 768, 6, 1024 and 2 for the model using in the IEMOCAP dataset. In terms of evaluation metrics, we exploit the standard weighted macro F1-score following previous work (Ghosal et al., 2019; Majumder et al., 2019; Zhong et al., 2019).

4.3 Baselines

We compare our model with the previous state-of-the-art models for emotion detection in conversations, which are listed as below: (1) **TextCNN** (Kim, 2014), a convolutional neural network for utterance-level classification without using contextual information in the conversation. (2) **c-LSTM** (Poria et al., 2017), a hierarchical LSTM model, where both contextual and utterance-level information are adopted. (3) **DialogueRNN** (Majumder et al., 2019), a sequence-based model that is composed of three GRUs to track the states of speakers, global contexts and historical emotions respectively. (4) **KET** (Zhong et al., 2019), a transformer-based model which exploits external commonsense knowledge to enhance contextual utterance representations. (5) **DialogueGCN** (Ghosal et al., 2019), a graph-based model where the nodes represent individual utterances and the edges represent the dependency between the speakers of the utterances. (6) **ConGCN** (Zhang et al., 2019), a graph-based model which exploits the GCN to propagate the information between utterance nodes and speaker nodes.

[1]https://pytorch.org

[2]The code is available at `https://github.com/ljynlp/HiTrans`

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

Method	EmoryNLP	MELD	IEMOCAP
TextCNN (Kim, 2014)	32.59	55.02	48.18
c-LSTM (Poria et al., 2017)	32.89	56.44	54.95
DialogueRNN (Majumder et al., 2019)	31.70	57.03	62.75
DialogueGCN (Ghosal et al., 2019)	–	58.10	64.18
KET (Zhong et al., 2019)	34.39	58.18	59.56
ConGCN (Zhang et al., 2019)	–	59.40	–
HiTrans*	36.60	61.66	63.81
HiTrans	**36.75**	**61.94**	**64.50**

Table 3: The F1s (%) on the test sets of the EmoryNLP, MELD and IEMOCAP datasets. Result with * is based on the fixed-weight loss ($0.5\mathcal{L}^{EDC} + 0.5\mathcal{L}^{PUSV}$) compared with Equation 5.

Method	EmoryNLP	MELD	IEMOCAP
HiTrans	36.75	61.94	64.50
HiTrans-PUSV	35.19 (-1.56)	61.12 (-0.82)	63.39 (-1.11)
HiTrans-PUSV+SE	36.45 (-0.30)	61.84 (-0.10)	60.89 (-3.61)

Table 4: Investigation for the effect of the PUSV task. The numbers in the brackets denote the decreases compared with the first row.

5 Results and Analyses

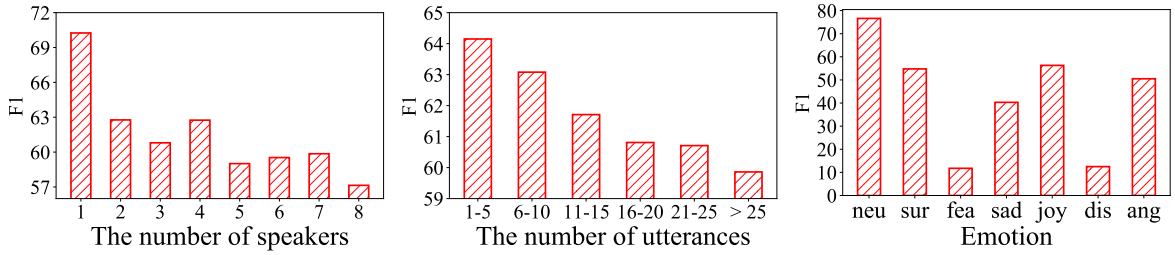
5.1 Comparisons with Previous SOTA Models

Table 3 shows the results of the test sets for the EmoryNLP, MELD and IEMOCAP datasets. As seen, our model achieves better performance than the previous state-of-the-art systems and outperforms the KET by 2.36% in the EmoryNLP dataset, the ConGCN by 2.54% in the MELD dataset and the DialogueGCN by 0.32% in the IEMOCAP dataset. From Table 3, we also observe that graph-based models (Ghosal et al., 2019; Zhang et al., 2019) generally perform better than sequence-based models (Kim, 2014; Poria et al., 2017; Majumder et al., 2019). This demonstrates that conversational emotion detection may benefit from context and speaker information. Therefore, our model that utilizes both kinds of information will mitigate the shortages of sequence-based models. Moreover, a noticeable observation is that our model performs better than the knowledge-enriched model, KET, even without using any external knowledge. On the other hand, this suggests that there is still a certain space to improve our model with the help of external knowledge. Finally, using the dynamic weights instead of static weights for the loss function gives 0.15% increase on EmoryNLP, 0.28% on MELD and 0.69% on IEMOCAP.

