A Pre-trained Document-Grounded Conversation Model with Learning Low-frequency Patterns

Anonymous ACL submission

Abstract

Currently, the Generative Pre-trained Transformer model (GPT-2) has achieved remarkable performance in document-grounded dialogue generation since high-frequency patterns in the large corpora are well memorized during its pre-train procedure. However, it is still hard to capture low-frequency task-specific patterns especially when directly taking given documents and dialogue context as input. Here we propose an encoder-decoder framework including a semantic-oriented encoder and GPT-2 decoder with knowledge-aware classification, which strengthens the learning of two following task-specific patterns. One pattern is how to semantically select the crucial information of dialogue context and corresponding history knowledge from documents; the other is when to generate a response with knowledge since many responses do not contain it. With learned high- and low-frequency patterns, empirical study shows that our method has better generative performance than state-of-the-arts.

1 Introduction

In the last years, there are tons of elaborate work thrusting into the field of open-domain dialogue generation and reaching good results (Vinyals and Le, 2015; Serban et al., 2016; Tian et al., 2017; Zhang et al., 2018a; Zheng et al., 2020b; Huang et al., 2020; Wang et al., 2021). However, the generic generation remains, which refers to the generated responses that are meaningless and boring, such as, "Yes, of course." They make conversations between agents and users difficult to continue. Many works (Xing et al., 2017; Chen et al., 2018; Ghazvininejad et al., 2018; Zheng et al., 2020b) attempted to address this problem using different techniques, but there is still much room to improve.

Recently, some researchers have realized that document-grounded conversation is an effective solution to solve generic responses, i.e., generating informative responses by selecting appropriate knowledge from given documents with dialogue context (Zheng et al., 2020a). In general, there are two key steps to generate document-grounded responses. The former step is to learn a $background$ pattern that can capture the crucial information from dialogue context and corresponding history knowledge as candidate information for the generation. The latter one is to acquire a $selection$ pattern that obtains most related parts in candidate information to select appropriate knowledge from given documents and then generate informative and coherent responses.

There is a line of research to learn the patterns better. In earlier work (Zhou et al., 2018; Zhao et al., 2019), an RNN-based model is used to learn the background pattern and encode dialogue context and documents. Meanwhile, another RNN-based model with the general attention mechanism (Bahdanau et al., 2015) is employed to learn the selection pattern and generate responses. Later, with the great success of Transformer (Vaswani et al., 2017), some research works (Dinan et al., 2018; Tang and Hu, 2019; Li et al., 2019; Kim et al., 2019, 2020) utilize its encoder to acquire a better semantic representation for dialogue context and documents, as well as, its decoder is employed for better dialogue generation. More importantly, the multi-head and dot-product attention of Transformer are effectively used to learn the background and selection patterns.

Very recently, large-scale pre-trained language models exhibit compelling performance in NLP generation task, such as GPT2 (Radford et al., 2019). Especially for open-domain dialogue, the prototype DialoGPT (Zhang et al., 2020) can reach unprecedented results, i.e., better semantic coherent responses with more contextual details. Then, the TransferTransfo (Thomas et al., 2019) initially utilizes the GPT-2 to address document-ground dialogue generation (i.e., the conversation based on

Document	**...The film stars Macaulay Culkin as Kevin McCallister, a boy who is mistakenly left behind when his family flies to Paris for their Christmas...**
Context	A: Hello! How did you like the movie Home Alone? B: It was entertaining, funny, and moving. How did you like it?
Human	A: I thought **Culkin did a nice job as Kevin McCallister**. It was entertaining. But it has mediocre reviews. I wonder why.
TransferTransfo	A: I thought it was an interesting movie and I was very interested in watching it.
Our model	A: I thought it was a great **family** movie. I think **Macaulay Culkin did a great job portraying Kevin McCallister**.

Figure 1: An example from testing datasets of CMU_DoG.

given documents containing the agent's persona). Without the task-specific architectures, the TransferTransfo directly put all kinds of information (containing dialogue context and all documents) into the GPT2 with the simple losses of a cross entropy and next-utterance classification, and its performance is also state-of-the-art at that time.

One reason is that high-frequency patterns in large corpora learned by the pre-train are well memorized by the GPT-2, such as grammar (Vig, 2019; Clark et al., 2019), syntactic (Hewitt and Manning., 2019), commonsense (Davison et al., 2019) and even world knowledge (Petroni et al., 2020; Wang et al., 2020). They greatly improve the semantic coherence and appropriateness of knowledge selection in generated responses. For example, in the line 4 of Figure 1, the proper word "interesting" is more likely generated when the commonsense "Entertaining, funny, and moving usually means interesting" is pre-learned by the GPT-2.

Although the trend of the new paradigm (Brown et al., 2020) is to remove the need for task-specific architectures and directly fine-tune pre-trained models for downstream tasks, here we argue that 1) high-frequency patterns cannot replace low-frequency patterns (e.g., background and selection patterns) (Shin et al., 2020; Liu et al., 2021); 2) without task-specific architectures, simultaneously training background and selection patterns significantly increase the difficulty of learning. In Table 1, TransferTransfo can generate a coherent response but it does not capture appropriates specific knowledge from given documents.

