Local Contextual Attention with Hierarchical Structure for Dialogue Act Recognition

Zhigang Dai[1], Jinhua Fu[2], Qile Zhu[3], Hengbin Cui[2], Xiaolong Li[2] and Yuan Qi[2]
[1]South China University of Technology
[2] Ant Financial Service Group

Abstract

Dialogue act recognition is a fundamental task for an intelligent dialogue system. Previous work models the whole dialog to predict dialog acts, which may bring the noise from unrelated sentences. In this work, we design a hierarchical model based on self-attention to capture intra-sentence and inter-sentence information. We revise the attention distribution to focus on the local and contextual semantic information by incorporating the relative position information between utterances. Based on the found that the length of dialog affects the performance, we introduce a new dialog segmentation mechanism to analyze the effect of dialog length and context padding length under online and offline settings. The experiment shows that our method achieves promising performance on two datasets: Switchboard Dialogue Act and DailyDialog with the accuracy of 80.34% and 85.81% respectively. Visualization of the attention weights shows that our method can learn the context dependency between utterances explicitly.

1 Introduction

Dialogue act (DA) characterizes the type of a speaker's intention in the course of producing an utterance and is approximately equivalent to the illocutionary act of Austin (1962) or the speech act of Searle (1969). The recognition of DA is essential for modeling and automatically detecting discourse structure, especially in developing a human-machine dialogue system. It is natural to predict the *Answer* acts following an utterance of type *Question*, and then match the *Question* utterance to each QA-pair in the knowledge base. The predicted DA can also guide the response generation process (Zhao et al., 2017). For instance, system generates a *Greeting* type response to former *Greeting* type utterance. Moreover, DA is beneficial to other online dialogue strategies, such as conflict avoidance (Nakanishi et al., 2018). In the offline system, DA also plays a significant role in summarizing and analyzing the collected utterances. For instance, recognizing DAs of a wholly online service record between customer and agent is beneficial to mine QA-pairs, which are selected and clustered then to expand the knowledge base. DA recognition is challenging due to the same utterance may have a different meaning in a different context. Table 1 shows an example of some utterances together with their DAs from Switchboard dataset. In this example, utterance "Okay." corresponds to two different DA labels within different semantic context.

DA	Utterance
conventional-opening	B: Hi,
conventional-opening	B: this is Donna Donahue.
conventional-opening	A: Hi, Donna.
conventional-opening	B: Hi.
yes-no-question	A: Ready to get started?
yes answers	B: Uh, yeah,
statement-non-opinion	B: I think so.
other	A: Okay.
statement-non-opinion	A: Sort of an interesting topic since I just got back from lunch here.
acknowledge	B: Okay.

Table 1: A snippet of a conversation with the DA labels from Switchboard dataset.

Many approaches have been proposed for DA recognition. Previous work relies heavily on handcrafted features which are domain-specific and difficult to scale up (Stolcke et al., 2000; Kim et al., 2010; Tavafi et al., 2013). Recently, with great ability to do feature extraction, deep learning has yielded state-of-the-art results for many NLP tasks, and also makes impressive advances in DA recognition. Liu et al. (2017); Bothe et al. (2018) built hierarchical CNN/RNN models to encode sentence and incorporate context information for DA recognition. Kumar et al. (2018) achieved promising performance by adding the CRF to enhance the dependency between labels. Raheja and Tetreault (2019) applied the self-attention mechanism coupled with a hierarchical recurrent neural network.

However, previous approaches cannot make full use of the relative position relationship between utterances. It is natural that utterances in the local context always have strong dependencies in our daily dialog. In this paper, we propose a hierarchical model based on self-attention (Vaswani et al., 2017) and revise the attention distribution to focus on a local and contextual semantic information by a learnable Gaussian bias which represents the relative position information between utterances, inspired by Yang et al. (2018). Further, to analyze the effect of dialog length quantitatively, we introduce a new dialog segmentation mechanism for the DA task and evaluate the performance of different dialogue length and context padding length under online and offline settings. Experiment and visualization show that our method can learn the local contextual dependency between utterances explicitly and achieve promising performance in two well-known datasets.

The contributions of this paper are:

- We design a hierarchical model based on self-attention and revise the attention distribution to focus on a local and contextual semantic information by the relative position information between utterances.
- We introduce a new dialog segmentation mechaism for the DA task and analyze the effect of dialog length and context padding length.
- In addition to traditional offline prediction, we also analyze the accuracy and time complexity under the online setting.