5.2 Investigation for the Effect of the PUSV task

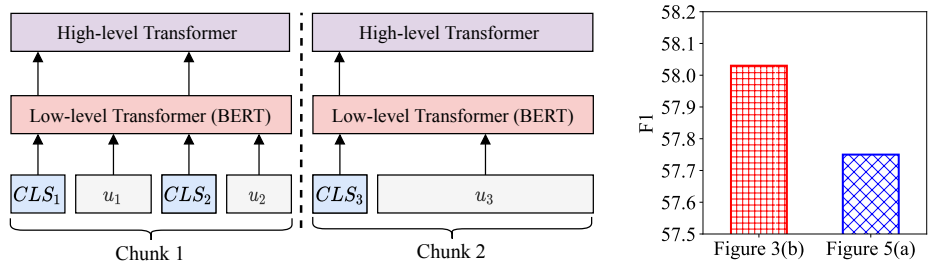
To demonstrate the effectiveness of the PUSV task, we build two baselines by ourselves, namely **HiTrans-PUSV** and **HiTrans-PUSV+SE**. In HiTrans-PUSV, we remove the modules that are related to the PUSV task. In HiTrans-PUSV+SE, we employ the same hierarchical transformer architecture for modeling utterances and conversations, but replace the modules related to the PUSV task with speaker embeddings. Concretely, we design a speaker embedding (SE) for every speaker and add the speaker embedding to every utterance representation in order to allow our model to be speaker-sensitive.

As shown in Table 4, HiTrans-PUSV+SE achieves better performance on EmoryNLP and MELD compared with HiTrans-PUSV. On these two datasets, the F1 improvements are 1.26% and 0.72%, respectively. Note that HiTrans-PUSV+SE attains a lower F1 on the IEMOCAP dataset, which may be because the speakers in the test set do not appear in the train set. Thus, untrained speaker embeddings in the test set become noise and are apt to influence the performance. By contrast, HiTrans outperforms HiTrans-PUSV by 1.56%, 0.82% and 1.11%, and surpasses HiTrans-PUSV+SE by 0.30%, 0.10% and 3.61% in the three datasets. This demonstrates the advantages of the PUSV task since it only needs



(a) (b) (c)

Figure 4: (a) Effect of the speaker number. (b) Effect of the utterance number. (c) Performance of each emotion category. The results come from the experiments in the MELD dataset.



(a) (b)

Figure 5: Performance comparison between the strategy of context encoding in Figure 3(b) and the strategy in Figure 5(a).

the information of speakers but does not need to train the representations of speakers. Therefore, it still works well for the situation that speakers do not occur in the training set.

5.3 Investigation for the Effect of the Speaker, Utterance Number and Emotion

In this section, we investigate the effect of the number of speakers and utterances for emotion detection in conversations. Figure 4(a) shows the F1 change trend as the number of speakers increases. As seen, if the number of speakers increases, the F1 generally decreases, which demonstrates that more speakers bring more challenges for the model to correctly detect emotions in conversations.

In addition, we also investigate the effect of the utterance number for EDC, and the results are shown in Figure 4(b). As seen, the F1 declines as the number of utterances in a conversation goes up. This indicates that it is more difficult for the model to detect emotions correctly if there are more utterances in a conversation. Overall, the observations in Figure 4 are consistent with human intuitions that the more the numbers of speakers or utterances are, the harder emotion detection becomes.

Moreover, we show the performance of each emotion category for the MELD dataset in Figure 4(c). Especially, our model achieves low performances on the *Fear* and *Disgust* emotions, which may be due to imbalanced data, since there are only 2.68% and 2.71% instances labeled with *Fear* and *Disgust* on the train set.

5.4 Comparing the Strategies for Context Encoding

Recall that in Section 3.1 and 3.2, we split the utterance sequence whose length exceeds 512 into chunks. Then each chunk is input into BERT to obtain the representations of the utterances in this chunk. Afterwards, we use another high-level transformer to encode the representations of all the utterances again, in order to make them be aware of each other. Therefore, the whole context information can be learnt by this way.

