Modeling Local Contexts for Joint Dialogue Act Recognition and Sentiment Classification with Bi-channel Dynamic Convolutions

Jingye Li[†], Hao Fei[†], Donghong Ji[‡]
Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University,Wuhan, China
{theodorelee,hao.fei,dhji}@whu.edu.cn

Abstract

In this paper, we target improving the joint dialogue act recognition (DAR) and sentiment classification (SC) tasks by fully modeling the local contexts of utterances. First, we employ the dynamic convolution network (DCN) as the utterance encoder to capture the dialogue contexts. Further, we propose a novel context-aware dynamic convolution network (CDCN) to better leverage the local contexts when dynamically generating kernels. We extended our frameworks into bi-channel version (i.e., BDCN and BCDCN) under multi-task learning to achieve the joint DAR and SC. Two channels can learn their own feature representations for DAR and SC, respectively, but with latent interaction. Besides, we suggest enhancing the tasks by employing the DiaBERT language model. Our frameworks[1] obtain state-of-the-art performances against all baselines on two benchmark datasets, demonstrating the importance of modeling the local contexts.

1 Intorduction

Dialogue act recognition (DAR) aims to detect speaker's intentions (e.g., *question*, *agreement* or *statement*) in each utterance, which can facilitate dialog systems to produce appropriate responses (Inui et al., 2001). Recent studies have further revealed that simultaneously recognizing the dialog act and detecting the sentiment in dialog can result in better grasping of speaker's intention (Cerisara et al., 2018; Qin et al., 2020). These two tasks are closely relevant, i.e., they mutually promote each other by being jointly performed. On the one hand, the DAR provides clues for sentiment classification (SC). In return, the sentiment transitions also can benefit dialogue act prediction. Taking the utterances in Table 1 as examples, it is quite common that a same sentiment following previous utterance's will be expressed once the dialogue act *Agreement* is assigned. Meanwhile, when the speaker changes the sentiment from *Negative* to *Neutral*, the dialogues act tends to transition into *Statement*.

Speaker	Utterance	Dialogue Act	Sentiment
A	Does anyone ever feel anxious and empty at the same time?	*Question*	*Negative*
B	All the time.	*Answer*	*Negative*
B	I feel like I'm losing my mind a little bit.	*Statement*	*Negative*
A	Relatable. I'm always anxious and if I'm not feeling empty or depressed I'm angry. Also usually dissociating.	*Aggrement*	*Negative*
B	I needa go smoke.	*Statement*	*Neutral*

Table 1: Example utterances from Mastodon dataset for joint dialogue act and sentiment detection.

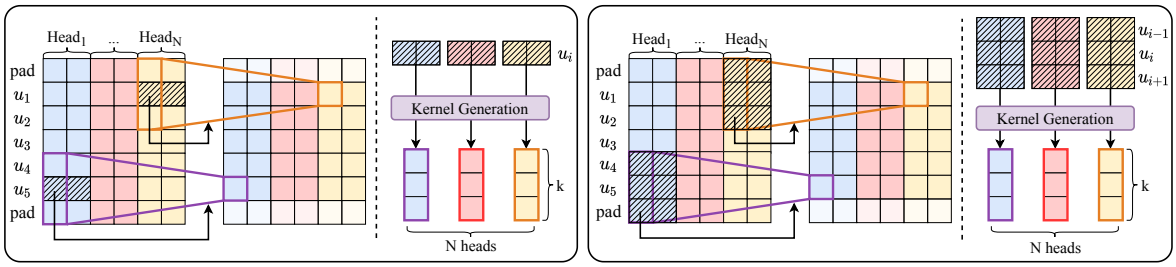
Prior works model the joint DAR and SC as sequence labeling problem, all accomplishing with recurrent-like neural models, e.g., Long Short-Term Memory Network (LSTM) (Chen et al., 2018; Raheja and Tetreault, 2019). However, one crucial drawback in these models is failing to fully incorporate

[1]Codes are publicly available at https://github.com/ljynlp/BCDCN.

[†]Equally Contributed.

[‡]Corresponding author.

Proceedings of the 28th International Conference on Computational Linguistics, pages 616–626
Barcelona, Spain (Online), December 8-13, 2020



(a) Dynamic convolution.

(b) Context-aware dynamic convolution.

Figure 1: The shadow boxes are the source context for generating the corresponding kernels. Dynamic convolution (a) of each head at time-step i generates a kernel with only current i-th input, while our proposed context-aware dynamic convolution (b) computes kernels within a wide context window.

the local contextual information among the dialog. Intuitively, for current utterance, its nearer dialogue neighbors are always more influential and informative than the remoter ones, due to the closer replying relationships between them. This can be exemplified by the two cases we mentioned above, where the inter-impacts between dialogue acts and sentiments more often occur within adjacent utterances. Therefore, in this paper we target fully capturing the local contexts for improving the joint tasks.

Convolutional Neural Networks (CNN) are the preferred alternatives on effectively extracting local-feature in discourses in natural language processing (NLP) community (Kalchbrenner et al., 2014; Kim, 2014). The recent advanced CNN variants, dynamic convolutions (Wu et al., 2019), have been proposed for bringing enhanced capabilities. As illustrated in Figure 1(a), it operates by multiple heads of convolutions with the shared dynamic kernel over dimension, which enables to dynamically generate different convolution kernels for different input elements at each time-step. Compared with vanilla CNN, the dynamic convolution network is much more flexible on mining the contextual features, and meanwhile effectively reduces model parameters (Kaiser et al., 2018). In this work, we consider taking advantage of the dynamic convolution network as the utterance context extractor for the joint tasks. However, we can notice that the dynamic convolutions generate kernels merely under current input utterance while ignoring the surroundings of the utterance, which may be problematic when encoding the utterance feature representation. Therefore, we further adapt the dynamic convolutions by designing a *context-aware dynamic convolution*, as in Figure 1(b). Our context-aware dynamic convolutions dynamically calculate kernels under the guidance of wider utterance contexts, which allows to better filter or integrate dialogue act and sentiment from various heads, and yield more informative contextual representation in dialogue.