2 Background

2.1 Related Work

DA recognition is aimed to assign a label to each utterance in a conversation. It can be formulated as a supervised classification problem. There are two trends to solve this problem: 1) as a sequence labeling problem, it will predict the labels for all utterances in the whole dialogue history (Dielmann and Renals, 2008; Lee and Dernoncourt, 2016; Kumar et al., 2018); 2) as a sentence classification problem, it will treat utterance independently without any context history (Kim et al., 2010; Khanpour et al., 2016). Early studies rely heavily on handcrafted features such as lexical, syntactic, contextual, prosodic and speaker information and achieve good results (Dielmann and Renals, 2008; Stolcke et al., 2000; Chen and Di Eugenio, 2013).

Recent studies have applied deep learning based model for DA recognition. Lee and Dernoncourt (2016) proposed a model based on RNNs and CNNs that incorporates preceding short texts to classify current DAs. Liu et al. (2017); Bothe et al. (2018) used hierarchical CNN and RNN to model the utterance sequence in the conversation, which can extract high-level sentence information to predict its label. They found that there is a small performance difference among different hierarchical CNN and RNN approaches. Kumar et al. (2018) added a CRF layer on the top of the hierarchical network to model the label transition dependency. Raheja and Tetreault (2019) applied the context-aware self-attention mechanism coupled with a hierarchical recurrent neural network and got a significant improvement over state-of-the-art results on SwDA datasets. On another aspect, Ji et al. (2016) combined a recurrent neural network language model with a latent variable model over DAs. Zhao et al. (2018) proposed a Discrete Information Variational Autoencoders (DI-VAE) model to learn discrete latent actions to incorporate sentence-level distributional semantics for dialogue generation.

2.2 Self-Attention

Self-attention (Vaswani et al., 2017) achieves great success for its efficiently parallel computation and long-range dependency modeling.

Given the input sequence $s = (s_1, ..., s_n)$ of n elements where $s_i \in \mathbb{R}^{d_s}$. Each attention head holds three parameter matrices, $W_h^Q, W_h^K, W_h^V \in \mathbb{R}^{d_s \times d_z}$ where h present the index of head. For the head h, linear projection is applied to the sequence s to obtain key (K), query (Q), and value (V) representations. the attention module gets the weight by computing dot-products between key/query pair and then $softmax$ normalizes the result. it is defined as:

$$ATT_h(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_z}}) \times V,$$

where $\sqrt{d_z}$ is the scaling factor to counteract this effect that the dot products may grow large in mag-

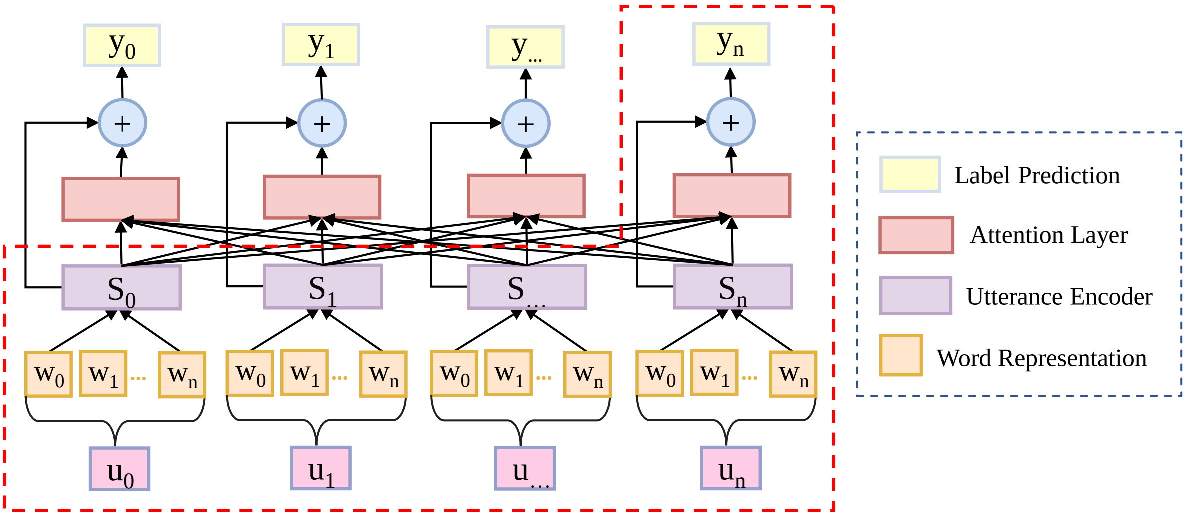


Figure 1: The model structure for DA recognition, where the LSTM with max pooling is simplified as utterance encoder in our experiment. The area in the red dashed line represents the structure for online prediction.

nitude. For all the heads,

$$output = Concat(ATT_1, ..., ATT_h) \times W^O,$$

where $W^O \in \mathbb{R}^{(d_z * h) \times d_s}$ is the output projection.