To show the importance of the whole context information and the effectiveness of our method. We build another strategy, as shown in Figure 5(a). In this strategy, the representations of the utterances in different chunks are input into the high-level transformer separately. Thus, the utterances in a chunk are

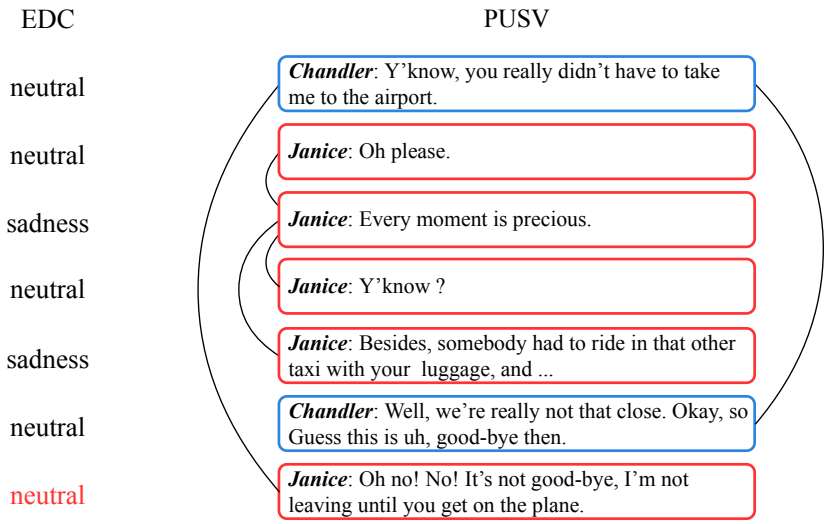


Figure 6: Case study for the predictions of the EDC and PUSV tasks. The label in red denotes the incorrect emotion label predicted by our model. The edges in the PUSV predictions represent our model predicts two utterances belong to the same speaker.

agnostic to the ones in other chunks. As a result, the whole context information cannot be embedded into the utterance representations. Here we study the performance gap between two strategies on the MELD dataset. As shown in Figure 5(b), the F1 of the strategy in Figure 3(b) is about 0.3% higher than the one in Figure 5(a), which demonstrates that the whole context information is helpful for the model to detect emotions in conversations.

5.5 Case Study for the Predictions of the EDC and PUSV Tasks

For better understanding the interaction between the EDC and PUSV tasks, we select a case from the MELD dataset and visualize the predictions for both tasks. As shown in Figure 6, our model predicts the 2nd, 3rd, 4th and 5th utterances belong to the same speaker "Janice", and predicts the 1st, 6th and 7th utterances belong to the same speaker "Chandler". Therefore, our model links the 7th utterance to the incorrect speaker "Chandler". As a result, it is influenced by such prediction when doing the EDC prediction. As seen, the emotion label of the 7th utterance is predicted as "neutral" rather than the correct one "sadness". This may be because the speaker "Chandler" generally expresses the "neutral" emotion in this conversation. If our model is able to link the 7th utterance to the correct speaker "Janice", it may perform a valid prediction since the speaker "Janice" has expressed the emotion "sadness" before.

6 Conclusion

In this work, we propose a transformer-based context- and speaker-sensitive model for emotion detection in conversations. We evaluate our model on three benchmark datasets and demonstrate its effectiveness compared with previous state-of-the-art models. Through experiments and analyses, we find that our model is able to effectively capture the whole context information of conversations and the speaker information. The multi-task learning between the EDC and PUSV tasks indeed helps our model to improve the performance. In the future, we will explore how to integrate our model with graph-based models to better emotion detection in conversations.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61772378 and 6160216). the National Key Research and Development Program of China (No. 2017YFC1200500), the Research Foundation of Ministry of Education of China (No. 18JZD015), the Major Projects of the National Social Science Foundation of China (No. 11&ZD189).

References

Cecilia Ovesdotter Alm, Dan Roth, and Richard Sproat. 2005. Emotions from text: machine learning for text-based emotion prediction. In *Proceedings of NAACL-HLT*, pages 579–586.

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. 2008. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42(4):335.

Erik Cambria, Soujanya Poria, Alexander Gelbukh, and Mike Thelwall. 2017. Sentiment analysis is a big suitcase. *IEEE Intelligent Systems*, 32(6):74–80.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.