Targeting the problems, we propose a document-grounded Double-classification Dialogue generation model (DcDial) with an encoder-decoder framework for learning separate. To learn the background pattern, our semantic-oriented encoder sequentially utilizes two modules to semantically select contextual key information and corresponding history knowledge as background information since a good response is a correct semantic extension of the dialogue context. Meanwhile, our GPT-2 decoder is primarily responsible for knowledge selection through one classification task. Generally, existing methods ignored the truth that many utterances do not contain knowledge from given documents. Thus, a binary classification task is introduced into the decoder as a soft gate for knowledge selection. It emphasizes the learning of knowledge-awareness for response generation.

To further reduce the difficulty of learning and remove potential noise from the encoder (Zhao et al., 2020), inspired by the wake-sleep algorithm (Ikeda et al., 1999), in the beginning, we first train the encoder separately and then train the decoder. Hence, a next-utterance classification task is set on the encoder as a binary classification, which distinguishes a correct next utterance from randomly sampled utterances from training datasets. Until the parameters of the encoder have converged by the classification task, the decoder is trained (fine-tuning) to optimize a combination of two-loss functions: a knowledge-aware binary classification loss and the cross-entropy loss between predicted word distribution and the true one.

Our contributions in this paper are three-fold:

- proposing an encoder-decoder framework for document-grounded dialogue generation, the semantic-oriented encoder and the GPT-2 decoder with knowledge-aware classification can successfully learn task-specific background and selection patterns respectively;
- separately training the encoder and decoder that siginificantly reduce the difficulty of learning and potential noise from the encoder;
- carried out a set of experiments on various datasets and the results show that our method outperforms other SOTA baselines.

2 Related work

Document-grounded dialogue generation is to generate informative responses by absorbing the proper knowledge from given documents. So far, most related works use an encoder-decoder framework plus a document/knowledge-selection module to generate responses. With the rapid development of neural networks, the three parts in generative methods continue to evolve from time to time.

In earlier work, RNN and standard attention (Bahdanau et al., 2015) are dominated. (Zhou et al., 2018) uses a shared bi-LSTM for encoding dialogue context and given documents while the decoder and knowledge selection module are implemented by another LSTM with global attention (Luong et al., 2015) and copy mechanism (See et al., 2017). (Ye et al., 2020) first use two bi-GRU for encoding dialogue context and ground-truth responses and 1D-CNN for documents encoding, then a double-attention mechanism on context and documents is implemented for knowledge selection. Finally, the decoder based on CVAE summarizes encoded information to guide response generation.

Later, following the framework of Transformer (Vaswani et al., 2017), many works (Dinan et al., 2018; Tang and Hu, 2019; Li et al., 2019; Kim et al., 2019, 2020) attempted to utilize Transformer's dot-product and multi-head attention to build their methods. For instance, in (Dinan et al., 2018), dialogue context and documents are encoded by a shared Transformer encoder as background information. Then the dot-product attention for the knowledge selection is applied to utilizing the context vector to select documents vectors. The concatenation of selected document and dialogue context vectors is feed into a Transformer decoder for response generation. In (Li et al., 2019), the authors provide an incremental encoder with multi-head self-attention for encoding dialogue context and corresponding documents sequentially. A two-pass Transformer decoder is used to improve context coherence (in the first pass) and the knowledge relevance (in the second pass). In (Tang and Hu, 2019) and (Kim et al., 2020), the variants of the Transformer's encoder and decoder are used for learning background information and response generation while VAE and deliberation models are used for knowledge selection respectively.

Nowadays, the pre-trained model is profoundly changing the domain in deep learning, like BERT (Devlin et al., 2019), GPT (Brown et al., 2020; Radford et al., 2018, 2019) and their variants (Lewis et al., 2020). They not only inherit the advantages of Transformer but also enjoy the benefits of the large-scale pre-trained parameters. More researchers try to use only a pre-trained model to address a downstream task by fine-tuning task-specific datasets, such as the following works. The authors in TransferTransfo first directly use GPT-2 for document-grounded conversation and reach a SOTA performance. Unlike the previous work, the GPT-2 model handles all three parts of learning, i.e., the encoder for background learning, attention modules for knowledge-selection learning and the decoder for generation. It strongly proves that high-frequency patterns in language captured by large-scale parameters are significantly helpful for three-part learning. Following (Thomas et al., 2019), KnowledGPT (Zhao et al., 2020) propose a more practical GPT-based conversation model. But the difference is that a retrieval-like module based on the BERT tailors given documents to meet the length constraint for a GPT-2 model.

