DialogConv: A Lightweight Fully Convolutional Network for Multi-view Response Selection

Anonymous ACL submission

Abstract

Current end-to-end retrieval-based dialogue systems are primarily based on Recurrent Neural Networks or Transformers with attention mechanisms. Despite promising results have been achieved, these models usually suffer from slow inference speed or an enormous amount of parameters. In this paper, we propose a novel lightweight fully convolutional architecture called DialogConv for the response selection. DialogConv is built exclusively on convolutions for distilling the matching features of context and response. The dialogue is modeled in a 3D view, where DialogConv conducts convolution operations on embedding dimension, word dimension and utterance dimension iteratively to capture richer semantic information from a multi-view of context. On four benchmark datasets, DialogConv is approximately 4.0x smaller and up to 27x faster in inference compared with strong baselines. Moreover, DialogConv can achieve competitive performance results on four public datasets.

1 Introduction

An important challenge for building intelligent dialogue systems is the response selection problem, which aims to select a proper response from a set of candidates given the context of a conversation. Such retrieval-based dialogue systems have drawn great attention from academic and industrial communities owing to the advantage of informative and fluent responses (Tao et al., 2021).

The existing retrieval-based dialogue systems can be categorized into two patterns: (i) Separate Pattern (Wu et al., 2017; Zhang et al., 2018b; Zhou et al., 2018; Gu et al., 2019); (ii) Concatenated Pattern (Tan et al., 2015; Zhou et al., 2016). Separate Pattern (i.e., Figure 1 (a)) encodes the utterance one by one separately, while Concatenated Pattern (i.e., Figure 1 (b)) concatenates all utterances into a consecutive word sequence. Methods based on these two patterns usually take RNNs (Hochreiter and Schmidhuber, 1997; Cho et al., 2014) and attention mechanism (Bahdanau et al., 2014) as the backbone. Although promising results have been achieved, these methods are often slow for both training and inference due to their recurrent nature.

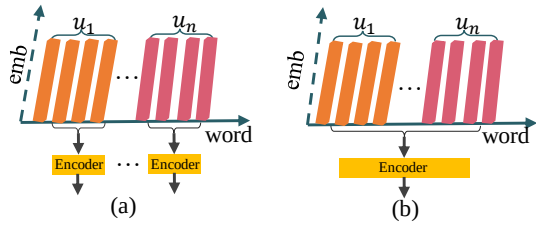
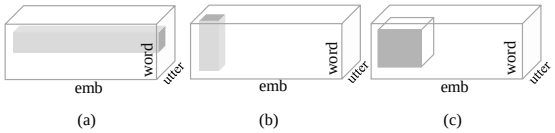


Figure 1: Flat pattern. (a) is separate pattern and (b) is concatenated pattern.

More recently, Pre-trained Language Model (PrLM) pattern (Cui et al., 2020; Gu et al., 2020; Liu et al., 2021) has obtained the state-of-the-art performance in response selection. However, these methods that taking Transformer as the de-facto standard architecture suffer from an enormous amount of parameters and heavy computational cost. The extremely large model scale not only leads to increased training cost but also prevents researchers from rapid iteration. Meanwhile, the slow inference speed hinders the dialogue systems from being deployed in real-world scenarios.

Additionally, previous studies (Sankar et al., 2019; Li et al., 2021) demonstrate that Concatenated Pattern is usually insensitive to dynamic associated features between utterances. On the other hand, the Separate Pattern lacks contextual information when encoding each individual utterance. In fact, the matching features for response selection are more likely to appear in local context (Lu et al., 2019). Convolution structure is naturally adept in capturing the local structure features of text, and thus suitable for capturing dynamic matching features between dialogue context and response.



(a) (b) (c)

Figure 2: Fully convolutional multi-view modeling. (a) is convolution in embedding view, (b) is convolution in word view and (c) is convolution in utterane view.

In this paper, we propose a fully[1] convolutional network model (dubbed as DialogConv) without any RNN or attention module for multi-view response selection. Different from previous study (Zhou et al., 2016) that models dialogue in plane view, DialogConv models the dialogue context and response together in 3D space from stereo view, namely embedding view, word view and utterance view (i.e. Figure 2). In embedding view, convolutions are conducted on the plane formed by the word sequence dimension and utterance dimension. In word view, convolutions are conducted on the plane composed of embedding dimension and utterance dimension. In utterance view, we conduct convolutions along the depth of conversation to extract dialogue discourse features across utterances. Intuitively, for the convolution operation in Figure 2 (b) and (c), we assume that the more semantic features can be distilled with considering a scalar of embedding as semantic unit (Zhang et al., 2018a).

DialogConv based exclusively on CNN utilizes much fewer parameters and computing resources. The average number of parameters of DialogConv is 12.4M, which is 3.5x to 4.0x smaller than RNN based models. The inference speed of the model can be up to 26.9x faster than existing models. Moreover, DialogConv attains competitive results on four benchmarks, and even better performance when pre-trained using contrastive learning. To summarize, we make the following contributions:

- We propose an efficient convolutional response selection model DialogConv, which, to the best of our knowledge, is the first response selection model exclusively built on multiple convolution layers without any RNN or attention module.
- We model the dialog from a stereo view, where 2D and 1D convolution operations are conducted on word, utterance and embedding dimensions iteratively, and thus finer-grained and dynamic matching features can be captured.
- Extensive experiments on four benchmark datasets show that DialogConv can achieve competitive results with faster speed and fewer computing resources.

[1]Here 'fully' means DialogConv is built exclusively on CNNs.

2 Related Work

2.1 Retrieval-based Dialogue Framework

Most existing methods follow the *Encoding-Interaction-Aggregation-Prediction* process, which takes *interaction* between dialogue context and response as the core. These methods try to mine deep-semantic features by sequence modeling, for example, using attention-based pairwise matching mechanism to capture the interactive features between dialogue context and candidate response. However, previous studies (Sankar et al., 2019; Li et al., 2021) demonstrate that most existing mehtods are insensitive to dynamic features across utterances. Besides, most existing methods employ recurrent structure to model the sequence features of utterances, which leads to the slow inference speed of the model. Although methods transformer-based get rid of the weakness of recurrent structure, these methods usually enjoy huge amount of parameters. Training and inference demand heavy computation cost. In this paper, we propose to model dialogue context from multi-view using the fully convolutional structure. DialogConv is a lightweight model which more small and faster compared with most existing methods.