We take the dynamic convolution network (DCN) and our context-aware dynamic convolution network (CDCN) as the utterance context encoders, respectively. As shown in Figure 2(a), to satisfy the joint DAR and SC tasks, we extend the models into bi-channel version (i.e., BDCN or BCDCN) under a multi-task learning framework, with each channel for each subtask. Two channels can separately learn their own feature representations but with latent interactions. First, the bi-directional LSTM (BiLSTM) layer encodes the input utterance into representations. Then, in utterance encoder layer, multi-layer BDCN (or BCDCN) captures the context representations for DAR and SC, respectively. Finally, after linear transformation, our model makes final predictions for two tasks separately. In addition, we consider exploiting the BERT pre-trained contextualized representations, which have been demonstrated to bring nearly 10% improvements for the joint DAR and SC tasks on Mastodon data (Qin et al., 2020). However, one entire dialog often consists of far more than two utterance sentences, while the BERT restricts the input with at maximum two sentence pieces, which consequently limits the utility. We thus adopt the DiaBERT (Liu and Lapata, 2019) to yield the enhanced utterance representations at dialogue level.

We experiment on two benchmarks, including Mastodon (Cerisara et al., 2018) and DailyDialog (Li et al., 2017). The results show that both our BDCN and BCDCN systems can beat the current best baseline with large margins for the joint DAR and SC tasks. Especially our BCDCN model achieves the state-of-the-art performances, demonstrating its advances. With the DiaBERT, the performances can be significantly boosted when data is sufficient (e.g., in DailyDialog). Further analysis shows the necessity

of capturing the local contexts for the joint tasks. In summary, we make following contributions. **1)** We are the first proposing to improve the joint dialog act recognition and sentiment classification by fully capturing the utterance local contexts. **2)** We employ the dynamic convolutions network for better encoding local contexts of utterances, based on which we further propose a novel context-aware dynamic convolutions network for enhancement. **3)** Our proposed frameworks achieve the state-of-the-art performances against the current best baselines on two tasks in two datasets. **4)** We obtain significantly reinforced results by employing the DiaBERT and BERT language models.

2 Related Work

Dialogue Act Recognition (DAR) plays an important role in building intelligent and interactive conversation systems and producing appropriate responses. Existing researches for DAR can be summarized into two categories. Initial works make a prediction for each dialogue utterance independently by traditional statistical classifiers with hand-crafted discrete features (Stolcke et al., 2000; Keizer et al., 2002; Surendran and Levow, 2006; Lendvai and Geertzen, 2007; Tavafi et al., 2013). Later, researchers treat DAR as a sequence labeling problem and apply neural network methods to reach significant improvements (Kalchbrenner and Blunsom, 2013; Chen et al., 2018; Kumar et al., 2018; Raheja and Tetreault, 2019; Colombo et al., 2020). On the other hand, capturing the sentiment polarity or opinions according to the texts, aka. sentiment classification has long been a heated research topic in NLP community (Ren et al., 2016; Amplayo et al., 2018; Fei et al., 2019; Fei et al., 2020b). It has been revealed recently that the dialogue act recognition and sentiment classification can be closely related, and the joint learning of these two tasks is more beneficial (Cerisara et al., 2018; Kim and Kim, 2018; Qin et al., 2020). In this work, we follow these works by employing the joint scheme of the two tasks. Differently, we aim to better capture the local contexts of dialogue utterances for task improvements.

This work also relates to the CNN models. CNN is prominent on retrieving the local features among texts or discourses (Kim, 2014; Zhang et al., 2015; Lei et al., 2015), functioning by capturing the n-gram patterns within input contents (Kim, 2014). A large body of studies considers retrofit CNN to yield better task performances. For example, Kaiser et al. (2018) apply depth-wise separable convolutions to neural machine translation, which perform convolutions independently over every channel dimension. Based on the work of Kaiser et al. (2018), Wu et al. (2019) introduce dynamic convolutions. They define a number of convolution heads where each head shares kernel over every dimension computed at each time step. In this work, we take advances of the CNN-like models for sufficiently mining the local context information in dialogue utterances. Based on the dynamic convolutions, we newly propose context-aware dynamic convolutions, which is expected to give the enhanced capability of local feature extraction for our task.