One weakness of self-attention model it that they cannot encode the position information efficiently. Some methods have been proposed to encode the relative or absolute position of tokens in the sequence as the additional input to the model. Vaswani et al. (2017) used sine and cosine functions of different frequencies and added positional encodings to the input embeddings together. It used absolute position embedding to capture relative positional relation by the characteristic of sine and cosine functions. Moreover, several studies show that explicitly modeling relative position can further improve performance. For example, Shaw et al. (2018) proposed relative position encoding to explicitly model relative position by independent semantic parameter. It demonstrated significant improvements even when entirely replacing conventional absolute position encodings. Yang et al. (2018) proposed to model localness for the self-attention network by a learnable Gaussian bias which enhanced the ability to model local relationship and demonstrated the effectiveness on the translation task.

In our study, we design a local contextual attention model, which incorporates relative position information by a learnable Gaussian bias into original attention distribution. Different from Yang et al. (2018), in our method, the distribution center is regulated around the corresponding utterance with a window, which indicates the context dependency preference, for capturing more local contextual dependency.

3 Methodology

Before we describe the proposed model in detail, we first define the mathematical notation for the DA recognition task in this paper. Given the dataset, $X = (D_1, D_2, ...D_L)$ with corresponding DA labels $(Y_1, Y_2, ...Y_L)$. Each dialogue is a sequence of N_l utterances $D_l = (u_1, u_2, ...u_{N_l})$ with $Y_l = (y_1, y_2, ...y_{N_l})$. Each utterance is padded or truncated to the length of M words, $u_j = (w_1, w_2, ...w_M)$.

Figure 1 shows our overall model structure. For the first layer, we encode each utterance u_j into a vector representation. Each word w_m of the utterance u_j is converted into dense vector representations e_m from one-hot token representation. And then, we apply LSTM (Hochreiter and Schmidhuber, 1997), a powerful and effective structure for sequence modeling, to encode the word sequence. Formally, for the utterance u_j:

$$e_m = embed(w_m) \quad \forall m \in \{1, ...M\}, \quad (1)$$

$$h_m = LSTM(h_{m-1}, e_m) \quad \forall m \in \{1, ...M\}, \quad (2)$$

where $embed$ represents the embedding layer which can be initialized by pre-trained embeddings. To make a fair comparison with previous work, we do not use the fine-grained embedding presented in Chen et al. (2018). LSTM helps us get the context-aware sentence representation for the input sequence. There are several approaches to represent the sentence from the words. Following Alexis et al. (2017), we add a max-pooling layer after LSTM, which selects the maximum value in each dimension from the hidden units. In our experiment, LSTM with max-pooling does perform a little better than LSTM with last-pooling, which is used in Kumar et al. (2018).

Afterwards, we get the utterances vector representations $u = (u_1, ..., u_{N_l})$ of N_l elements for the dialogue D_l where $u_j \in \mathbb{R}^{d_s}$, d_s is the dimension of hidden units. As we discussed in section 2.2, given the sequence $s \in \mathbb{R}^{N_l * d_s}$, self-attention mechanism calculates the attention weights between each pair of utterances in the sequence and get the weighted sum as output. The attention module explicitly models the context dependency between utterances. We employ a residual connection (He et al., 2016) around the attention module, which represents the dependency encoder between utterances, and the current utterance encoder s:

$$output = output + s. \quad (3)$$

Finally, we apply a two-layer fully connected network with a Rectified Linear Unit (ReLU) to get the final classification output for each utterance.

3.1 Modeling Local Contextual Attention

The attention explicitly models the interaction between the utterances. However, for context modeling, original attention mechanism always considers all of the utterances in a dialogue which inhibits the relation among the local context and is prone to overfitting during training. It is natural that utterances in the local context always have strong dependencies in our daily dialog. Therefore, we add a learnable Gaussian bias with the local constraint to the weight normalized by $softmax$ to enhance the interaction between concerned utterances and its neighbors.