2.2 Convolutional Neural Networks

In the past few years, CNN has been the go-to model in the computer vision field. The main reason is that CNN enjoys the advantage of parameter sharing and high concurrency. Besides, convolutional structure is better at modeling the local structure. A large number of excellent architectures based on CNN have been proposed (Krizhevsky et al., 2012; Simonyan and Zisserman, 2014; Szegedy et al., 2015; He et al., 2016; Dai et al., 2021). Compared with Recurrent Neural Networks (Hochreiter and Schmidhuber, 1997), convolutional structure is adept at capturing local local dependencies of text and faster. Compared with Transformer, convolutional structure is more lightweight. In this paper, we propose a novel fully

convolutional architecture using convolution layers to encode dialogue utterances from multi-view. Compared with existing methods, DialogConv not only enjoys the advantage of faster inference speed but also can capture finer-grained and dynamic matching features.

3 Methodology

3.1 Problem Formalization

The instance in the dialogue dataset can be represented as (c, r, y). Here we treat multi-turn response selection in chatbots as binary classification task. Specifically, the union of dialogue context c and response r are regarded as the overall context $c = \{u_1, u_2, ..., u_{t-1}, r\}$. r is a candidate response and $y \in \{0, 1\}$, where $y = 1$ indicates that r is a proper response for c; otherwise $y = 0$. As the core of retrieval based dialogue system, the aim of response selection is to build a discriminator $g(c)$ on (c, y), which measures the matching score between c and r.

3.2 Fully Convolutional Multi-view Matching

We propose a fully convolution encoder for multi-view response selection. Multiple views include: embedding view, word view, and utterance view. In embedding view, the convolution operations are conducted in the plane formed by word dimension and utterance dimension, which allow communication between different embeddings. In word view, the convolution operations are conducted in the plane formed by embedding dimension and utterance dimension, which allow communication between different words. In word view, the utterance-level and context-level features will be distilled when convolution operations are across a single utterance and whole dialogue sequence, respectively. In utterance view, we consider each utterance as an utterance-level semantic flow and extract the dialogue flow feature along the depth of dialogue.

Note that the structure of DialogConv, which follows the matching process of *Local-Contextual-Discourse*, is not strictly designed according to the views. DialogConv will mix features iteratively from multiple views in each matching stage. Figure 3 shows an overview of our proposed DialogConv that contains six layers: (i) Embedding Layer; (ii) Local matching layer; (iii) Context matching layer; (iv) Discourse matching layer; (v) Aggregation layer; (vi) Prediction Layer.

Symbol Definition: conv@i represents the i-th convolution operation in Figure 3, d represents the dimension of word embedding, s stands for length of utterance, and t is the number of utterances including response. $G \in R^{t*s*d}$ is a 3D tensor which represents the input of DialogConv. The prediction layer is a fully-connected layer.

3.2.1 Local Matching Layer

The local matching layer is in charge of distilling features of each utterance. The local matching stage contains features from embedding and word view. First, we employ 1×1 convolutions in word view and embedding view respectively. The process can be described formally as:

$$G_1 = Conv2D^{embedding}_{1\times 1_{t\times s}}(\sigma(G)) \quad (1)$$

$$G_2 = Conv2D^{word}_{1\times 1_{t\times d}}(G_1) + G \quad (2)$$

where $\sigma(\cdot)$ stands for GELUs (Hendrycks and Gimpel, 2016) activation function, $Conv2D^{embedding}_{1\times 1_{t\times s}}$ represents 2D convolution with a convolution kernel size of 1×1 is employed in word view and $Conv2D^{word}_{1\times 1_{t\times d}}$ represents 2D convolution with a convolution kernel size of 1×1 is performed in embedding view. The 1×1 convolution pays attention to the information of the current element itself and does not consider the influence of the local context. When 1×1 convolution is employed in embedding view, the abstract semantic features of words will be captured by communication between different words. When 1×1 convolution is employed in word view, the features of words will be captured by communication between different embeddings. Multi-scale convolution (Szegedy et al., 2015; Gao et al., 2019) has been proven to be effective in extracting local features. Therefore, we employ 1×3 convolution in word view and 1×1 convolution in embedding view to capture the local matching features. The formal description is as follows:

$$G_3 = Conv2D^{embedding}_{1\times 3_{t\times s}}(\sigma(G_2)) \quad (3)$$

$$G_4 = Conv2D^{word}_{1\times 1_{t\times d}}(G_3) + G_2 \quad (4)$$

Note that we focuse on features for a single utterance in local matching layer.

3.2.2 Context Matching Layer

The context matching layer is responsible for distilling the matching features based on whole dialogue context. First, we flatten G_4 into a two-dimensional tensor $G_5 \in R^{(t*s)*d}$. This is equivalent to concatenating all utterances into a single consecutive word

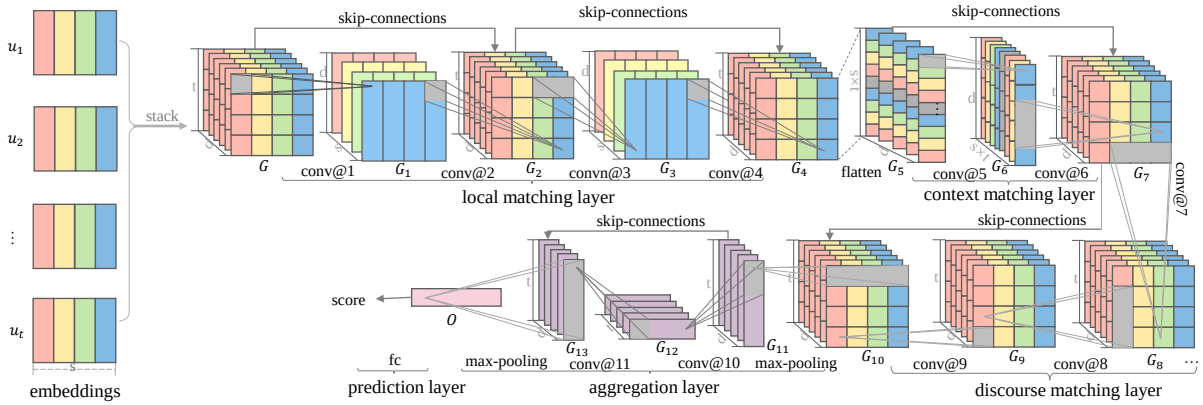


Figure 3: Overview of our DialogConv.