3 Framework

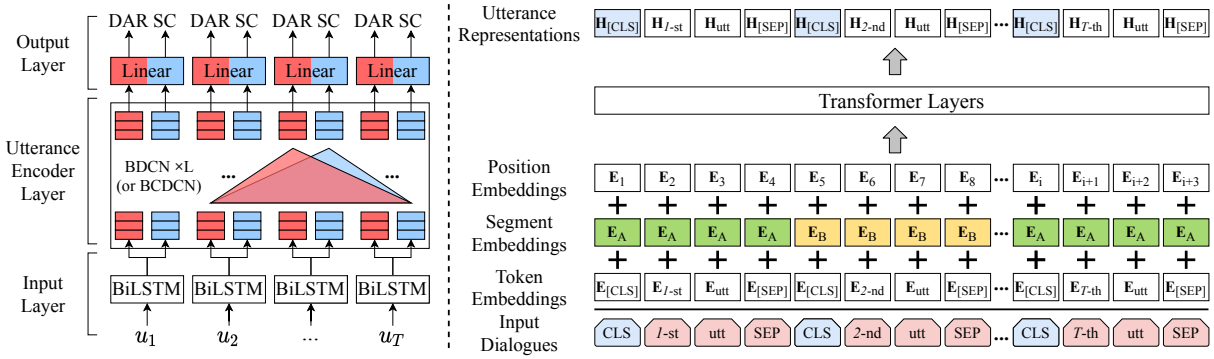
We cast both the DAR and SC as a sequence labeling problem. Given a dialogue C with a sequence of utterances $U = \{u_1, u_2, \ldots, u_T\}$, where T is the length of the sequence, the objective of DAR is to predict the corresponding dialogue act labels $Y^d = \{y_1^d, y_2^d, \ldots, y_T^d\}$. In SC, the goal is to determine the utterance sequence to the corresponding sentiment labels $Y^s = \{y_1^s, y_2^s, \ldots, y_T^s\}$.

Our framework is built based on either the dynamic convolution network (DCN) or context-aware dynamic convolution network (CDCN). We achieve the joint tasks of DAR and SC by extending these networks into bi-channel version (i.e., BDCN or BCDCN), where each subtask takes one channel but with latent interactions. These two tasks can be jointly trained under the multi-task learning framework.

As illustrated in Figure 2(a), the overall architecture of our framework consists of three tiers. At the input layer, the BiLSTM encodes the input utterance texts into representations. Then at the utterance encoder layer, the stacked multi-layer BDCN (or BCDCN) captures the context representations for DAR and SC, respectively. Finally, at the output layer, the model predicts labels for two tasks separately.

3.1 Input Layer

We denote the word sequence in t-th utterance u_t as $X_t = \{w_{t,1}, w_{t,2}, \ldots, w_{t,r}\}$, where r is the length of the sequence. We first map the surface words X_t into vectorial representation $\boldsymbol{X}_t$ via a look-up table.



(a) Joint model for DAR and SC. (b) The structure of DiaBERT language model

Figure 2: Illustrations of our proposed framework architecture (a) and the DiaBERT language model (b).

We then adopt BiLSTM to encode the embeddings and obtain the input utterance representations:

$$\mathbf{h}_t = \text{BiLSTM}(\mathbf{X}_t)\ , \tag{1}$$

where $\mathbf{h}_t$ is the desired input utterance representations for u_t. The order information of the utterance sequence is quite crucial to our task. We thus consider adding the sinusoidal position embedding (Vaswani et al., 2017) to encode the absolute position of each utterance. We concatenate the input utterance representation with this position embedding for each utterance.

$$\mathbf{h}_t := \mathbf{h}_t + \text{PE}(t)\ , \tag{2}$$

where $\text{PE}(\cdot)$ denotes the desired representation from positional encoding function.

3.2 Utterance Encoder Layer

As we described earlier that the nearer neighbors of an utterance u_t can be more informative than farther ones, we consider fully modeling the local contexts via convolution networks. We use either the dynamic convolution network (DCN) or context-aware dynamic convolution network (CDCN) as our utterance encoder, respectively. In what follows we first elaborate the technical details of DCN and CDCN separately. Then we demonstrate how to integrate DAR and SC jointly with two channels.

Dynamic convolution network. Compared with vanilla text convolutions (Kim, 2014), dynamic convolutions can dynamically generate varying convolution kernels for different input elements at each time-step, being more flexible on mining the contextual features. Dynamic convolution is based on the depth-wise separable convolution (Kaiser et al., 2018), where a convolution is performed independently over each input dimension. Based on the t-th utterance, the input of the convolution with the k window width (i.e., kernel size of $k = 2k' + 1$) is denoted as $\mathbf{C}_t = [\mathbf{h}_{t-k'}, \cdots, \mathbf{h}_{t+k'}]$. Then the depth-wise separable convolution (*SConv*) operates as:

$$\boldsymbol{o}_t = \text{SConv}(\mathbf{C}_t|\boldsymbol{W}_c) = \textstyle\sum_{i=1}^{d} \boldsymbol{W}_{c,i}\mathbf{c}_{t,i}\ , \tag{3}$$

where $\mathbf{W}_c$ is the kernel (i.e., convolution weight), $\mathbf{o}_t$ is the output representation from the depth-wise separable convolution, d is the dimension of the hidden state, $\boldsymbol{c}_{t,i}$ is the dimension i of $\boldsymbol{C}_t$.

In dynamic convolutions (*DConv*), each head of the convolutions is based on the depth-wise separable convolution, while tying the convolution weights over the dimension of multiple heads (i.e., N heads), and learning the weights dynamically over time. This mechanism extends a similar spirit to the recent multi-head self-attention (Vaswani et al., 2017), where multiple heads can jointly capture latent features from different representation subspaces. The calculation in dynamic convolutions can be formulated as:

$$\text{DConv}(\mathbf{C}_t|\boldsymbol{W}_{\text{DConv}}) = (\text{head}_1 \oplus \cdots \oplus \text{head}_N)\boldsymbol{W}^O\ , \tag{4}$$

$$where\ \text{head}_n = \text{SConv}(\mathbf{C}_t|\boldsymbol{W}_{\text{DConv}})\ , \tag{5}$$

$$\boldsymbol{W}_{\text{DConv}} = \text{KerGen}(\boldsymbol{h}_t) \equiv \text{softmax}(\boldsymbol{W}_n\mathbf{h}_t)\ , \tag{6}$$

where $\oplus$ denotes the concatenation operation, $\boldsymbol{W}_n$ is learnable parameters. $\boldsymbol{W}_{\text{DConv}}$ is the kernel in n-th convolution head dynamically generated by the kernel generation function $\text{KerGen}(\cdot)$ in Eq. 6.