Unlike (Thomas et al., 2019) and (Zhao et al., 2020), our model build on the classical encoder-decoder framework instead of one main GPT-2 model in order to separately learn low-frequency patterns for background and knowledge selection. Such task-specific modules help to reduce the burden of GPT-2 on learning low-frequency patterns. In addition, the traditionally training method (training the encoder and decoder together) will lead to $vanishing\ phenomenon$ since the decoder (GPT2) is stronger than our encoder, i.e., the output of the encoder is ignored and the whole model degenerated into a GPT-2 model (like TransferTransfo) when the quality of the encoder result is low at the beginning phase of the training process (Fu et al., 2019; Bowman et al., 2016). Thus, we introduce new classification task and different optimizing method to address the problem.

3 Problem formalization

The problem is formally defined as follows. At the T-th turn, let $X = \{U_1, ..., U_t, ..., U_T\}$ be a dialogue history (also referred as dialogue context) and each U_t represents an utterance from a user or an agent. Each utterance is a sequence of discrete words with varying length $U_t = \{w_{t,1}, w_{t,2}, ..., w_{t,|U_t|}\}$ where $w_{t,i} (1 \leq i \leq |U_t|)$ is the i-th word and $|U_t|$ is the length of utterance U_t. For each utterance U_t, there is a specified relevant document $D_t = \{d_{t,1}, ..., d_{t,|D_t|}\}$ where $d_{t,j} (1 \leq j \leq |D_t|)$ is the j-th word and $|D_t|$ is the length of document D_t. Note that $D_1, ..., D_{T+1}$ may be identical. Our goal is to generate a next response $\bar{U}_{T+1}$ given its dialogue context X, its relevant documents $D_{\leq T}$ and D_{T+1} (which are the knowledge of $\bar{U}_{T+1}$ selected from).

$$P(\bar{U}_{T+1}|X, D_{\leq T+1}; \theta) = \prod_{i=1}^{|\bar{U}_{T+1}|} P(w_{T+1,i}|w_{T+1,<i}, X, D_{\leq T+1}; \theta) \quad (1)$$

where $w_{T+1,<i} = w_{T+1,1}, ..., w_{T+1,i-1}$.

4 Our model

Our model is based on an encoder-decoder framework, i.e., the semantic-oriented encoder with next-utterance classification and the GPT-2 decoder with a knowledge-aware classification. Figure 2 shows the overview of our model.

4.1 Semantic-oriented encoder

As we mentioned before, a good response must be a correct semantic extension of its dialogue context with the knowledge, and usually the last utterance is the bond to connect the response and the context. Thus, first we use one shared self-attention module from Transformer to encode dialogue context X and last utterance U_T respectively. For each module, its input is U_t embedded as follows:

$$Em(U_t) = [e(w_{t,1}), ..., e(w_{t,|U_t|})] \quad (2)$$

where $e(w_{t,i})$ $(1 \leq t \leq T)$ is the word embedding implemented by one matrix borrowed from the counterpart of GPT-2 model (Radford et al., 2019). Each self-attention module contains a stack of N identical layer, each layer has two sub-layers, the first sub-layer is a multi-head self-attention. Each head attention takes a query matrix Q, a key matrix K and a value matrix V as input and the attention function is shown in Equation 3. Here Q, K and V are from the products of the matrix $[Em(U_1), ..., Em(U_T)]$ and three different matrixes due to the self-attention.

$$Z_i = Softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (3)$$

where $i \leq h$ (h is the number of the heads) is the head index and d_k is the size of the dimension of K. The output of the first layer is the matrix $A = [Z_1; ...; Z_h]W^o$ (W^o a transformation matrix). The second sub-layer is fully connected feed-forward network (FFN). The FFN includes two linear transformations with ReLU activation function, its input and output are A and $Y = FFN(A)$ ($FFN(x) = max(0, xW_1 + b_1)W_2 + b_2$). Notice that residual connection and layer normalization are used in each layer as sub-layers. For simplicity, they are omitted here.

After encoding dialogue context and last utterance, the encoded last utterance is used to select the information from the encoded context through a context-attention module, which contains N layers and each one has three sublayers: a multi-head self-attention, a multi-head context-attention and a FFN. Here the multi-head context-attention is almost same as the aforementioned self-attention except that K and V are the output of the multi-head attention for $U_{<T}$.

Similarly, the relevant documents of dialogue context are encoded by another self-attention module and its key information is learned by a knowledge attention module. For the knowledge attention, its K and V are the encoded history documents and Q is the output of the context-attention module that contains the learned key information of dialogue context. It means that the crucial information (i.e., knowledge) of documents is learned by the selected dialogue context. After a knowledge attention module, so far, the output of the encoder is Y_T (the encoder result) that has semantically acquired the key information of context and documents guided by the last utterance.