sequence in chronological order. Then we employ convolutions across words sequence with kernel size of 1 in embedding view, and kernel size of 5 across embedding dimension in word view. The details are as follows:

$$G_6 = Conv1D_{1_l}^{embedding}(\sigma(G_5)) \quad (5)$$

$$G_7 = f_{reshape}(Conv1D_{5_d}^{word}(G_6)) + G_5 \quad (6)$$

where $f_{reshape}$ is a function, standing for reshaping the output of the convolution into the same shape as G_5, $G_7 \in R^{t*s*d}$ and $l = t * s$. The context-level features can be aggregated through communication between different words concatenating all utterances. Context-level information is fundamental for extracting discourse features across different utterances.

3.2.3 Discourse Matching Layer

Previous layers are responsible for understanding dialogue utterances according the contextual information. The discourse matching layer pays attention to the implied non-sequential dialogue flow features along the depth of the dialogue. Capturing dynamic semantic features across utterances is very important to selecting correct response. We employ orthogonal convolution to extract dynamic dialogue flow features across utterances. The specific process is as follows:

$$G_8 = Conv2D_{1\times3_{t\times s}}^{embedding}(\sigma(G_7)) \quad (7)$$

$$G_9 = Conv2D_{3\times1_{t\times s}}^{embedding}(G_8) \quad (8)$$

$$G_{10} = Conv2D_{1\times1_{t\times s}}^{embedding}(G_9) + G_7 \quad (9)$$

The 1×3 convolution and 3×1 convolution are called as orthogonal convolutions because the directions of their convolution kernels are vertical. The 1×3 convolution is responsible for building semantic flow based on context-level features for single utterance and 3×1 convolution distills the dialogue flow features across depth of dialogue. Finally, we integrate the infomation by 1×1 convolution. Note that we adopt orthogonal convolution in word view to realize the extraction of the features of the dialogue flow. The detailed reasons will be discussed in the experiment section.

3.2.4 Aggregation Layer

The aggregation layer is responsible for getting high-level semantic infomation by integrating mathching features from previous layers. First, we employ max-pooling to obtain the sentential representation $G_{11} \in R^{t*d}$. Then we adopt two layers of convolution to distill matching features along embedding dimension and depth of dialogue respectively. The description is as follows:

$$G_{12} = Conv1D_{t}^{embedding}(G_{11}) \quad (10)$$

$$G_{13} = Conv1D_{s}^{utterance}(G_{12}) + G_{11} \quad (11)$$

We employ the max-pooling operation again based on G_{13} to get the final contextual representation O.

3.3 Self-supervised Pre-training

As a lightweight neural structure, the performance of DialogConv can be further improved by pre-training strategy using small-scale corpus. The way of masked language model pre-training (Devlin et al., 2019; Lan et al., 2020) usually requires large-scale corpus, while the self-supervised contrastive learning can learn general representation feature in relatively small-scale corpus.

Therefore, we employ contrastive learning to learn effective representation by pulling semantically close neighbors together and pushing apart

non-neighbors (Hadsell et al., 2006). Given a set of paired examples $D = (x_i, x_i^+)$, where x_i is the conversation context c, x_i^+ is the correct response. We adopt the previous contrastive learning framework and take a cross-entropy objective with negatives x_i^- that includes responses with $y = 0$ and in-batch negatives (Chen et al., 2017). The training objective is:

$$l_c = log \frac{e^{sim(x_i, x_i^+)/\tau}}{\sum_{j=1}^{|x_i^-|} e^{sim(x_i, x_{ij}^-)/\tau} + e^{sim(x_i, x_i^+)/\tau}} \quad (12)$$

where τ is a temperature hyperparameter and $sim(\cdot, \cdot)$ is the cosine similarity.

4 Experiments

4.1 Datasets

In this paper, we conduct extensive experiments on four public datasets: (i) Ubuntu Dialogue (Ubuntu) (Lowe et al., 2015); (ii) Multi-Turn Dialogue Reasoning (MuTual) (Cui et al., 2020); (iii) Douban Conversation Corpus (Douban) (Wu et al., 2016); (iv) E-commerce Dialogue Corpus (ECD) (Zhang et al., 2018b).

Ubuntu consists of 1 million context-response pairs for training, 0.5 million pairs for validation, and 0.5 million pairs for testing. The ratio of the positive and the negative is 1:1 in training, and 1:9 in validation and testing. **Douban** consists of 1 million context-response pairs for training, 50k pairs for validation, and 10k pairs for testing. Response candidates are retrieved from Sina Weibo and labeled by human judges. **ECD** contains 1 million context-response pairs for training, 10k pairs for validation, and 10k pairs for testing and consists of five different types of conversations (e.g., commodity consultation, logistics express, recommendation, negotiation and chitchat) based on over twenty commodities. **MuTual** is the first human-labeled reasoning-based dataset for multi-turn dialogue, which contains 7,088 context-response pairs for training, 886 pairs for validation, and 886 pairs for testing. The ratio of the positive and the negative is 1:3 in training, validation and testing. Note that the description of the Metrics is in the appendix.