Context-aware dynamic convolution network. Instead of using the current t-th utterance representation $\mathbf{h}_t$ for producing the kernels as in dynamic convolution, we propose the context-aware dynamic convolution (*CDConv*) to generate kernels (i.e., in Eq 6) with wider contexts.

$$\text{CDConv}(\mathbf{C}_t|\boldsymbol{W}_{\text{CDConv}}) = \text{DConv}(\boldsymbol{C}_t|\boldsymbol{W}_{\text{CDConv}})\,, \tag{7}$$

$$where\ \boldsymbol{W}_{\text{CDConv}} = \text{KerGen}(\boldsymbol{V}_t) \equiv \text{softmax}(\boldsymbol{W}_n\mathbf{C}_t) \tag{8}$$

$\boldsymbol{V}_t$=$[\mathbf{h}_{t-m'}, \cdots, \mathbf{h}_{t+m'}]$ (m=$2m'$+1 is window width) is the kernel-generation context for t-th utterance.

We can abstract all the above convolution operations, and encapsulate them into a layer:

$$\hat{\boldsymbol{h}}_t = \text{ConvLayer}(\boldsymbol{h}_t)\,, \tag{9}$$

where $\hat{\boldsymbol{h}}_t$ is the corresponding output representation for t-th utterance. We can take either *DConv* or *CDConv* as the underlying convolutions in ConvLayer, referred to as DCN or CDCN.

Multi-layer bi-channel convolution network for joint DAR and SC. In our practice, we first stack the DCN or CDCN with multiple layers, so as to expand the receptive field for fully extracting the context features among the dialogue level. For l-th layer, the t-th utterance representation $\mathbf{h}_t^{(l)}$ is updated as:

$$\boldsymbol{h}_t^{(l)} = \text{ReLU}(\text{ConvLayer}^{(l)}(\text{LayerNorm}(\boldsymbol{h}_t^{(l-1)})))\,. \tag{10}$$

where $\text{LayerNorm}(\cdot)$ is a layer normalization operation (Ba et al., 2016).

Next, inspired by the multi-channel CNN (Kim, 2014), we use two same separate convolution networks as two channels to model the DAR and SC parallel (namely, BDCN and BCDCN), as shown in Figure 2(a). Two channels perform learning separately but with latent interactions, such that the two tasks are expected to propagate information from different aspects, and aggregate them with dynamic convolution operations. To reach this, we let the input representation of each channel at l-th layer as the concatenation of the output representations of the both two channels at the last layer.

$$\mathbf{h}_t^{(l),*} = \text{ReLU}(\text{ConvLayer}(\text{LayerNorm}(\mathbf{h}_t^{(l-1),d} \oplus \mathbf{h}_t^{(l-1),s})))\,, \tag{11}$$

where the superscript $^* \in \{d, s\}$ presents either the channel for DAR (d) or for SC (s), alternatively.

3.3 Output Layer

At the last layer (i.e., L-th layer), we separately apply two linear layers concatenated with softmax functions to calculate the output probability for the two tasks, respectively:

$$\mathbf{y}_t^* = \text{Softmax}(\text{Linear}(\mathbf{h}_t^{L,*}))\,. \tag{12}$$

3.4 Learning

We optimize the model by minimizing the negative cross-entropy between the predictions $\mathbf{y}^*$ and the gold labels $\hat{\mathbf{y}}^*$ for both DAR and SC:

$$\mathcal{L}_* = -\frac{1}{T}\sum_{t=1}^{T}\hat{\mathbf{y}}_t^* \log \mathbf{y}_t^*\,. \tag{13}$$

We perform joint learning of two tasks by combing the above loss for them (i.e., $\mathcal{L}_d$ and $\mathcal{L}_s$). Specifically, instead of using constant coupling co-efficiency, we employ the homoscedastic uncertainty strategy(Kendall et al., 2018) for learning weights automatically during the training:

$$\mathcal{L} = \frac{1}{2\sigma_d^2}\mathcal{L}_d + \frac{1}{2\sigma_s^2}\mathcal{L}_s + \log \sigma_d\sigma_s\,, \tag{14}$$

where σ_d and σ_s are the variances of DAR loss and SC loss over training instances, respectively.

4 DiaBERT: Pre-trained Contextualized Utterance Representation

In §3.1 we use a look-up table for getting the initial word embedding. Pre-trained contextualized word representations from BERT language model (Devlin et al., 2019) have brought great benefits to a wide range of downstream NLP tasks (Jia et al., 2019; Fei et al., 2020a). The very recent work (Qin et al., 2020) demonstrates that using BERT brings large-scale task improvements for DAR and SC. In this work we intend to borrow these advances from BERT as well. Nevertheless, original BERT limits the input

Dataset	#Dialogues			#Utterances			#Labels	
	Train	Dev	Test	Train	Dev	Test	SC	DAR
Dailydialog	11,118	1,000	1,000	87,170	8,069	7,740	7	4
Mastodon	240	25	266	979	84	1,142	3	15

Table 2: Statistics of the two datasets.

with maximum two sentence pieces, while often one dialog can comprise far more than two utterance sentences. Directly using BERT can lead to discourse information incoherence.