4.2 Next-utterance classification

In order to ensure that the background information learned by the encoder is useful, we take the encoder result out and concatenate it with the wrong reply or the golden reply separately plus a CLS token in the end for classification as follows.

$$In = [Y_T; Em(U_F)/Em(U_{T+1}); C] \quad (4)$$

where $Em(U_F)$ and $Em(U_{T+1})$ is the embedded false reply randomly sampling from the rest responses and the golden reply respectively (here the ratio of the number of U_{T+1} to the number of U_F is $\frac{1}{5}$) and C is the embedded CLS token. Then In is input into the aforementioned multi-head self-attention module ($MultiHead()$) and a linear transformation ($Linear()$) is build on the attention for classification, as Equation 5

$$Re = Linear(MultiHead(In, In, In)) \quad (5)$$

where Re is 2-D vector representing the probabilities of the distribution on True and False response. Note that only the hidden state of the CLS

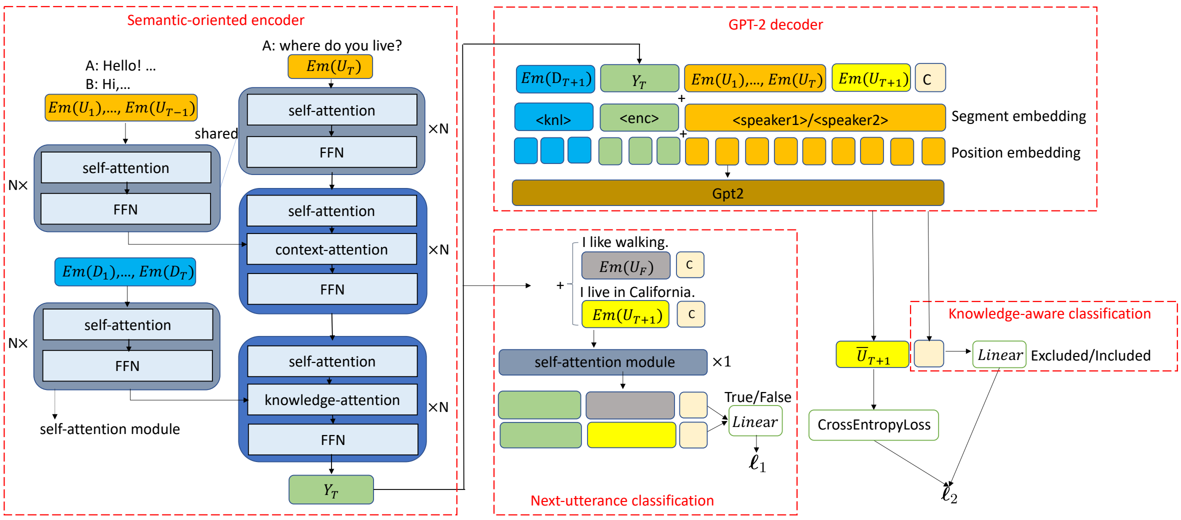


Figure 2: The overview of our model.

token is sent to the linear layer for classification task, which can capture whether the encoder result has learned the correct semantic meaning of dialogue context and corresponding knowledge. Although our next-utterance classification is simple but it successfully pushes the encoder to learn the background information very well.

4.3 GPT-2 decoder with knowledge-aware classification

Generally, existing methods ignore the truth that many utterances do not contain knowledge from given documents. They directly stuff the knowledge into the model and do not consider whether generated responses require the knowledge. Meanwhile, most available datasets for document-grounded dialogue rarely indicate whether the response includes the knowledge, and it is the reason why existing methods neglect this problem.

To achieve real knowledge-ware responses, we selected n daily and non-informative utterances from the datasets labelled as "excluded", other responses are marked as "included". Our purpose is not to encourage our model to generate knowledge-excluded responses but is to let our model to generate responses with knowledge at the right time.

For each utterance, we calculated the semantic similarity between the utterance and given documents (knowledge), then we also calculated the semantic similarities between the utterance and n selected knowledge-excluded responses.

$$score_{in} = sim(U_{T+1}, D_{T+1}) \qquad (6)$$

where U_{T+1} is the response to generate and $sim()$ is the cosine similarity function. D_{T+1} is the relevant document of U_{T+1}.

$$score_{ex} = \max_{1 \leq i \leq n} (sim(U_{T+1}, U_i)) \qquad (7)$$

where U_i is the i-th utterance of n selected utterances.

The labeling rule is shown in Equation 8, i.e., if an utterance is more similar with its relevant document than the most similar one among selected knowledge-excluded utterances, its tag is set to 1, otherwise, it is 0.

$$tag = \begin{cases} 0, & score_{ex} > score_{in}, \\ 1, & score_{ex} \leq score_{in}. \end{cases} \qquad (8)$$

where 0 is the tag for knowledge unused and 1 is for knowledge used. After labeling, the classification task is introduced to the GPT-2 decoder. Then a CLS token is added at the last position of the input of the decoder and finally its hidden state are input into the linear classifier which is same as the next-utterance linear classifier.