4.2 Baselines

TF-IDF (Lowe et al., 2015) is a traditional method of information retrieval. **LSTM** is a Long Short-Term Memory neural network. **BiLSTM** is a bidirectional long and short-term memory neural network. **MV-LSTM** (Wan et al., 2016) is a semantic matching method based on LSTM. **QANET** (Yu et al., 2018) is a machine reading comprehension method based on CNN. **MH-LSTM** (Wang and Jiang, 2016) is an extractive machine reading comprehension model based on LSTM. **BIDAF** (Seo et al., 2017) is a machine reading comprehension model based on bi-directional attention flow. **R-NET** (Wang et al., 2017) is a machine reading comprehension model with multiple attention. **Multi-View** (Zhou et al., 2016) is a multi-turn dialogue retrieval-based method based on token view and utterance view. **DL2R** (Yan et al., 2016) is a multi-turn retrieval-based dialogue model based on sentence pair matching. **SMN** (Wu et al., 2017) is a matching model based on attention mechanism. **DUA** (Zhang et al., 2018b) is a hierarchical interaction model based on attention mechanism. **DAM** (Zhou et al., 2018) is a deep interaction method based on attention. **IMN** (Gu et al., 2019) is a retrieval-based dialogue model with bi-directional matching. **MRFN** (Tao et al., 2019) is a retrieval-based dialogue model with multiple types of representations. **IoI** (Tao et al., 2019) is a retrieval-based dialogue model based on multiple interactions. **MSN** (Yuan et al., 2019) is a retrieval-based dialogue model with multi-hop selector mechanism. **BERT** (Devlin et al., 2019) is an autoencoding language model based on Transformer.

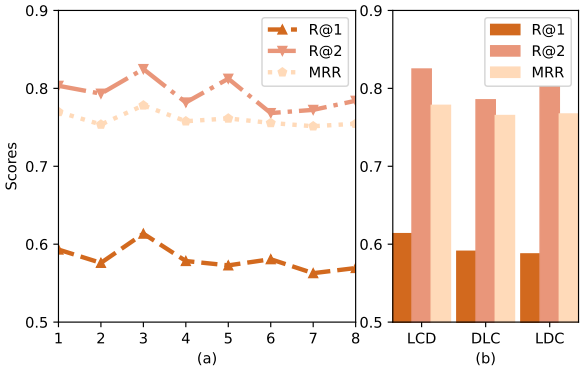


Figure 4: (a) is the impact of different granularity of convolution along depth of dialogue (i.e., conv@8). (b) is the performance comparison of DialogConv-like models.

4.3 Implementation Details

We implement DialogConv using Tensorflow 2, and train DialogConv on Intel(R) Core(TM) i7-10700 CPU 2.90HZ*16 with a single GeForce RTX 2070 SUPER (8G). We consider at most 10 turns and

	Ubuntu			Douban					
Method	R10@1	R10@2	R10@5	MAP	MRR	P@1	R10@1	R10@2	R10@5
CNN	0.549	0.684	0.896	0.417	0.440	0.226	0.121	0.252	0.647
LSTM	0.638	0.784	0.949	0.485	0.537	0.320	0.187	0.343	0.720
Bi-LSTM	0.630	0.780	0.944	0.479	0.514	0.313	0.184	0.330	0.716
MV-LSTM	0.653	0.804	0.946	0.498	0.538	0.348	0.202	0.351	0.710
MH-LSTM	0.653	0.799	0.944	0.500	0.537	0.345	0.202	0.348	0.720
Multi-View	0.662	0.801	0.951	0.505	0.543	0.342	0.202	0.350	0.729
DL2R	0.626	0.783	0.944	0.488	0.527	0.330	0.193	0.342	0.705
SMN	0.726	0.847	0.961	0.529	0.569	0.397	0.233	0.396	0.724
DUA	0.757	0.868	0.962	0.551	0.599	0.421	0.243	0.421	0.780
DAM	0.767	0.874	0.969	0.550	0.601	0.427	0.254	0.410	0.757
MRFN	0.786	0.886	0.976	0.571	0.617	0.448	0.276	0.435	0.783
IMN	0.794	0.889	0.974	0.570	0.615	0.433	0.262	0.452	0.789
IoI	0.796	0.894	0.974	0.573	0.621	0.444	0.269	0.451	0.786
MSN	0.800	0.899	0.978	0.587	0.632	**0.470**	**0.295**	0.452	0.788
BERT	**0.808**	0.897	0.975	**0.591**	0.633	0.454	0.280	**0.470**	**0.828**
DialogConv	0.788	0.883	**0.979**	0.571	0.624	0.432	0.272	0.453	0.785
DialogConv*	0.801	**0.904**	0.976	0.572	**0.634**	0.457	0.282	0.452	0.825

Table 1: Results on Ubuntu and Douban datasets. The first group model adopts Concatenated Pattern. The second group model belongs to Separate Pattern. The third group model belongs to PrLM-based Pattern. DialogConv* represents the performance when pre-training using contrastive learning. **Bold** indicates the best result and underline indicates the second best result.

	ECD			MuTual		
Method	R10@1	R10@2	R10@5	R@1	R@2	MRR
TF-IDF	-	-	-	0.279	0.536	0.542
CNN	0.328	0.515	0.792	-	-	-
LSTM	0.365	0.536	0.828	-	-	-
Bi-LSTM	0.365	0.536	0.825	0.260	0.491	0.743
MV-LSTM	0.412	0.591	0.857	-	-	-
QANET	0.455	0.662	0.920	0.247	0.517	0.522
BIDAF	0.491	0.708	0.933	0.357	0.589	0.589
RNET	0.362	0.500	0.770	0.270	0.435	0.513
MH-LSTM	0.410	0.590	0.858	-	-	-
Multi-View	0.421	0.601	0.861	-	-	-
DL2R	0.399	0.571	0.842	-	-	-
SMN	0.453	0.654	0.886	0.299	0.585	0.595
DUA	0.501	0.700	0.921	0.437	0.698	0.658
DAM	0.526	0.727	0.933	0.458	0.718	0.673
IMN	0.621	0.797	0.964	0.404	0.622	0.638
IoI	0.563	0.768	0.950	0.421	0.686	0.647
MSN	0.606	0.770	0.937	0.420	0.677	0.646
BERT	0.610	0.814	**0.973**	**0.648**	0.847	**0.795**
DialogConv	0.827	0.889	0.962	0.602	0.834	0.769
DialogConv*	**0.844**	**0.891**	0.963	0.622	**0.854**	0.782

Table 2: Results on ECD and MuTual datasets. The first group model adopts Concatenated Matching. The second group model belongs to Separate Interaction. The third group model belongs to PrLM-based Interaction. DialogConv* represents the performance when pre-training using contrastive learning. **Bold** indicates the best result and underline indicates the second best result.