To this end, we leverage a dialogue-level (discourse-level) BERT-based encoder *DiaBERT* (Liu and Lapata, 2019) to take the whole dialogue and output the complete representation. In BERT there can be only one [CLS] token for splitting at most a pair of utterance sentences, while DiaBERT treats each utterance as a segment, each with a [CLS] token at the start of the utterance for distinguishing different utterance, as illustrated in Figure 2(b). DiaBERT entails token-level and dialog-level output features.

$$\{\mathbf{h}_1^{DB}, \cdots, \mathbf{h}_T^{DB}\} = \text{DiaBERT}(X_1, \cdots, X_T)\ , \tag{15}$$

where $\mathbf{h}^{DB}$ denotes the output for input utterance X. So we can either make use of the word representation from DiaBERT, or take the one from each [CLS] token as the corresponding utterance representation.

5 Experimental Setting

5.1 Dataset and Evaluation

We evaluate our methods on two benchmark datasets, Mastodon[2] and DailyDialog[3]. Table 2 shows the detailed statistics. Specifically, since no developing set is in Mastodon, we randomly split 10% of the training set dialogues. Following Qin et al. (2020), we use the macro-averaged Precision (P), Recall (R) and F1 as the major metrics measuring two tasks on DailyDialog. On Mastodon, following Cerisara et al. (2018), we ignore the neutral sentiment label in SC, and for DAR we adopt the average of the F1 scores weighted by the prevalence of each dialogue act. For all experiments, we pick the model that performs best on developing set, and all the results on the test set are presented on average after 20 times running.

5.2 Hyper-parameter and Resource

The BDCN and BCDCN are set to 3 layers for best performances according to preliminary experiments. We adopt Adam as the optimizer with an initial learning rate of 5e-4 with weight decay of 1e-5. The mini-batch size is 16. To alleviate overfitting, we use a dropout rate of 0.5 on the input layer and the output layer. For fair comparisons, we randomly initialize word embedding without pre-training, following Qin et al. (2020), and the dimension is selected in [100,200,300,400]. Our DiaBERT shares the same architecture with the official BERT[4] (Base version) and is pre-trained in the same way.

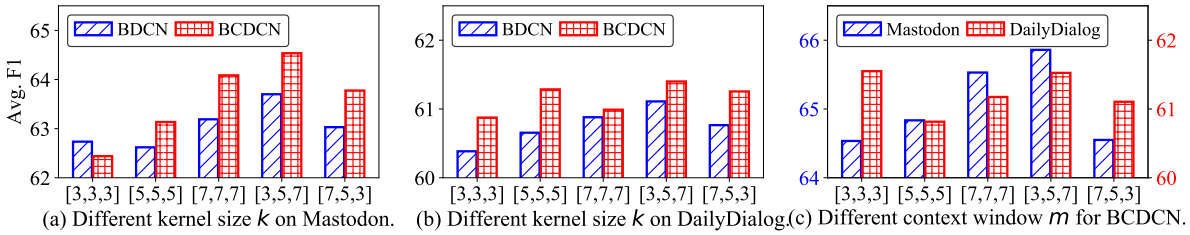
5.3 Baseline System

We divide the state-of-the-art baseline models into three categories. **1)** The models for DAR, including HEC (Kumar et al., 2018), CRF-ASN (Chen et al., 2018) and CASA (Raheja and Tetreault, 2019). **2)** The models for SC, including VDCNN (Conneau et al., 2017), Region.emb (Qiao et al., 2018) and DialogueRNN (Majumder et al., 2019). **3)** the joint learning models for DAR and SC, including JointDAS (Cerisara et al., 2018), IIIM (Kim and Kim, 2018) and DCR-Net (Qin et al., 2020). Also, we run our DCN and CDCN for two single separate tasks.

5.4 Developing Experiment

We first conduct developing experiments for our BDCN and BCDCN encoders based on the developing sets of two datasets, to study the best hyper-parameters of k and m. Figure 3 plots the varying results.

[2] https://github.com/cerisara/DialogSentimentMastodon
[3] http://yanran.li/dailydialog
[4] https://github.com/google-research/bert



(a) Different kernel size k on Mastodon. (b) Different kernel size k on DailyDialog. (c) Different context window m for BCDCN.

Figure 3: Developing experiments for BDCN and BCDCN encoders. 'Avg. F1' is the averaged F1 score for DAR and SC tasks. Each x values (e.g., [3,3,5]) refers to the k or m values at each of three layers.