The attention module formula is revised as:

$$ATT(Q, K) = softmax(\frac{QK^T}{\sqrt{d}} + POS). \quad (4)$$

The first term is the original dot product self-attention model. $POS \in \mathbb{R}^{N \times N}$ is the bias matrix, where N is the length of dialogue. The element $POS_{i,j}$ is defined following by gaussian distribution:

$$POS_{i,j} = -\frac{(j - c_i)^2}{2w_i^2}, \quad (5)$$

$POS_{i,j}$ measures the dependency between the utterance u_j and the utterance u_i in terms of the relative position prior. w_i represents for the standard deviation, which controls the weight decaying. Because of local constraint, $|c_i - i| <= C$, for each utterance u_i, the predicted center position c_i and window size w_i is defined as followed:

$$c_i = i + C \times tanh(W_i^c \times \overline{K}), \quad (6)$$

$$w_i = D \times sigmoid(W_i^d \times \overline{K}), \quad (7)$$

where $W_i^c, W_i^d \in \mathbb{R}^{1*N}$ are both learnable parameters. We initialized the parameter W_i^c to 0, which leads to center position $c_i = i$ by default. Furthermore, c_i and w_i are both related to the semantic context of the utterances, so we assign the mean of key $\overline{K}$ in attention mechanism to represent the context information. Moreover, the central position also indicates the dependency preference of the preceding utterances or subsequent utterances.

It is worth noting that there is a little difference with Yang et al. (2018), although we both revise the attention module by the Gaussian distribution. In our method, for the given utterance u_i, the distribution center c_i is regulated for capturing the not only local but also contextual dependency, which can be formally expressed as: $c_i \in (i - C, i + C)$. However, in their work, the distribution center can be anywhere in the sequence, and it is designed for capturing the phrasal patterns, which are essential for Neural Machine Translation task.

3.2 Online and Offline Predictions

Previous work mainly focuses on the offline setting where we can access the whole utterances in the dialogue and predict all the DA labels simultaneously. However, the online setting is the natural demand in our real-time applications. For the online setting, we only care about the recognition result of the last utterance in the given context, as seen in the area with the red dashed line in Figure 1, our model is well compatible with online setting, we can calculate the attention between the last utterance and the other utterances directly where $K \in \mathbb{R}^{1 \times d}, Q \in \mathbb{R}^{n \times d}, V \in \mathbb{R}^{n \times d}$. For LSTM, we still have to model the entire sequence, which is slower than attention based models. Table 2 shows the time complexity comparison excluding the time cost of first layer encoding, and the dialogue length n is smaller than the representation dimension d. Our model is easy to expand into the online setting, however, to have a fair comparison with previous work, in our experiments, we applied the models under the offline setting by default.

3.3 Separate into Sub-dialogues

The length of different dialogues in the dataset varies a lot. It is worth noting that the length of dialog affects the model prediction. On the one hand, under the offline setting, we can access the whole utterances in the dialogue and predict all the DA labels simultaneously, so the more utterances, the more efficient. However, on the other hand, if we put too many utterances in once prediction, it will model too much unrelated dependency in the long utterances sequence for both LSTM and attention mechanism based model. The sub-dialogues with the same length also enable efficiently batch training. To study how the dialogue length and context padding length will affect the performance, so we defined a sliding window W which is the sub-dialogue length. Then, we separate each long dialogue into several small sub-dialogues. For example, the dialog D is a sequence of utterances with length n, and we will get $\lceil x/w \rceil$ sub-dialogues, for the k-th sub-dialogues, the utterances sequence is $(u_{(k-1)*W+1}, u_{(k-1)*W+2}, ..., u_{k*W})$. In order to avoid losing some context information caused by being separated, which will affect the context modeling for the utterances in the begin and end of the sub-dialog, we add the corresponding context with P (stands for context padding) utterances at the begin and the end of each sliding window, so for the k-th sub-dialogues, the revised utterances sequence is $(u_{(k-1)*W-P+1}, u_{(k-1)*W-P+2}, ..., u_{k*W+P})$. Moreover, we mask the loss for the context padding utterances, which can be formally expressed as:

$$loss = \frac{1}{W}\sum_{i} M(i)L(\hat{y}_i, y_i), \qquad (8)$$

$M(i) = 0$ if utterance i is in the context padding otherwise 1, L is the cross entropy.