50 words for Ubuntu, Douban, ECD while at most 8 turns and 50 words for MuTual in the experiments. The dimension of word embeddings is set to 200. Two-dimensional convolution is used in 1st, 2nd, 3rd, 4th, 8th, 9th and 10th layers while one-dimensional convolution is employed in 6th, 7th, 12th and 13th layers. The stride of all convolution layer is [1,1] or 1. The filters size of 1st, 2nd, 4th, 5th, 9th and 11st convolution layers is [1,1]. [1,3], 5, [1,3], [3,1] and 3 are filters size for the 3rd, 6th, 7th, 8th and 10th convolution layer respectively. In the supervised learning stage, we train DialogConv and other models with Adam optimizer while use Stochastic Gradient Descent (SGD) for optimizing in unsupervised stage. In the supervised training stage, staged optimization strategy is employed and learning rates are initialized as 1e-3, 5e-4, 1e-4, 5e-5 and 1e-5. The batch-size is 32 for the MuTual and 64 for the other datasets. In the pre-training stage, we only set the batch size 128. Larger batch size can be set for better results under the devices allow. The temperature τ is set to 0.007.

4.4 Results

Table 1 and 2 report the testing results of DialogConv as well as all comparative models on four datasets. Although DialogConv does not achieve state-of-the-art performance, the model can attain near-optimal results in most cases. Besides, we calculate the confidence level ($p<0.05$) of DialogConv, which demonstrates the results of DialogConv are credible. The best results are shown in bold text. On Ubuntu dataset, DialogConv outperforms most classic models such as SMN, DUA, DAM, and has comparable performance with MRFN. When pre-training with contrastive learning, the performance of DialogConv is close to BERT, even outperform-

Models	Per	Nodes	Edges	Inference Time(CPU/GPU)				Parameters(M)			
				Ubuntu(m)	Douban(s)	ECD(s)	MuTual(s)	Ubuntu	Douban	ECD	MuTual
SMN	1.508	2,417	4,283	39.0/22.2	46.8/30.1	43.4/28.6	14.9/12.4	90.2	68.5	9.8	4.4
DAM	1.453	12,85	22,226	176.7/45.3	227.3/68.1	226.8/65.7	90.5/37.9	94.8	67.1	13.1	8.0
DUA	1.461	4,412	7,797	142.9/49.3	176.1/64.3	174.6/64.2	63.7/26.3	96.4	69.7	15.7	14.8
IOI	1.493	1,1704	283,731	346.7/39.2	421.3/48.5	429.1/47.0	156.7/22.2	96.0	69.3	15.3	10.2
MSN	1.452	955	1,468	105.3/12.9	127.9/16.7	125.5/13.8	44.6/7.0	89.1	62.4	10.5	13.1
DialogConv	**1.424**	**216**	**329**	**12.9/5.4**	**17.5/7.4**	**16.4/7.0**	**7.0/3.4**	**22.9**	**13.3**	**9.3**	**4.1**

Table 3: Comparison of model complexity and inference time. The perplexity (i.e., $Per = 2^L$ and L is the average loss on validation dataset of all corpora.) is computed based on the average loss on a validation dataset of all corpora. Nodes represent the number of nodes in the model graph. Edges represent the number of edges in the model graph.

ing BERT on R10@2. On Douban dataset, the performance is 1.3% lower than the best result on R10@1. However, the performance of pre-trained DialogConv can achieve near-optimal results. It is a surprise on ECD dataset that DialogConv has an absolute advantage of 21.7% on R10@1 and 7.5% on R10@2. DialogConv on MuTual dataset outperforms the compared baseline models[2], including some classic machine reading comprehension models such as QANET, BIDAF RENT. The pre-trained DialogConv can achieve comparable results with BERT. Note that DialogConv does not use large-scale pre-trained word vectors, such as GloVe based on Common Crawl corpus[3].

DialogConv achieves relatively better results on ECD and MuTual dataset. We conduct further analysis for this phenomenon and find that the MuTual dataset contains many reused contexts. In other words, the context of one example is likely to be part of the context of the other examples. We conjecture that DialogConv based on convolution structure is good at capturing local dynamic features across utterances compared with general sequence models. For the ECD dataset, compared with the Douban and Ubuntu datasets, the positive and negative responses are easier to identify because the fact that the difference is obvious. DialogConv can incorporate features of multi-view stereo, which is more sensitive to differences in semantic and makes it easier to select the correct response from candidate responses.

4.5 Model Complexity and Inference Time

To measure the simplicity of our base model, we analyze the model from multiple dimensions. Here we have selected some relatively lightweight models among the existing methods. It is obvious that language models are large and bloated. For example, the parameter quantity of BERT$_{base}$ is 110M and BERT$_{large}$ is 340M. Compared with language models, the advantages of DialogConv are obvious. Table 3 compares the model complexity and inference time of DialogConv and some classic models. According to perplexity, the result of DialogConv is reliable. The third and fourth column show the number of nodes and edges in the model graph. DialogConv possesses 216 nodes and 329 edges. The number of nodes in DialogConv is 4.4x to 54.2x less than other models. The number of edges in DialogConv is 4.5x to 864.4x less than other models. The faster inference speed and fewer model parameters are important in real-world scenarios. The average parameter of DialogConv is 12.4M, which is 3.5x to 4.0x smaller than other models.

Besides, we test the practical inference time of models on CPU and GPU. DialogConv has an absolute speed advantage over other models, no matter on CPU or GPU. DialogConv is 2.15x to 9.65x faster on the GPU device and 2.61x to 19.90x faster on the CPU device than other models. The CPU and GPU are described in Implementation Details subsection above. DialogConv is faster than other models because it employs lightweight CNN structure, which has greater advantages in terms of speed compared to Recurrent Neural Networks. The main reason is that DialogConv employ fully convolutional structure and does not rely on complex attention-based interaction structures, which consume huge computing resources.