	Mastodon							DailyDialog						
	SC			DAR			Avg.	SC			DAR			Avg.
	F1	P	R	F1	P	R		F1	P	R	F1	P	R	
• **Separate learning of two tasks**														
HEC	-	-	-	56.1	56.5	55.7	-	-	-	-	77.8	77.8	76.5	-
CRF-ASN	-	-	-	55.1	56.5	53.9	-	-	-	-	76.0	78.2	75.6	-
CASA	-	-	-	56.4	55.7	57.1	-	-	-	-	78.0	77.9	76.5	-
VDCNN	39.6	**44.0**	31.6	-	-	-	-	39.7	55.2	35.6	-	-	-	-
Region.emb	40.3	42.8	33.6	-	-	-	-	41.0	56.4	36.6	-	-	-	-
DialogueRNN	41.5	40.5	42.8	-	-	-	-	40.3	44.5	37.7	-	-	-	-
BERT	57.4	45.4	77.9	66.3	65.3	67.4	61.9	50.8	56.0	47.9	80.9	81.4	80.5	65.9
DCN	44.1	36.0	60.2	58.0	54.7	**61.7**	51.1	47.4	52.3	44.9	79.6	79.1	80.2	63.5
CDCN	**44.7**	36.2	**60.7**	**58.7**	**56.4**	61.2	**51.7**	**47.8**	**55.0**	**45.5**	**79.9**	**79.5**	**80.4**	**63.9**
+BERT	59.2	47.2	79.9	68.5	67.8	69.2	63.9	52.3	56.8	49.6	81.7	82.1	81.0	67.0
+DiaBERT	58.0	45.8	78.9	67.3	66.7	68.0	62.7	53.0	58.3	51.8	82.1	81.0	82.5	67.6
• **Joint learning of two tasks**														
JointDAS	37.6	36.1	41.6	53.2	55.6	51.9	45.4	31.2	35.4	28.8	75.1	76.2	76.2	53.2
IIIM	39.4	38.7	40.1	54.3	56.3	52.2	46.9	33.0	38.9	28.5	75.7	76.5	76.5	54.4
DCR-Net	45.1	**43.2**	47.3	58.6	**60.3**	56.9	51.9	45.4	**56.0**	40.1	79.1	79.1	79.0	62.3
+BERT	55.1	56.5	56.5	67.1	69.2	65.2	61.1	48.9	63.2	46.9	80.0	80.2	79.9	64.5
BDCN	45.6	37.4	61.8	59.0	57.1	61.0	52.3	48.1	55.2	43.9	80.0	79.3	**80.8**	64.1
BCDCN	**45.9**	38.2	**62.0**	**59.4**	57.3	**61.7**	**52.7**	**48.6**	55.2	**45.7**	**80.3**	**80.0**	80.6	**64.5**
+BERT	60.3	47.7	82.2	69.9	68.1	71.7	65.1	52.9	57.3	51.1	82.0	82.2	82.0	67.5
+DiaBERT	59.5	47.5	81.4	68.3	67.4	69.2	63.9	54.4	58.8	53.7	82.6	82.0	83.4	68.5

Table 3: Main performances. `Avg.` represents the averaged F1 score over SC and DAR on the dataset. All the scores of baseline models are reprinted from Qin et al. (2020) without any modification.

BDCN and BCDCN perform the best when kernel sizes k are both set as [3,5,7] on both Mastodon and DailyDialog datasets. Meanwhile, keeping the best k as [3,5,7] for BCDCN, the utilities are the best when the context window m of kernel generation is set as [3,5,7] on Mastodon, and [3,3,3] on DailyDialog. We keep these configurations for BDCN and BCDCN for the following experiments.

6 Result and Analysis

6.1 Main Result

We compare the performances by all systems under 1) the separate learning of two tasks of SC and DAR, and 2) the joint learning of two tasks, respectively. In Table 3 we show the main results on Mastodon and DailyDialog datasets, respectively, from which we can have the following observations.

First of all, on both two datasets, the models under joint task learning universally outperform these under separate learning of two single tasks, which can be informed by the comparisons between the results from DCN/CDCN and those from BDCN/BCDCN. This verifies the necessity of conducting joint training of DAR and SC. Among the models by separate task learning, our DCN and CDCN perform much better than baselines, demonstrating the effectiveness of capturing the local contexts for the tasks.

Model	Param.	SC	DAR
BCDCN	9.42M	48.6	80.3
w/o position embedding	9.42M	47.3	80.1
w/o bi-channerl interaction (in Eq. 11)	9.12M	47.5	79.6
w/o dynamic weight	9.03M	47.4	79.7
Vanilla convolution (CNN, k=3)	17.66M	45.5	78.8
Vanilla convolution (CNN, k=3,5,7)	26.30M	46.2	79.2
Depth-wise convolution (SConv)	9.06M	47.2	79.6
Dynamic convolution (DConv)	9.24M	48.1	80.0
Fixed loss co-efficiency ($0.5\mathcal{L}_d + 0.5\mathcal{L}_s$)	9.42M	47.6	79.9

Table 4: Ablation study on BCDCN model. Values for SC and DAR are F1 scores. 'w/o' means 'without'.

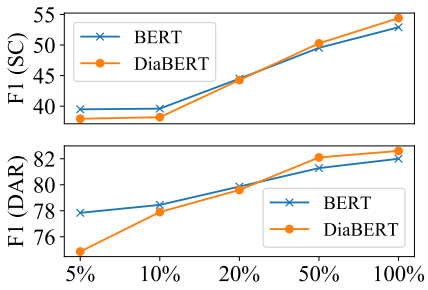


Figure 4: F1 scores against varying data scale for fine-tuning BERT and DiaBERT.

Second, among joint learning of two tasks, our BDCN consistently shows better results than the current state-of-the-art model (i.e., DCR-Net). Especially our BCDCN achieves the best F1 scores, with an average 52.7% on Mastodon and 64.5% on DailyDialog. This again proves the advances in the local feature extraction for DAR and SC. Besides, we find that improvements by our BCDCN (than best baseline) are overall higher on DailyDialog (2.2%=64.5-62.3) than on Mastodon (0.8%=52.7-51.9).