Laurence Devillers and Laurence Vidrascu. 2006. Real-life emotions detection with lexical and paralinguistic cues on human-human call center dialogs. In *Proceedings of InterSpeech*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Timothy Dozat and Christopher D Manning. 2017. Deep biaffine attention for neural dependency parsing. In *Proceedings of ICLR*.

Kate Forbes-Riley and Diane Litman. 2004. Predicting emotion in spoken dialogue from multiple knowledge sources. In *Proceedings of NAACL-HLT*, pages 201–208.

Deepanway Ghosal, Navonil Majumder, Soujanya Poria, Niyati Chhaya, and Alexander Gelbukh. 2019. Dialoguegcn: A graph convolutional neural network for emotion recognition in conversation. In *Proceedings of EMNLP-IJCNLP*, pages 154–164.

Devamanyu Hazarika, Soujanya Poria, Rada Mihalcea, Erik Cambria, and Roger Zimmermann. 2018a. Icon: interactive conversational memory network for multimodal emotion detection. In *Proceedings of EMNLP*, pages 2594–2604.

Devamanyu Hazarika, Soujanya Poria, Amir Zadeh, Erik Cambria, Louis-Philippe Morency, and Roger Zimmermann. 2018b. Conversational memory network for emotion recognition in dyadic dialogue videos. In *Proceedings of NAACL-HLT*, pages 2122–2132.

Wenxiang Jiao, Haiqin Yang, Irwin King, and Michael R Lyu. 2019. Higru: hierarchical gated recurrent units for utterance-level emotion recognition. In *Proceedings of NAACL-HLT*, pages 397–406.

Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of CVPR*, pages 7482–7491.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of EMNLP*, pages 1746–1751.

Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of ICLR*.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML*, volume 1, pages 282–289.

Shoushan Li, Lei Huang, Rong Wang, and Guodong Zhou. 2015. Sentence-level emotion classification with label and context dependence. In *Proceedings of ACL*, pages 1045–1053.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2019. A unified mrc framework for named entity recognition. *arXiv preprint arXiv:1910.11476*.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of EMNLP-IJCNLP*, pages 3730–3740.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-task learning for text classification. In *Proceedings of ACL*, pages 1–10.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th ACL*, pages 1064–1074.

Navonil Majumder, Soujanya Poria, Devamanyu Hazarika, Rada Mihalcea, Alexander Gelbukh, and Erik Cambria. 2019. Dialoguernn: An attentive rnn for emotion detection in conversations. In *Proceedings of AAAI*, pages 6818–6825.

Soujanya Poria, Erik Cambria, Devamanyu Hazarika, Navonil Majumder, Amir Zadeh, and Louis-Philippe Morency. 2017. Context-dependent sentiment analysis in user-generated videos. In *Proceedings of ACL*, pages 873–883.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2019. Meld: A multimodal multi-party dataset for emotion recognition in conversations. In *Proceedings of ACL*, pages 527–536.

Panagiotis Tzirakis, George Trigeorgis, Mihalis A Nicolaou, Björn W Schuller, and Stefanos Zafeiriou. 2017. End-to-end multimodal emotion recognition using deep neural networks. *IEEE Journal of Selected Topics in Signal Processing*, 11(8):1301–1309.

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 *Proceedings of NeurIPS*, pages 5998–6008.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. In *Proceedings of the 2019 Conference on EMNLP*, pages 5788–5793.

Wenbo Wang, Lu Chen, Krishnaprasad Thirunarayan, and Amit P Sheth. 2012. Harnessing twitter" big data" for automatic emotion identification. In *Proceedings of SOCIALCOM-PASSAT*, pages 587–592.

Shiyang Wen and Xiaojun Wan. 2014. Emotion classification in microblog texts using class sequential rules. In *Proceedings of AAAI*, pages 187–193.

Sayyed M Zahiri and Jinho D Choi. 2018. Emotion detection on tv show transcripts with sequence-based convolutional neural networks. In *Proceedings of Workshops at AAAI*, pages 44–52.

Yuhao Zhang, Peng Qi, and Christopher D Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on EMNLP*, pages 2205–2215.

Dong Zhang, Liangqing Wu, Changlong Sun, Shoushan Li, Qiaoming Zhu, and Guodong Zhou. 2019. Modeling both context-and speaker-sensitive dependence for emotion detection in multi-speaker conversations. In *Proceedings of IJCAI*, pages 10–16.

Peixiang Zhong, Di Wang, and Chunyan Miao. 2019. Knowledge-enriched transformer for emotion detection in textual conversations. In *Proceedings of EMNLP-IJCNLP*, pages 165–176.