4.4 Training procedure

Unlike traditional encoder and decoder training together, we divide the training procedure into two stages. Firstly the encoder is trained at the first stage individually by using the next-utterance classification until the parameters converges. Then, the decoder is trained with the basis of the trained encoder at the second stage. Equation 9 shows the loss of the first stage.

$$\ell_1 = -\sum_{i=1}^{m} \log P(y_1^i | U_{\leq T}^i, R_f^i / U_{T+1}^i, D_{\leq T}^i) \qquad (9)$$

where i is the index of training examples and y_1^i is the labels of the i-th example.

$$\ell_2 = -\sum_{i=1}^{m'} (\lambda \log P(y_2^i | U_{\leq T}^i, D_{\leq T+1}^i, Y_T^i) + \sum_{j=1}^{|U_{T+1}^i|} \log P(w_{T+1,j}^i | w_{<j}^i, U_{\leq T+1}^i, D_{T+1}^i, Y_T^i)) \quad (10)$$

where ℓ_2 is the loss function of the second stage, λ is the hyper-parameter, y_2 is the label of the j example. The former/latter item of Equation 10 and is the classification/cross entropy loss.

5 Experiments

5.1 Dataset

We evaluate our model with CMU Document Grounded Conversations (CMU_DoG) and PERSONA-CHAT datasets. They are built upon crowd-sourcing where human conversations are based on given documents. The CMU_DoG dataset records the conversations between two persons who discuss the given movie document. The PERSONA-CHAT dataset contains multi-turn dialogues between two persons conditioned on artificial personas. Two datasets are downloaded from the URLs[1] [2]. Their statistics is shown in Table 1.

Statistics	CMU_DoG	PERSONA-CHAT
training	13541	16878
evaluation	780	1000
testing	2476	1000
#T/C	21.4	14.8
#W/U	18.6	11.2
#W/D	229	7.2

Table 1: Statistics of CMU_DoG and PERSONA-CHAT datasets. (training/evaluation/testing: the number of examples in training/evaluation/testing datasets; T/C is the average number of turns per conversation; W/U and W/D are the average lengths of utterances and given documents respectively.)

5.2 Baselines

We compare our model with the SOTA models: 1) **[TMN]:** A transformer-based dialogue model (Dinan et al., 2018) using given documents, the code is downloaded from the URL[3]; 2) **[ITDD:]**The model uses an incremental encoder and deliberation decoder (Li et al., 2019). We implement the model code from the URL[4]; 3) **[TransferTransfo:]** A model based on GPT-2 (Thomas et al., 2019) concatenates documents, dialogue context and responses into a sequences for generation. The code is available at the URL[5]; 4) **[DRD:]** With the shortage of the datasets of knowledge-grounded dialogues, the model (Zhao et al., 2019) isolates the parameters trained by knowledge-grounded dialogues from the pre-trained parameters for ungrounded dialogues and documents.

5.3 Evaluation Metrics

We compare the performance of all models with automatic and manual metrics:

Automatic Metrics: Following (Zhang et al., 2018b), Avglen and Entropy are used to measure response diversity, and Avglen is the average length of generated responses, i.e.,the number of tokens. BLEU (Papineni et al., 2002), Rouge-L (Lin, 2004), METEOR (Lavie and Agarwal, 2007), F1 measure (Dinan et al., 2018) and perplexity(PPL) (Bengio et al., 2003) are used to measure word level similarity between golden reply and reply generated from different perspectives. For evaluating the sentence-level performance, we use Embedding metrics (Liu et al., 2016) : Average, Extrema and Greedy (Liu et al., 2016; Serban et al., 2017), which they describe the semantic similarity between generated and golden responses.

Manual Metrics: since there is only one golden reply while dialogue answers are flexible (Liu et al., 2016; Tao et al., 2018), we introduce three manual metrics to evaluate generated answers from different angles. (1) **Fluency** evaluates generated responses in terms of naturalness and fluency; (2) **Knowledge Relevance** evaluates whether generated responses include the knowledge from documents or not; (3) **Knowledge Fitness** indicates that knowledge is selected based on dialogue context; five volunteers who are not involved in our work are given 300 examples for each dataset, and they need to choose the best answer for the 600 examples at each manual metric.To be fair, the model name of each response is hidden and the examples are randomly selected.

[1] https://github.com/festvox/datasets-CMU_DoG

[2] https://s3.amazonaws.com/datasets.huggingface.co/personachat/personachat_self_original.json

[3] https://github.com/facebookresearch/ParlAI/blob/master/projects/wizard_of_wikipedia

[4] https://github.com/lizekang/ITDD

[5] https://github.com/huggingface/transfer-learning-conv-ai

Model	ParamsNum	F1	PPL	Average	Extrema	Greedy	AvgLen	Entropy	Rouge-L	METEOR	BLEU
TMN	8×10^7	11.8	32.3	0.814	0.427	0.640	**12.9**	10.0	0.111	0.063	0.02
ITDD	11×10^7	9.9	26.0	0.765	0.383	0.592	11.7	10.6	0.117	0.051	0.02
TransferTransfo	12×10^7	12.4	15.7	0.823	0.430	0.649	10.0	**10.7**	0.115	0.058	0.02
DRD	-	10.8	46.1	0.791	0.406	0.613	-	-	-	-	-
KnowledGPT	24×10^7	13.5	20.6	**0.837**	0.437	0.654	-	-	-	-	-
DcDial	17×10^7	**14.6**	**16.0**	0.825	**0.448**	**0.657**	9.9	9.8	**0.146**	**0.071**	**0.03**

Table 2: The results of automatic evaluation on CMU_DoG dataset. ("-": no data in the paper (Zhao et al., 2020).)