Method	MuTual			ECD		
	R@1	R@2	MRR	R10@1	R10@2	R10@5
DialogConv	0.614	0.825	0.778	0.833	0.901	0.988
-LocM	0.580	0.786	0.754	0.813	0.881	0.958
-ConM	0.577	0.801	0.759	0.806	0.823	0.919
-DisM	0.578	0.785	0.753	0.810	0.845	0.910
-Agg	0.573	0.783	0.750	0.804	0.824	0.870

Table 4: Module-level ablation experiment results of DialogConv on validation set.

[2]https://nealcly.github.io/MuTual-leaderboard/

[3]https://github.com/stanfordnlp/GloVe

4.6 Ablation Study

Table 4 reports the results of module ablation. **-LocM**: removing the local matching layer; **-ConM**: removing the context matching layer; **-DisM**: removing the discourse matching layer; **-Agg**: replacing the aggregation layer with max-pooling. We can observe that each sub-module plays a critical role in DialogConv. Specifically, the local matching layer can capture the utterance-level features by mixing features from embedding and word view. The context matching layer will update matching features based on whole dialogue context and response. According to Table 4, the local matching layer has the least impact on model performance. We conjecture that the context matching layer can distill local features to some extent due to the characteristics of convolution layer. The discourse matching layer allows word with local contextual information interaction in different utterances, which can distill implied dynamic features across utterances. Therefore discourse matching plays a vital role in extracting the matching features.

4.7 Discussion

4.7.1 Discourse Matching

According to the above analysis, the matching features may appear in local context. Note that the local here is relative to the dialogue context. In other words, the local utterance sequences contain more valuable feature information. In fact, it should conduct convolutions on the plane formed by embedding dimension and word sequence dimension. However, these operations will capture features across whole depth of dialogue not the local features. Therefore, we employ orthogonal convolutions along the depth of dialogue in the word view due to the sparsity and feature locality of the dialogue. We conduct in-depth experiments to investigate the influence of convolution along the depth of dialogue on DialogConv (i.e., conv@8). As shown in Figure 4 (a), the performance of DialogConv begins to decline when the filter size increases to a certain extent. This phenomenon verifies the feature locality of the dialogue. In addition, we replace the conv@8 with $Conv2D^{utterance}_{3\times 1_{s\times d}}(G_8)$ to distill matching features along the depth of whole dialogue. The results of DialogConv drop dramatically. We believe that this phenomenon is consistent with the characteristics of dialogue. The correlation between the farther utterances (e.g u_1 and u_{10}) is weak or even irrelevant. Intuitively, the closer utterances are more relevant (e.g., u_1 and u_2), fusion of infomation in local context is beneficial for distilling matching features.

4.7.2 Fully Convolution Structure

DialogConv only employs convolutions to distill the matching features between dialogue context and response. In multi-turn dialogues, the local context of dialogue is time-sensitive. This is because the topic and intention in a conversation may change over time (Feng et al., 2021). The dynamic implied features are benefical for selecting correct response. Compared with RNN, CNN is better at modeling the local dependencies. In fully convolution setting, we encode the dialogue context from multi-view.

Different from the flat pattern based dialogue modeling in previous studies (Zhou et al., 2016), DialogConv models the dialogue in a stereo view. Another difference is that convolutions we employed are better for capturing local dynamic features. In order to explore the rationality of full convolutional encoder, we conduct further experimental analysis. For comparison, we denote the original model as LCD (i.e., Local-Contextual-Discourse). We exchange discourse matching layer with local matching layer and context matching layer respectively, getting the model denoted as DLC and LDC. We think that the premise for discourse mathcing layer works is based on contextual features. As shown in the results of Figure 4 (b), the performance of DialogConv is best when discourse matching layer based on the context matching layer.

4.8 Conclusion

In this paper, we propose DialogConv, a multi-view lightweight architecture based exclusively on CNN. DialogConv conducts convolutions on embedding view, word view, and utterance view iteratively to capture matching features. DialogConv can capture more richer semantic infomation through fusing features from multi-view. The model we proposed is faster and has fewer parameters compared with existing models. Experiment results show that DialogConv requires less computing resources to achieve competitive results on Ubuntu, Douban, ECD and MuTual datasets. DialogConv provides a valuable reference for the dialogue system being deployed in the real-world scenarios.

References

Ricardo A. Baeza-Yates and Berthier Ribeiro-Neto. 1999. *Modern Information Retrieval*.

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

Ting Chen, Yizhou Sun, Yue Shi, and Liangjie Hong. 2017. On sampling strategies for neural network-based collaborative filtering.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Leyang Cui, Yu Wu, Shujie Liu, Yue Zhang, and Ming Zhou. 2020. Mutual: A dataset for multi-turn dialogue reasoning. *arXiv preprint arXiv:2004.04494*.

Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. 2021. Coatnet: Marrying convolution and attention for all data sizes. *arXiv preprint arXiv:2106.04803*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT 2019: Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186.

Xiachong Feng, Xiaocheng Feng, Libo Qin, Bing Qin, and Ting Liu. 2021. Language model as an annotator: Exploring dialogpt for dialogue summarization. *arXiv preprint arXiv:2105.12544*.

Shanghua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip HS Torr. 2019. Res2net: A new multi-scale backbone architecture. *IEEE transactions on pattern analysis and machine intelligence*.

Jia-Chen Gu, Tianda Li, Quan Liu, Zhen-Hua Ling, Zhiming Su, Si Wei, and Xiaodan Zhu. 2020. Speaker-aware bert for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2041–2044.

Jia-Chen Gu, Zhen-Hua Ling, and Quan Liu. 2019. Interactive matching network for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2321–2324.

R. Hadsell, S. Chopra, and Y. LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742.

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.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

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

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *ICLR 2020 : Eighth International Conference on Learning Representations*.

Zekang Li, Jinchao Zhang, Zhengcong Fei, Yang Feng, and Jie Zhou. 2021. Conversations are not flat: Modeling the dynamic information flow across dialogue utterances. *arXiv preprint arXiv:2106.02227*.

Yongkang Liu, Shi Feng, Daling Wang, Kaisong Song, Feiliang Ren, and Yifei Zhang. 2021. A graph reasoning network for multi-turn response selection via customized pre-training. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13433–13442.

Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. *arXiv preprint arXiv:1506.08909*.

Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 285–294.