The W and P are both hyperparameters; in the experiment 4.2, we will talk about the effect of the window size and the context padding length.

model	offline setting	online setting
LSTM	$n \times d^2$	$n \times d^2$
Self-Attention	$n^2 \times d$	$n \times d$

Table 2: Time complexity between LSTM and self-attention for both online and offline predictions excluding the time cost of first layer encoding. The parameter n represents for the dialogue length in the sliding window and d represent for the dimension of representation unit.

Dataset	$\|C\|$	$\|U\|$	train	validation	test
SwDA	42	176	1K(177K)	112(18K)	19(4K)
Daily	4	8	11K(87K)	1K(8K)	1K(8K)

Table 3: $|C|$ indicates the number of classes. $|U|$ indicates the average length of dialogues. The train/validation/test columns indicate the number of dialogues (the number of sentences) in the respective splits.

4 Experiments

4.1 Datasets

We evaluate the performance of our model on two high-quality datasets: Switchboard Dialogue Act Corpus (SwDA) (Stolcke et al., 2000) and DailyDialog (Li et al., 2017). SwDA has been widely used in previous work for the DA recognition task. It is annotated on 1155 human to human telephonic conversations about the given topic. Each utterance in the conversation is manually labeled as one of 42 dialogue acts according to SWBD-DAMSL taxonomy (Jurafsky et al., 1997). In Raheja and Tetreault (2019), they used 43 categories of dialogue acts, which is different from us and previous work. The difference in the number of labels is mainly due to the special label "+", which represents that the utterance is interrupted by the other speaker (and thus split into two or more parts). We used the same processing with Milajevs and Purver (2014), which concatenated the parts of an interrupted utterance together, giving the result the tag of the first part and putting it in its place in the conversation sequence. It is critical for fair comparison because there are nearly 8% data has the label "+". Lacking standard splits, we followed the training/validation/test splits by Lee and Dernoncourt (2016). DailyDialog dataset contains 13118 multi-turn dialogues, which mainly reflect our daily communication style. It covers various topics about our daily life. Each utterance in the conversation is manually labeled as one out of 4 dialogue act classes. Table 3 presents

models	Acc(%)
previous approaches	
BLSTM+Attention+BLSTM (2019)	82.9
Hierarchical BLSTM-CRF (2018)	79.2
CRF-ASN (2018)	78.7[1]
Hierarchical CNN (window 4) (2017)	78.3[2]
mLSTM-RNN (2018)	77.3
DRLM-Conditional (2016)	77.0
LSTM-Softmax (2016)	75.8
RCNN (2013)	73.9
CNN (2016)	73.1
CRF (2010)	72.2
reimplemented and proposed approaches	
CNN	75.27
LSTM	75.59
BERT(2018)	76.88
LSTM+BLSTM	80.00
LSTM+Attention	80.12
LSTM+Local Contextual Attention	**80.34**
Human annotator	84.0

Table 4: Comparison results with the previous approaches and our approaches on SwDA dataset.

W	P	models	Acc(%)
1	0	CNN	75.27
		LSTM	75.59
		BERT	76.88
1	1	LSTM+BLSTM	78.60
		LSTM+Attention	78.74
1	3	LSTM+BLSTM	79.36
		LSTM+Attention	79.98
1	5	**LSTM+BLSTM**	**80.00**
		LSTM+Attention	**80.12**
5	5	LSTM+BLSTM	78.50
		LSTM+Attention	79.43
		LSTM+LC Attention	80.27
10	5	LSTM+BLSTM	78.31
		LSTM+Attention	79.00
		LSTM+LC Attention	**80.34**
20	5	LSTM+BLSTM	78.55
		LSTM+Attention	78.57
		LSTM+LC Attention	80.17
online prediction			
		LSTM+LSTM	78.62
		LSTM+Attention	78.86
		LSTM+LC Attention	**78.93**

Table 5: Experiment results about the hyperparameter W and P on SwDA dataset and online prediction result. W, P indicate the size of sliding window and context padding length during training and testing.

the statistics for both datasets. In our preprocessing, the text was lowercased before tokenized, and then sentences were tokenized by WordPiece tokenizer (Wu et al., 2016) with a 30,000 token vocabulary to alleviate the Out-of-Vocabulary problem.