Model	F1	PPL	Average	Extrema	Greedy	AvgLen	Entropy	Rouge-L	METEOR	BLEU
TMN (Dinan et al., 2018)	13.8	32.5	0.850	0.466	0.660	9.1	9.2	0.118	0.065	0.02
ITDD (Li et al., 2019)	13.8	22.1	0.849	0.480	0.663	8.9	7.4	0.146	0.063	0.03
TransferTransfo (Thomas et al., 2019)	13.2	14.3	0.833	0.462	0.651	**9.5**	**10.2**	0.117	0.068	0.02
DcDial	**18.7**	**13.8**	**0.868**	**0.500**	**0.685**	9.4	9.3	**0.161**	**0.091**	**0.04**

Table 3: The results of automatic evaluation on PERSONA-CHAT dataset.(DRD and KnowledGPT have no data)

5.4 Implementation Details

We implement our model based on the work of (Thomas et al., 2019) [6]. All models are trained using 20 epochs and tested on the server with three 2080ti GPUs. Our DcDial uses the Adam optimization method (Kingma and Ba, 2014) and GPT-2 (117M). Learning rate starts from 6.25e-5 and reduces linearly to 0 during training. The drop rate is 0.1. The number of transformer layers in the encoder is $N = 3$ and the number of transformer layers in the next-sentence classifier is 1. The hyper parameter λ of Equation 10 is 1. The training batch size at stage 1/2 is 33/3. The hidden dimension is 768 and the number of attention heads is 12.

5.5 Experimental Results

Table 2 and 3 shows the automatic evaluation results of baselines and our model on CMU_DoG and PERSONA-CHAT datasets. Except the PPL, the larger value indicates the better performance. Because KnowledGPT and DRD did not publish the training code, their metric values are directly quoted from the original paper (Zhao et al., 2020).

5.5.1 Automatic evaluation

From the two tables, we have two observations: one is that all models based on GPT-2 have better results than the rest ones; the other is that our model performs the best on most metrics. For the first observation, the performance of ITDD and TransferTransfo has an obvious gap even if the numbers of their parameters. It demonstrates high-frequency patterns leaned by GPT-2 are very helpful for improving the generation capability. For the second observation, we first compare our model with TransferTransfo. In Table 3, our model significantly outperforms TransferTransfo at all metrics except "AvgLen" and "Entropy". It proves that generated responses of our model contain more identical words and similar semantics with the ground truth based on the learned low-frequency patterns. Note that If our background learning and information selection do not work, our model will degenerate into (or even worse than) the TransferTransfo model. For "AvgLen" and "Entropy", a very likely reason is that our encoder 'filters out' much irrelevant information from dialogue context and documents and it could reduce the diversity of generation. Table 2 has similar results with Table 3 but is slightly worse. It is probably caused by that the knowledge documents of CMU_DoG are much bigger than the documents for describing persons (see Table 1). It reduces the performance of our task-specific architectures and our model degenerates into TransferTransfo.

There is an interesting phenomenon, e.g. the performance of KnowledGPT is between ours and TransferTransfo. In Table 2, our model/KnowledGPT wins 4 out of 5 metrics compared with KnowledGPT/TransferTransfo. As we mentioned, KnowledGPT has an extra knowledge-selection module based on BERT. Although the module is to tailor given knowledge documents to meet the length constraint for a GPT-2 model or even shorter, fewer documents reduce the complex of the learning of GPT2. More importantly, the selection process utilizes dialogue context to rank related document, which works like our background pattern at coarse-grained level.

5.5.2 Ablation study

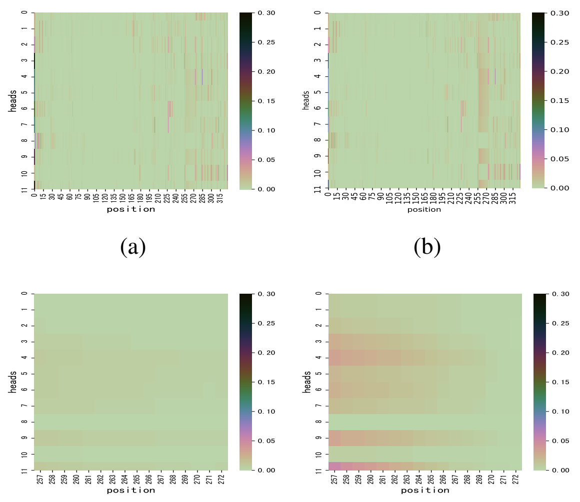
Here we remove the knowledge-aware classification, the encoder and the two-stage training procedure respectively to to verify their contribution. Table 5 is the result of ablation experiment and we have the following observations: 1) Removing the

[6] The code of our model and metrics are open source at `https://github.com/hanying980919/DcDial`.