Third, we see that with the help of pre-trained contextual language models (i.e., BERT and DiaBERT), the performances on both the separate task learning and the joint tasks can be greatly boosted. Notably, the enhancements are much significant on Mastodon than on DailyDialog. For example, with BERT, our BCDCN gives increase by 12.4%(65.1-52.7) F1 on Mastodon, and 3.4%(67.5-64.1) on DailyDialog. It is largely due to that the small size of the Mastodon data can give limited supervision signals for training, and in this case the external information can be greatly beneficial. Also, with language models, BCDCN even gains much higher F1 scores than DCR-Net. Last but not least, similar to BERT, DiaBERT can bring great improvements for the tasks, where however the help for DailyDialog are more significant than for Mastodon. This largely lies in the differences of the data scale provided for fine-tuning.

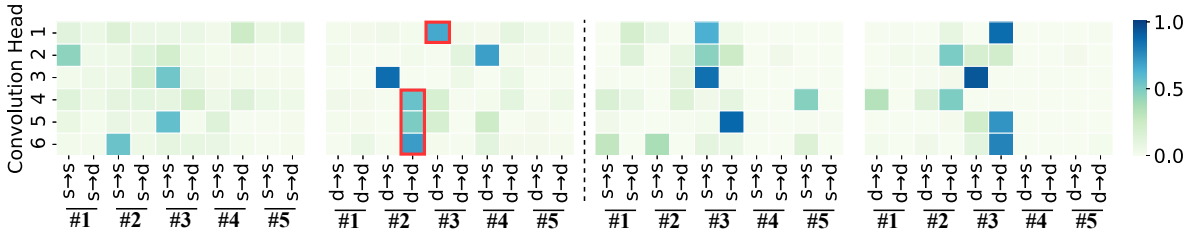
6.2 Ablation Study

We now take a further step, studying our model under various configurations. We conduct ablation experiments for the BCDCN model based on DailyDialog data, as shown in Table 4. First, without the position embeddings at the input layer, we see the slight performance drops. Next, we focus on the convolution part of our model. Further, we cancel the latent interactions between DAR and SC by taking $\mathbf{h}_t^{(l-1),*}$ instead of $\mathbf{h}_t^{(l-1),d} \oplus \mathbf{h}_t^{(l-1),s}$ (in Eq. 11) as $*$ channel's input in l-th layer, separately. Removing the dynamic kernel of BCDCN model also results in minor drops. When replacing the context-aware dynamic convolution with 1) vanilla text convolution, 2) depth-wise convolution and 3) dynamic convolution, respectively, we can observe varying degrees of performance decreases. In particular, by employing advanced convolution rather than the vanilla text convolution, the number of model parameters can be greatly reduced. Significant reductions for both two tasks can consequently be witnessed. Finally, using the fixed coupling co-efficiency for the multi-task training loss (i.e., $0.5\mathcal{L}_d + 0.5\mathcal{L}_s$), rather than the adopted homoscedastic uncertainty strategy, can result in suboptimal learning performances.

6.3 The Impact of Data Scale for Fine-tuning Language Model

In Table 3 we notice that the DiaBERT can outperform BERT on the tasks only where comparatively larger data for fine-tuning is offered (i.e., DailyDialog). Otherwise, the improvements become inferior to that by BERT (i.e., Mastodon). Here we explore the influence that the DiaBERT is subject to data scarcity. Based on DailyDialog data, we vary the training set for fine-tuning DiaBERT and BERT, and test the F1 scores for DAR and SC by BCDCN. From the trends in Figure 4, we see that the performances with BERT are better than that with DiaBERT when the used training data are less than 20%. In addition, DiaBERT gains more improvements than BERT when training signals are abundant.

Index	Speaker	Utterance	Dialogue Act	Sentiment
#1	A	There's no way to make a post visible to just your local TL and not federated TL	*Statement*	*Negative*
#2	B	correct ?	*Question*	*Negative*
#3	A	I don't think there is	*Answer*	*Negative*
#4	B	thanks	*Thanking*	*Positive*
#5	A	didn't think so.	*Agreement*	*Negative*



(a) Kernel weights in SC (left) and DAR (right) channel in BDCN. (b) Kernel weights in SC (left) and DAR (right) channel in BCDCN.

Figure 5: Visualization of kernel weights in different channels and heads (bottom heatmaps), based on the dialog (upper table). The Current input for models is the 3-rd utterance. Each x label (e.g., s→d) indicates the attentive interaction (by Eq. 6) of one channel (e.g., SC) to another channel (e.g., DAR).

6.4 Visualization on Bi-channel Dynamic Convolution Mechanism

Finally, we empirically visualize the convolution weights to better understand the mechanism of the bi-channel dynamic convolution in BDCN and BCDCN encoders. Specifically, we illustrate the dynamic kernel weights with 3-rd utterance as the current time step, under the attentive interaction (by Eq. 6) of one channel to another channel, as in Figure 5. Firstly, we can learn that the dynamic kernels can help to retrieve informative clues for the corresponding tasks, which is inferred by those highlighted weights in different heads and channels of convolutions in both two encoders. Each channel actively pays more attention to its own responsibility, respectively.

More importantly, the dynamic convolutions in different channels can work collaboratively with another channel. For example, the highly-weighted kernel of the heads for utterance in the DAR channel, as highlighted by the red box in Figure 5(a) (right), where the kernel shows attention to both SC and DAR channels (i.e., d→s), with the previous utterance's dialogue act *Question* and the current sentiment *Negative*, the model is consequently informed to transitioning the dialogue act from *Question* to *Answer*. This indicates that our proposed interactive bi-channel mechanism is effective for coordinating the joint tasks with latent interaction. Besides, we see that there are more activated highlighted-weights in BCDCN than in BDCN. With wider local contexts modeled by dynamic kernel generation mechanism, BCDCN can be more prominent on mining informative clues for the joint DAR and SC tasks.