4.2 Results on SwDA

In this section, we evaluate the proposed approaches on SwDA dataset. Table 4 shows our experimental results and the previous ones on SwDA dataset. It is worth noting that Raheja and Tetreault (2019) combined GloVe(Pennington et al., 2014) and pre-trained ELMo representations(Peters et al., 2018) as word embeddings. However, in our work, we only applied the pre-trained word embedding. To illustrate the importance of context information, we also evaluate several sentence classification methods (CNN, LSTM, BERT) as baselines. For baseline models, both CNN and LSTM, got similar accuracy (75.27% and 75.59% respectively). We also fine-tuned BERT (Devlin et al., 2018) to do recognition based on single utterance. As seen, with the powerful unsupervised pre-trained language model, BERT (76.88% accuracy) outperformed LSTM and CNN models for single sentence classification. However, it was still much lower than the models based on context information. It indicates that context information is crucial in the DA recognition task. BERT can boost performance in a large margin. However, it costs too much time and resources. In this reason, we chose LSTM as our utterance encoder in further experiment.

By modeling context information, the performance of the hierarchical model is improved by at least 3%, even compared to BERT. In order to better analyze the semantic dependency learned by attention, in our experiments, we removed the CRF module. In terms of different hierarchical models, our LSTM+BLSTM achieved good result. The accuracy was 80.00% which is even a little better than Hierarchical BLSTM-CRF (Ku-

[1] The author claimed that they achieved 78.7%(81.3%) accuracy with pre-trained word embedding (fine-grained embedding). For a fair comparison, both previous and our work is simply based on pre-trained word embedding.

[2] The author randomly selected two test sets which are different from previous and our work and achieved 77.15% and 79.74%, and we reimplemented in standard test sets.

mar et al., 2018). Relying on attention mechanism and local contextual modeling, our model, LSTM+Attention and LSTM+Local Contextual Attention, achieved 80.12% and 80.34% accuracy respectively. Compared with the previous best approach Hierarchical BLSTM-CRF, we can obtain a relative accuracy gain with 1.1% by our best model. It indicated that self-attention model can capture context dependency better than the BLSTM model. With adding the local constraint, we can get an even better result.

To further illustrate the effect of the context length, we also performed experiments with different sliding window W and context padding P. Table 5 shows the result. It is worth noting that it is actually the same as single sentence classification when $P = 0$ (without any context provided). First, we set W to 1 to discuss how the length of context padding will affect. As seen in the result, the accuracy increased when more context padding was used for both LSTM+BLSTM and LSTM+Attention approaches, so we did not evaluate the performance of LSTM+LC Attention when context padding is small. There was no further accuracy improvement when the length of context padding was beyond 5. Therefore, we fixed the context padding length P to 5 and increased the size of the sliding window to see how it works. With sliding window size increasing, the more context was involved together with more unnecessary information. From the experiments, we can see that both LSTM+BLSTM and LSTM+Attention achieved the best performance when window size was 1 and context padding length was 5. When window size increased, the performances of these two models dropped. However, our model (LSTM+LC Attention) can leverage the context information more efficiently, which achieved the best performance when window size was 10, and the model was more stable and robust to the different setting of window size.

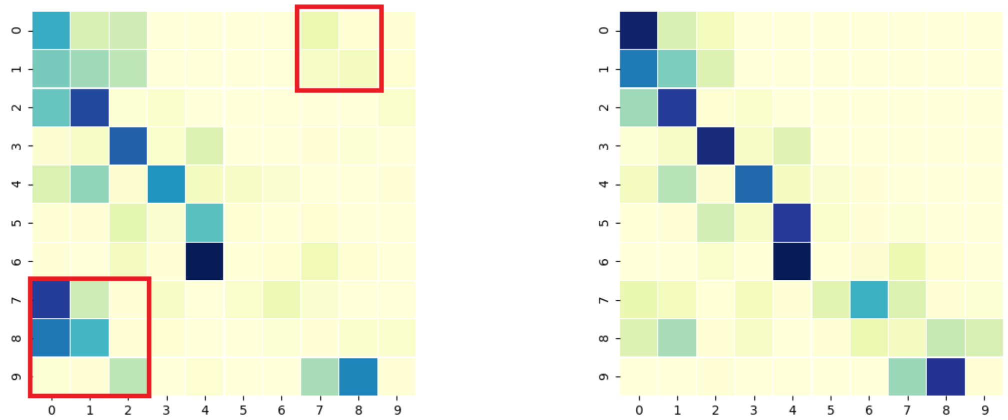
For online prediction, we only care about the recognition result of the last utterance in the given context. We added 5 preceding utterances as context padding for every predicted utterance because we cannot access subsequent utterances in the online setting. As seen in Table 5, without subsequent utterances, the performances of these three models dropped. However, LSTM+LC Attention still outperformed the other two models.