Model	F1	PPL	Average	Extrema	Greedy	AvgLen	Entropy	rouge-l	meteor	BLEU
DcDial	14.6	16.0	0.825	0.448	0.657	9.9	9.8	0.146	0.071	0.03
-classification	14.0	16.2	0.820	0.445	0.652	11.0	9.9	0.139	0.065	0.03
-encoder	13.1	16.8	0.819	0.435	0.649	10.7	9.7	0.128	0.063	0.03
-stage	12.0	17.0	0.814	0.428	0.641	10.0	10.2	0.117	0.052	0.02

Table 4: Ablation study on CMU_DoG (-classification: remove the classification in decoder; -stage: train the encoder and decoder together instead of separate training;-encoder: remove the encoder.).

knowledge-aware classification (-classification) in decoder leads to an apparently worse results. Without the module, our model could wrongly introduce more knowledge into generated responses whose ground truth do not include knowledge from given documents. 2) Cutting off the encoder (-encoder) significantly reduce the performance of our model, the F1 and Rough-L metrics drop more than 11%. 3) Stopping two-stage training (-stage) results in the greatest decline in most of metrics. The maximum drop can reach around 20% since the vanishing phenomenon makes the output of the encoder noise. All observations shows that our specific-task architectures can indeed improve the performance of document-grounded dialogue.



(a) (b) (c) (d)

Figure 3: Heat map of attention distributions.

5.5.3 Visualization

Here we visualized the attention distribution of the last layer of the decoder when generating the last word. Figure 3 is the heat map of attention distribution on each input of the decoder. In this case, x-axis is the position ID and y-axis is 12 head attentions. The position IDs from 0 to 256 represent the embeddings of the relevant document D_{T+1}, the position IDs of the encoder result Y_T is from 257 to 272, and the embeddings of dialogue context sit in the position IDs from 273 to the end. Subfigure (a)/(b) shows the attention distribution with training together/separately. Subfigure (c)/(d) is the encoder results of Subfigure(a)/(b). Here we can find that 1) the position IDs of the encoder result in sub-figure(c) has much less weights (very light red), which shows the vanishing phenomenon; 2) the position IDs of the encoder result in sub-figure(b) has more weights than others (darker red), which prove the advantages of two-stage training.

5.5.4 Human evaluation

Table 5 is the voting results in terms of three aspects. For Fluency, the results of TransferTransfo and DcDial models are comparable and outperform others since both models are based on the GPT-2 model, which has advantages in language mode learning. For Knowledge Relevance, the votes of TransferTransfo is accounted for 33% and more than ours (30%). The reason is that, compared to our model, TransferTransfo do not learn the knowledge-aware classification and tends to insert more knowledge into generated responses. For Knowledge Fitness, our model has better performance than others benefit from the semantic-oriented encoder and the large-scale GPT-2.

	Fluency	Knowledge Relevance	Knowledge Fitness
TMN	12%	17%	18%
TransferTransfo	30%	**33%**	22%
ITDD	27%	20%	25%
DcDial	**31%**	30%	**35%**

Table 5: The result of human evaluation.

6 Conclusion

In this paper, we proposed a semantic-oriented knowledge-aware model (DcDial) for document-grounded dialogue generation. Through the semantic-oriented encoder with utterance prediction, the model can learn the specific low-frequency for accurately capturing background information. Meanwhile, the GPT-2 decoder with the knowledge prediction can implement real knowledge-aware dialogue generation. Empirical results show that our model can generate responses with much more coherence and knowledge-filled compared with the state-of-the-art baselines.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In ICLR.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. The journal of machine learning research, 3:1137–1155.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In CoNLL, pages 10–21. ACL.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, and Sam McCandlish. 2020. Language models are few-shot learners. arXiv preprint arXiv:2005.14165.

Hongshen Chen, Zhaochun Ren, Jiliang Tang, Yihong Eric Zhao, and Dawei Yin. 2018. Hierarchical variational memory network for dialogue generation. In WWW, pages 1653–1662. ACM.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does bert look at? an analysis of bert's attention. In ACL, pages 276–286.

Joe Davison, Joshua Feldman, and Alexander M. Rush. 2019. Commonsense knowledge mining from pretrained models. In EMNLP-IJCNLP, pages 1173–1178.

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

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of wikipedia: Knowledge-powered conversational agents. In ICLR.

Hao Fu, Chunyuan Li, Xiaodong Liu, Jianfeng Gao, Asli Celikyilmaz, and Lawrence Carin. 2019. A simple approach to mitigating kl vanishing. In NAACL, volume 1, pages 240–250.

Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. 2018. A knowledge-grounded neural conversation model. In AAAI, pages 5110–5117. AAAI Press.

John Hewitt and Christopher D. Manning. 2019. A structural probe for find- ing syntax in word representations. In NAACL, pages 4129–4138.

Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. 2020. Challenges in building intelligent open-domain dialog systems. TOIS, 38(3):1–32.

Shiro Ikeda, Shun ichi Amari, and Hiroyuki Nakahara. 1999. Convergence of the wake-sleep algorithm. In NIPS, pages 239–245.

Byeongchang Kim, Jaewoo Ahn, and Gunhee Kim. 2019. Sequential latent knowledge selection for knowledge-grounded dialogue. In ICLR.

Sihyung Kim, Oh-Woog Kwon, and Harksoo Kim. 2020. Knowledge-grounded chatbot based on dual wasserstein generative adversarial networks with effective attention mechanisms. Applied Science, 10(9):1–11.

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

Alon Lavie and Abhaya Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In the second workshop on statistical machine translation, pages 228–231.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pretraining for natural language generation, translation, and comprehension. In ACL, pages 7871–7880.

Zekang Li, Cheng Niu, Fandong Meng, Yang Feng, Qian Li, and Jie Zhou. 2019. Incremental transformer with deliberation decoder for document grounded conversations. In ACL, pages 12–21.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In Text summarization branches out, pages 74–81.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In EMNLP, pages 2122–2132.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. Gpt understands, too. arXiv preprint arXiv:2103.10385.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In ACL, pages 311–318.

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020. How context affects language models' factual predictions. In Proceedings of InAutomated Knowledge Base Construction.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. OpenAI blog, 1(8).

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In ACL, pages 1073–1083.

Iulian Vlad Serban, Alessandro Sordoni, and el al. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In AAAI, pages 3776–3784.

Iulian Vlad Serban, Alessandro Sordoni, and et al. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In AAAI, pages 3295–3301. AAAI Press.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. Eliciting knowledge from language models using automatically generated prompts. In EMNLP, pages 4222–4235.

Xiangru Tang and Po Hu. 2019. Knowledge-aware self-attention networks for document grounded dialogue generation. In In International Conference on Knowledge Science, Engineering and Management, pages 400–411.

Chongyang Tao, Lili Mou, Dongyan Zhao, and Rui Yan. 2018. Ruber: An unsupervised method for automatic evaluation of open-domain dialog systems. In AAAI, pages 722–729.

Wolf Thomas, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019. Transfertransfo: A transfer learning approach for neural network based conversational agents. arXiv preprint arXiv:1901.08149.

Zhiliang Tian, Rui Yan, Lili Mou, Yiping Song, Yansong Feng, and Dongyan Zhao. 2017. How to make context more useful? an empirical study on context-aware neural conversational models. In ACL, pages 231–236. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, and Łukasz Kaiser. 2017. Attention is all you need. arXiv preprint arXiv:1706.03762.

Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. In ACL, pages 37–42.

Oriol Vinyals and Quoc V. Le. 2015. A neural conversational model. In ICML.

Chenguang Wang, Xiao Liu, and Dawn Song. 2020. Language models are open knowledge graphs. arXiv preprint arXiv:2010.11967.

Yan Wang, Yanan Zheng, Shimin Jiang, Yucheng Dong, Jessica Chen, and Shaohua Wan. 2021. Generating contextually coherent responses by learning structured vectorized semantics. In DASFAA, 70-86.

Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2017. Topic aware neural response generation. In AAAI, pages 3351–3357. AAAI Press.

Hao Tong Ye, Kai Ling Lo, Shang Yu Su, and Yun Nung Chen. 2020. Knowledge-grounded response generation with deep attentional latent-variable model. Computer Speech and Language.

Weinan Zhang, Yiming Cui, Yifa Wang, Qingfu Zhu, Lingzhi Li, and et al. 2018a. Context-sensitive generation of open-domain conversational responses. In COLING, pages 2437–2447.

Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. 2018b. Generating informative and diverse conversational responses via adversarial information maximization. In NeurIPS, pages 1815–1825.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. Zhang, yizhe, et al. "dialogpt: Large-scale generative pre-training for conversational response generation. In ACL, pages 270–278.

Xueliang Zhao, Wei Wu, Chongyang Tao, Can Xu, Dongyan Zhao, and Rui Yan. 2019. Low-resource knowledge-grounded dialogue generation. In ICLR.

Xueliang Zhao, Wei Wu, Can Xu, Chongyang Tao, Dongyan Zhao, and Rui Yan. 2020. Knowledge-grounded dialogue generation with pre-trained language models. In EMNLP, pages 3377–3390.

Chujie Zheng, Yunbo Cao, Daxin Jiang, and Minlie Huang. 2020a. Difference-aware knowledge selection for knowledge-grounded conversation generation. In EMNLP, 115-125.

Yanan Zheng, Yan Wang, Lijie Wen, and Jianmin Wang. 2020b. A latent-constrained variational neural dialogue model for information-rich responses. In CIKM, pages 1351–1360.

Kangyan Zhou, Shrimai Prabhumoye, and Alan W Black. 2018. A dataset for document grounded conversations. In EMNLP, pages 708–713.