Junyu Lu, Chenbin Zhang, Zeying Xie, Guang Ling, Tom Chao Zhou, and Zenglin Xu. 2019. Constructing interpretive spatio-temporal features for multi-turn responses selection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 44–50.

Chinnadhurai Sankar, Sandeep Subramanian, Christopher Pal, Sarath Chandar, and Yoshua Bengio. 2019. Do neural dialog systems use the conversation history effectively? an empirical study. *arXiv preprint arXiv:1906.01603*.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *ICLR 2017 :*

International Conference on Learning Representations 2017.

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.

Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2015. Lstm-based deep learning models for non-factoid answer selection. *arXiv preprint arXiv:1511.04108*.

Chongyang Tao, Jiazhan Feng, Rui Yan, Wei Wu, and Daxin Jiang. 2021. A survey on response selection for retrieval-based dialogues. IJCAI.

Chongyang Tao, wei wu, Can Xu, Wenpeng Hu, Dongyan Zhao, and Rui Yan. 2019. One time of interaction may not be enough: Go deep with an interaction-over-interaction network for response selection in dialogues. In *ACL 2019 : The 57th Annual Meeting of the Association for Computational Linguistics*, pages 1–11.

Chongyang Tao, Wei Wu, Can Xu, Wenpeng Hu, Dongyan Zhao, and Rui Yan. 2019. Multi-representation fusion network for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the twelfth ACM international conference on web search and data mining*, pages 267–275.

Ellen M Voorhees et al. 1999. The trec-8 question answering track report. In *Trec*, volume 99, pages 77–82.

Shengxian Wan, Yanyan Lan, Jiafeng Guo, Jun Xu, Liang Pang, and Xueqi Cheng. 2016. A deep architecture for semantic matching with multiple positional sentence representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 1.

Shuohang Wang and Jing Jiang. 2016. Machine comprehension using match-lstm and answer pointer. *arXiv preprint arXiv:1608.07905*.

Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 189–198.

Yu Wu, Wei Wu, Chen Xing, Ming Zhou, and Zhoujun Li. 2016. Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. *arXiv preprint arXiv:1612.01627*.

Yu Wu, Wei Wu, Chen Xing, Ming Zhou, and Zhoujun Li. 2017. Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 496–505.

Rui Yan, Yiping Song, and Hua Wu. 2016. Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 55–64.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. In *International Conference on Learning Representations*.

Chunyuan Yuan, Wei Zhou, Mingming Li, Shangwen Lv, Fuqing Zhu, Jizhong Han, and Songlin Hu. 2019. Multi-hop selector network for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 111–120.

Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. 2018a. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856.

Zhuosheng Zhang, Jiangtong Li, Pengfei Zhu, Hai Zhao, and Gongshen Liu. 2018b. Modeling multi-turn conversation with deep utterance aggregation. *arXiv preprint arXiv:1806.09102*.

Zhuosheng Zhang and Hai Zhao. 2021. Advances in multi-turn dialogue comprehension: A survey. *arXiv preprint arXiv:2103.03125*.

Xiangyang Zhou, Daxiang Dong, Hua Wu, Shiqi Zhao, Dianhai Yu, Hao Tian, Xuan Liu, and Rui Yan. 2016. Multi-view response selection for human-computer conversation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 372–381.

Xiangyang Zhou, Lu Li, Daxiang Dong, Yi Liu, Ying Chen, Wayne Xin Zhao, Dianhai Yu, and Hua Wu. 2018. Multi-turn response selection for chatbots with deep attention matching network. In *ACL 2018: 56th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1118–1127.

Xiangyang Zhou, Lu Li, Daxiang Dong, Yi Liu, Ying Chen, Wayne Xin Zhao, Dianhai Yu, and Hua Wu. 2018. Multi-turn response selection for chatbots with deep attention matching network. In *Proceedings of the 56th Annual Meeting of the Association*

for Computational Linguistics (Volume 1: Long Papers), pages 1118–1127.

A Appendix

A.1 Metrics

We follow previous studies (Zhang and Zhao, 2021) employing evaluation metric Rn@k to measure model performance for datasets Ubuntu, Douban, ECD, which calculates the proportion of true positive response among the top-k selected responses from the list of n available candidates for one context. Besides, additional conventional metrics MAP (Mean Average Precision) (Baeza-Yates and Ribeiro-Neto, 1999) and MRR (Mean Reciprocal Rank) (Voorhees et al., 1999) are employed on Douban. We employ recall at position 1 in 4 candidates(R@1), recall at position 2 in 4 candidates(R@2) and MRR are used for MuTual, which follow previous study (Liu et al., 2021).

A.2 Convolution Visualization

Feature visualization is a more intuitive way to observe model behavior. Figure 6 presents visualized result of some utterances of an example (i.e., Figure 5) from MuTual and Figure 7 presents the visualized result of the correct response for the corresponding example. Figure 6 shows the visualization of the output for the discourse matching layer. Some key features are marked by the red rectangle. DialogConv is easier to learn features that appear in the response. For example, DialogConv learns the key features of *"teachers"* and *"electronically"* that appear in the correct response. To our surprise that DialogConv learns some indirect features that do not appear directly in the response, for instance, *"school"* related to teacher and *"green"* related to *"electronically"*. We conjecture that the multi-view modeling method allows DialogConv to extract matching features from stereo view, which endows the model the ability to find the association between features.

A: Tim, You 're going to talk about your project and how to lead a **greener life**. Why did you choose that subject ?

B: Well. We 'd learned a lot about the environment in our science lessons, so I decided to see what I could do in my own life rather than just act completely helpless. And I knew the rest of my family would be interested.

A: Did you find it easy to get information ?

B: Yeah, I discovered there were lots of people at my age trying to be **green**. I 'd always gone to school by car. catching a bus would be better, but there 's no bus. where we live. **So I 've gone for riding my bike to school now.**

A: Ok. And what about being **green** once you 're actually at school?

B: Well, I realized that although all school paper was recycled and most of my friends use both sides of paper. We use huge quantities and I thought we should **cut down** and then it came to me that we should be **sending in** most of our **work electronically**. I 'm going to **recommend it to our teachers.**

A: So you are going to **advice your teachers** to ask students to send in their **homework electronically**, right

Figure 5: An example from MuTual (Cui et al., 2020). The last utterance is the correct response. Direct key information has been marked in blue. And indirect key information is marked in purple.