W	P	models	Acc(%)
1	0	CNN	82.22
		LSTM	82.58
		BERT	83.22
1	1	LSTM+BLSTM	84.88
		LSTM+Attention	85.10
1	2	LSTM+BLSTM	**85.06**
		LSTM+Attention	**85.36**
1	3	LSTM+BLSTM	84.97
		LSTM+Attention	85.05
5	2	LSTM+BLSTM	85.01
		LSTM+Attention	85.26
		LSTM+LC Attention	**85.81**
10	2	LSTM+BLSTM	84.97
		LSTM+Attention	85.13
		LSTM+LC Attention	85.72
online prediction			
		LSTM+LSTM	84.55
		LSTM+Attention	84.68
		LSTM+LC Attention	**84.83**

Table 6: Experiment results on DailyDialog dataset.

4.3 Result on DailyDialog

The classification accuracy of DailyDialog dataset is summarized in Table 6. As for sentence classification without context information, the fine-tuned BERT still outperformed LSTM and CNN based models. From table 3 we can see that, the average dialogue length $|U|$ in DailyDialog is much shorter than the average length of SwDA. So, in our experiment, we set the maximum of the W to 10, which almost covers the whole utterances in the dialogue. Using the same way as SwDA dataset, we, first, set W to 1 and increased the length of context padding. As seen, modeling local context information, hierarchical models yielded significant improvement than sentence classification. There was no further accuracy improvement when the length of context padding was beyond 2, so we fixed the context padding length P to 2 and increased the size of sliding window size W. From the experiments, we can see that LSTM+Attention always got a little better accuracy than LSTM+BLSTM. With window size increasing, the performances of these two models dropped. Relying on modeling local contextual information, LSTM+LC Attention achieved the best accuracy (85.81%) when the window size was 5. For the longer sliding window, the performance of LSTM+LC Attention was still better and



(a) original attention weight matrix

(b) local contextual attention weight matrix

Figure 2: Visualization of original attention and local contextual attention. Each colored grid represents the dependency score between two sentences. The deeper the color is, the higher the dependency score is.

more robust than the other two models. For online prediction, we added 2 preceding utterances as context padding, and the experiment shows that LSTM+LC Attention outperformed the other two models under the online setting, although the performances of these three models dropped without subsequent utterances.

4.4 Visualization

In this section, we visualize the attention weights for analyzing how local contextual attention works in detail. Figure 2 shows the visualization of original attention and local contextual attention for the example dialogue shown in Table 1. The attention matrix M explicitly measures the dependency among utterances. Each row of grids is normalized by $softmax$, M_{ij} represents for the dependency score between the utterance i and utterance j. As demonstrated in Figure 2a, there are some wrong and uninterpretable attention weights annotated with red color, which is learned by the original attention. The original attention model gives the utterance “B: Hi” (position 0) and “A: Okay.” (position 7) a high dependency score. However, local contextual attention weakens its attention weights due to the long distance apart.

Overall, the additional Gaussian bias trend to centralize the attention distribution to the diagonal of the matrix, which is in line with our linguistic intuition that utterances that are far apart usually don’t have too strong dependencies. As demonstrated in Figure 2b, benefiting of the additional Gaussian bias, the revised attention mechanism weakens the attention weights between utterances which cross the long relative distance. For the grids near diagonal, it strengthens their dependency score and doesn’t bring other useless dependencies for its learnable magnitude.

5 Conclusions and Future Work

In the paper, we propose our hierarchical model with local contextual attention to the Dialogue Act Recognition task. Our model can explicitly capture the semantic dependencies between utterances inside the dialogue. To enhance our model with local contextual information, we revise the attention distribution by a learnable Gaussian bias to make it focus on the local neighbors. Based on our dialog segmentation mechanism, we find that local contextual attention reduces the noises through relative position information, which is essential for dialogue act recognition. And this segmentation mechanism can be applied under online and offline settings. Our model achieves promising performance in two well-known datasets, which shows that modeling local contextual information is crucial for dialogue act recognition.

There is a close relation between dialogue act recognition and discourse parsing (Asher et al., 2016). The most discourse parsing process is composed of two stages: structure construction and dependency labeling (Wang et al., 2017; Shi and Huang, 2018). For future work, a promising direction is to apply our method to multi-task training with two stages jointly. Incorporating supervised information from dependency between utterances may enhance the self-attention and further improve the accuracy of dialogue act recognition.