7 Conclusion

In this work, we proposed to improve the joint dialogue act recognition (DAR) and sentiment classification (SC) by fully modeling the local contexts of utterances. Based on the dynamic convolution network, we proposed a context-aware dynamic convolution network for better leveraging the local dialogue contexts when generating convolution kernels. We extended our frameworks into bi-channel version under multi-task learning for the joint DAR and SC. Our models obtained state-of-the-art performances on two benchmark datasets, demonstrating the significance of modeling the local contexts for the joint task.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61772378), 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

Reinald Kim Amplayo, Jihyeok Kim, Sua Sung, and Seung-won Hwang. 2018. Cold-start aware user and product attention for sentiment classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2535–2544.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Christophe Cerisara, Somayeh Jafaritazehjani, Adedayo Oluokun, and Hoa Le. 2018. Multi-task dialog act and sentiment recognition on mastodon. In *Proceedings of COLING*, pages 745–754.

Zheqian Chen, Rongqin Yang, Zhou Zhao, Deng Cai, and Xiaofei He. 2018. Dialogue act recognition via crf-attentive structured network. In *Proceedings of SIGIR*, pages 225–234.

Pierre Colombo, Emile Chapuis, Matteo Manica, Emmanuel Vignon, Giovanna Varni, and Chloe Clavel. 2020. Guiding attention in sequence-to-sequence models for dialogue act prediction. In *Proceedings of AAAI*, pages 7594–7601.

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2017. Very deep convolutional networks for text classification. In *Proceedings of EACL*, pages 1107–1116.

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.

Hao Fei, Yafeng Ren, and Donghong Ji. 2019. Implicit objective network for emotion detection. In *Proceedings of NLPCC*, pages 647–659.

Hao Fei, Meishan Zhang, and Donghong Ji. 2020a. Cross-lingual semantic role labeling with high-quality translated training corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7014–7026.

Hao Fei, Yue Zhang, Yafeng Ren, and Donghong Ji. 2020b. Latent emotion memory for multi-label emotion classification. In *Proceedings of AAAI*, pages 7692–7699.

Nobuo Inui, Toshiaki Ebe, Bipin Indurkhya, and Yoshiyuki Kotani. 2001. A case-based natural language dialogue system using dialogue act. In *Proceedings of SMC*, pages 193–198.

Chen Jia, Xiaobo Liang, and Yue Zhang. 2019. Cross-domain NER using cross-domain language modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2464–2474.

Lukasz Kaiser, Aidan N Gomez, and Francois Chollet. 2018. Depthwise separable convolutions for neural machine translation. In *Proceedings of ICLR*.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent convolutional neural networks for discourse compositionality. *arXiv preprint arXiv:1306.3584*.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of ACL*, pages 655–665.

Simon Keizer, Rieks op den Akker, and Anton Nijholt. 2002. Dialogue act recognition with bayesian networks for dutch dialogues. In *Proceedings of SIGDIAL Workshop on Discourse and Dialogue*, pages 88–94.

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.

Minkyoung Kim and Harksoo Kim. 2018. Integrated neural network model for identifying speech acts, predicators, and sentiments of dialogue utterances. *Pattern Recognition Letters.*, 101:1–5.

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

Harshit Kumar, Arvind Agarwal, Riddhiman Dasgupta, and Sachindra Joshi. 2018. Dialogue act sequence labeling using hierarchical encoder with CRF. In *Proceedings of AAAI*, pages 3440–3447.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2015. Molding cnns for text: non-linear, non-consecutive convolutions. In *Proceedings of EMNLP*, pages 1565–1575.

Piroska Lendvai and Jeroen Geertzen. 2007. Token-based chunking of turn-internal dialogue act sequences. In *Proceedings of SIGDIAL*, pages 174–181.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of IJCNLP*, pages 986–995.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of EMNLP-IJCNLP*, pages 3728–3738.

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.

Chao Qiao, Bo Huang, Guocheng Niu, Daren Li, Daxiang Dong, Wei He, Dianhai Yu, and Hua Wu. 2018. A new method of region embedding for text classification. In *Proceedings of ICLR*.

Libo Qin, Wanxiang Che, Yangming Li, Minheng Ni, and Ting Liu. 2020. Dcr-net: A deep co-interactive relation network for joint dialog act recognition and sentiment classification. In *Proceedings of AAAI*, pages 8665–8672.

Vipul Raheja and Joel Tetreault. 2019. Dialogue act classification with context-aware self-attention. In *Proceedings of NAACL-HLT*, pages 3727–3733.

Yafeng Ren, Yue Zhang, Meishan Zhang, and Donghong Ji. 2016. Improving twitter sentiment classification using topic-enriched multi-prototype word embeddings. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 3038–3044.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3):339–373.

Dinoj Surendran and Gina-Anne Levow. 2006. Dialog act tagging with support vector machines and hidden markov models. In *Proceedings of InterSpeech*.

Maryam Tavafi, Yashar Mehdad, Shafiq Joty, Giuseppe Carenini, and Raymond Ng. 2013. Dialogue act recognition in synchronous and asynchronous conversations. In *Proceedings of SIGDIAL*, pages 117–121.

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.

Felix Wu, Angela Fan, Alexei Baevski, Yann N Dauphin, and Michael Auli. 2019. Pay less attention with lightweight and dynamic convolutions. In *Proceedings of ICLR*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of NeurIPS*, pages 649–657.