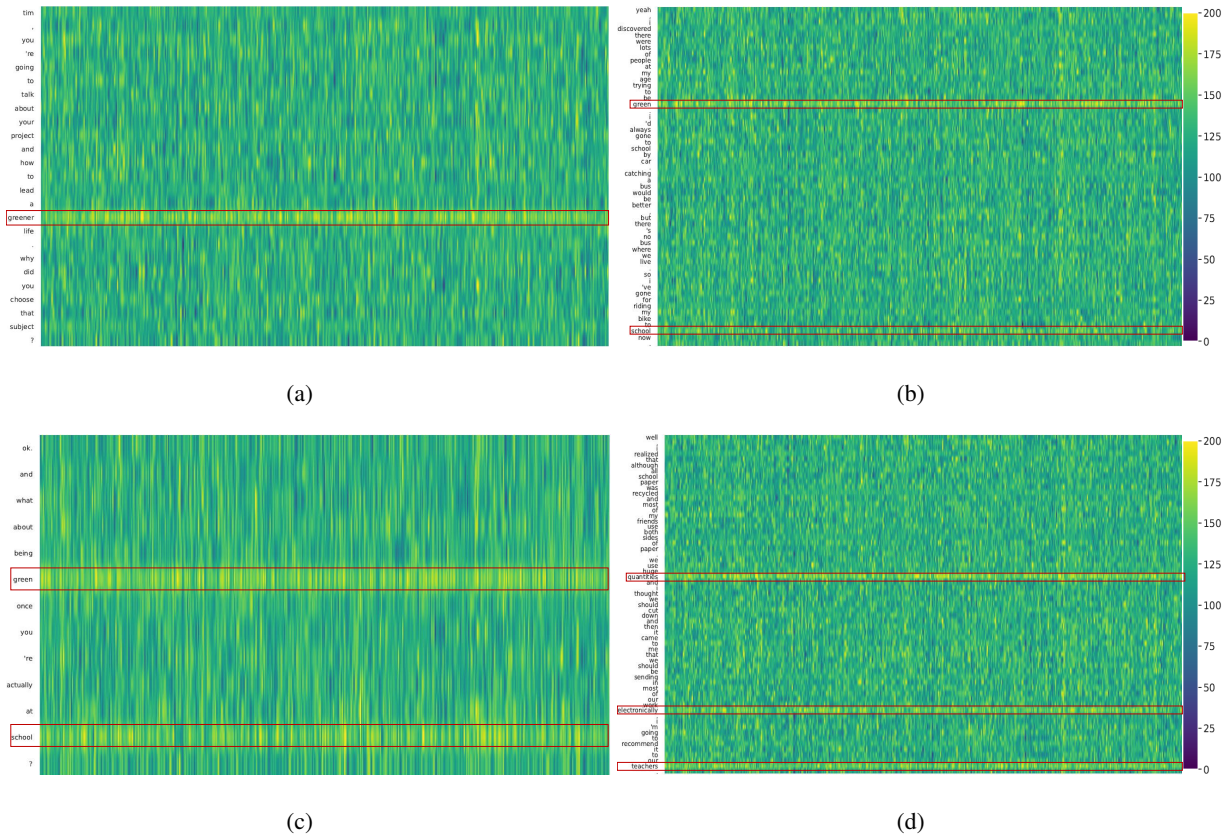


Figure 6: The feature visualization results G_{10} of the matching layers. (a) is the result of the first utterance's convolution feature visualization. (b) is the result of the fourth utterance's convolution feature visualization. (c) is the result of the fifth utterance's convolution feature visualization. (d) is the result of the sixth utterance's convolution feature visualization. The larger the color value is, the more important the feature is. Keywords recognized by DialogConv have been marked in the red rectangle.

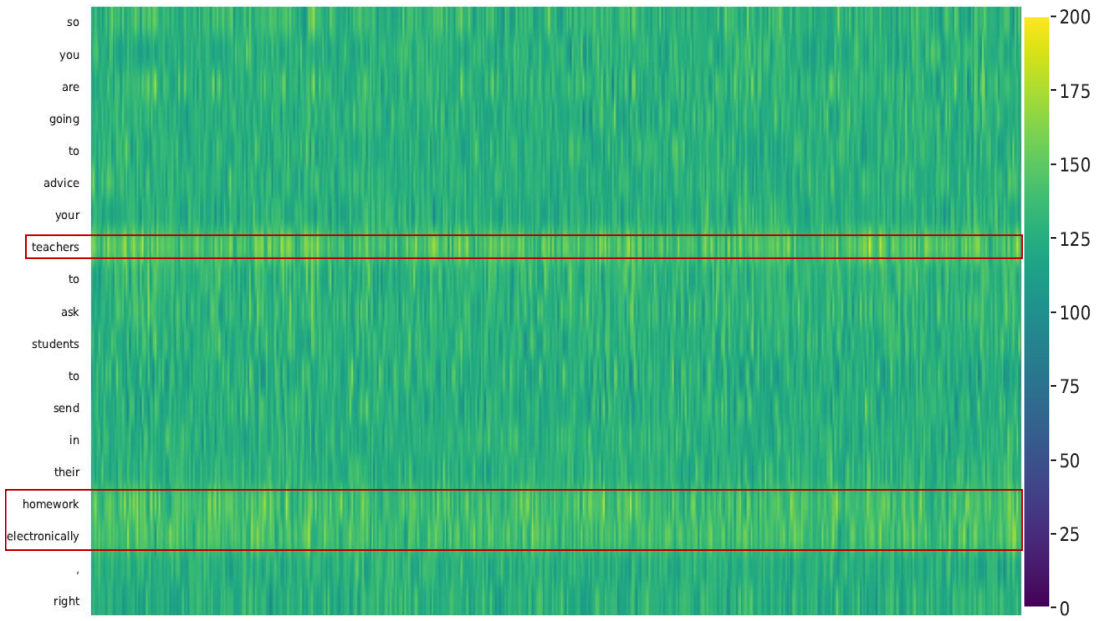


Figure 7: The feature visualization result of the correct response.