References

Conneau Alexis, Kiela Douwe, Schwenk Holger, Barraul Lo c, and Bordes Antoine. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, page 670680.

Nicholas Asher, Julie Hunter, Mathieu Morey, Farah Benamara, and Stergos D Afantenos. 2016. Discourse structure and dialogue acts in multiparty dialogue: the stac corpus. In *LREC*.

J. L. Austin. 1962. *How to Do Things with Words*. Clarendon Press.

Chandrakant Bothe, Cornelius Weber, Sven Magg, and Stefan Wermter. 2018. A context-based approach for dialogue act recognition using simple recurrent neural networks. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*.

Lin Chen and Barbara Di Eugenio. 2013. Multimodality and dialogue act classification in the robohelper project. In *Proceedings of the SIGDIAL 2013 Conference*, pages 183–192.

Zheqian Chen, Rongqin Yang, Zhou Zhao, Deng Cai, and Xiaofei He. 2018. Dialogue act recognition via crf-attentive structured network. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 225–234. ACM.

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

Alfred Dielmann and Steve Renals. 2008. Recognition of dialogue acts in multiparty meetings using a switching dbn. *IEEE transactions on audio, speech, and language processing*, 16(7):1303–1314.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

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

Yangfeng Ji, Gholamreza Haffari, and Jacob Eisenstein. 2016. A latent variable recurrent neural network for discourse relation language models. In *Proceedings of NAACL-HLT*, pages 332–342.

Dan Jurafsky, Elizabeth Shriberg, and Debra Biasca. 1997. Switchboard swbd-damsl labeling project coders manual. *Draft 13. Technical Report 97-02*.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent convolutional neural networks for discourse compositionality. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, pages 119–126.

Hamed Khanpour, Nishitha Guntakandla, and Rodney Nielsen. 2016. Dialogue act classification in domain-independent conversations using a deep recurrent neural network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2012–2021.

Su Nam Kim, Lawrence Cavedon, and Timothy Baldwin. 2010. Classifying dialogue acts in one-on-one live chats. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 862–871. Association for Computational Linguistics.

Harshit Kumar, Arvind Agarwal, Riddhiman Dasgupta, and Sachindra Joshi. 2018. Dialogue act sequence labeling using hierarchical encoder with crf. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Ji Young Lee and Franck Dernoncourt. 2016. Sequential short-text classification with recurrent and convolutional neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 515–520.

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 the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995. Asian Federation of Natural Language Processing.

Yang Liu, Kun Han, Zhao Tan, and Yun Lei. 2017. Using context information for dialog act classification in dnn framework. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2170–2178.

Dmitrijs Milajevs and Matthew Purver. 2014. Investigating the contribution of distributional semantic information for dialogue act classification. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, pages 40–47.

Ryosuke Nakanishi, Koji Inoue, Shizuka Nakamura, Katsuya Takanashi, and Tatsuya Kawahara. 2018. Generating fillers based on dialog act pairs for smooth turn-taking by humanoid robot. In *Proc. Intl Workshop Spoken Dialogue Systems (IWSDS)*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237.

Vipul Raheja and Joel Tetreault. 2019. Dialogue act classification with context-aware self-attention. *arXiv preprint arXiv:1904.02594*.

John Rogers Searle. 1969. *Speech acts: An essay in the philosophy of language*, volume 626. Cambridge university press.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, volume 2, pages 464–468.

Zhouxing Shi and Minlie Huang. 2018. A deep sequential model for discourse parsing on multi-party dialogues. *arXiv preprint arXiv:1812.00176*.

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.

Maryam Tavafi, Yashar Mehdad, Shafiq Joty, Giuseppe Carenini, and Raymond Ng. 2013. Dialogue act recognition in synchronous and asynchronous conversations. In *Proceedings of the SIGDIAL 2013 Conference*, 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 *Advances in Neural Information Processing Systems*, pages 5998–6008.

Yizhong Wang, Sujian Li, and Houfeng Wang. 2017. A two-stage parsing method for text-level discourse analysis. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 184–188.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Baosong Yang, Zhaopeng Tu, Derek F Wong, Fandong Meng, Lidia S Chao, and Tong Zhang. 2018. Modeling localness for self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4449–4458.

Tiancheng Zhao, Kyusong Lee, and Maxine Eskenazi. 2018. Unsupervised discrete sentence representation learning for interpretable neural dialog generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1098–1107. Association for Computational Linguistics.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. *arXiv preprint arXiv:1703.10960*.