Incremental Transformer with Deliberation Decoder for Document Grounded Conversations

Zekang Li[†◇], Cheng Niu[‡], Fandong Meng[‡*], Yang Feng[◇], Qian Li[♠], Jie Zhou[‡]
[†]Dian Group, School of Electronic Information and Communications
Huazhong University of Science and Technology
[‡]Pattern Recognition Center, WeChat AI, Tencent Inc, China
[◇]Key Laboratory of Intelligent Information Processing
Institute of Computing Technology, Chinese Academy of Sciences
[♠]School of Computer Science and Engineering, Northeastern University, China
zekangli97@gmail.com, {chengniu,fandongmeng,jiezhou}@tencent.com
fengyang@ict.ac.cn, qianli@stumail.neu.edu.cn

Abstract

Document Grounded Conversations is a task to generate dialogue responses when chatting about the content of a given document. Obviously, document knowledge plays a critical role in Document Grounded Conversations, while existing dialogue models do not exploit this kind of knowledge effectively enough. In this paper, we propose a novel Transformer-based architecture for multi-turn document grounded conversations. In particular, we devise an Incremental Transformer to encode multi-turn utterances along with knowledge in related documents. Motivated by the human cognitive process, we design a two-pass decoder (Deliberation Decoder) to improve context coherence and knowledge correctness. Our empirical study on a real-world Document Grounded Dataset proves that responses generated by our model significantly outperform competitive baselines on both context coherence and knowledge relevance.

1 Introduction

Past few years have witnessed the rapid development of dialogue systems. Based on the sequence-to-sequence framework (Sutskever et al., 2014), most models are trained in an end-to-end manner with large corpora of human-to-human dialogues and have obtained impressive success (Shang et al., 2015; Vinyals and Le, 2015; Li et al., 2016; Serban et al., 2016). While there is still a long way for reaching the ultimate goal of dialogue systems, which is to be able to talk like humans. And one of the essential intelligence to achieve this goal is the ability to make use of knowledge.

There are several works on dialogue systems exploiting knowledge. The Mem2Seq (Madotto et al., 2018) incorporates structured knowledge into the end-to-end task-oriented dialogue. Liu et al. (2018) introduces fact-matching and knowledge-diffusion to generate meaningful, diverse and natural responses using structured knowledge triplets. Ghazvininejad et al. (2018), Parthasarathi and Pineau (2018), Yavuz et al. (2018), Dinan et al. (2018) and Lo and Chen (2019) apply unstructured text facts in open-domain dialogue systems. These works mainly focus on integrating factoid knowledge into dialogue systems, while factoid knowledge requires a lot of work to build up, and is only limited to expressing precise facts. Documents as a knowledge source provide a wide spectrum of knowledge, including but not limited to factoid, event updates, subjective opinion, etc. Recently, intensive research has been applied on using documents as knowledge sources for Question-Answering (Chen et al., 2017; Huang et al., 2018; Yu et al., 2018; Rajpurkar et al., 2018; Reddy et al., 2018).

The Document Grounded Conversation is a task to generate natural dialogue responses when chatting about the content of a specific document. This task requires to integrate document knowledge with the multi-turn dialogue history. Different from previous knowledge grounded dialogue systems, Document Grounded Conversations utilize documents as the knowledge source, and hence are able to employ a wide spectrum of knowledge. And the Document Grounded Conversations is also different from document QA since the contextual consistent conversation response should be generated. To address the Document Grounded Conversation task, it is important to: 1) Exploit document knowledge which are relevant to the

[*] Fandong Meng is the corresponding author of the paper. This work was done when Zekang Li was interning at Pattern Recognition Center, WeChat AI, Tencent.

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 12–21
Florence, Italy, July 28 - August 2, 2019.

conversation; 2) Develop a unified representation combining multi-turn utterances along with the relevant document knowledge.

In this paper, we propose a novel and effective Transformer-based (Vaswani et al., 2017) architecture for Document Grounded Conversations, named Incremental Transformer with Deliberation Decoder. The encoder employs a transformer architecture to incrementally encode multi-turn history utterances, and incorporate document knowledge into the the multi-turn context encoding process. The decoder is a two-pass decoder similar to the Deliberation Network in Neural Machine Translation (Xia et al., 2017), which is designed to improve the context coherence and knowledge correctness of the responses. The first-pass decoder focuses on contextual coherence, while the second-pass decoder refines the result of the first-pass decoder by consulting the relevant document knowledge, and hence increases the knowledge relevance and correctness. This is motivated by human cognition process. In real-world human conversations, people usually first make a draft on how to respond the previous utterance, and then consummate the answer or even raise questions by consulting background knowledge.

We test the effectiveness of our proposed model on Document Grounded Conversations Dataset (Zhou et al., 2018). Experiment results show that our model is capable of generating responses of more context coherence and knowledge relevance. Sometimes document knowledge is even well used to guide the following conversations. Both automatic and manual evaluations show that our model substantially outperforms the competitive baselines.

Our contributions are as follows:

- We build a novel Incremental Transformer to incrementally encode multi-turn utterances with document knowledge together.

- We are the first to apply a two-pass decoder to generate responses for document grounded conversations. Two decoders focus on context coherence and knowledge correctness respectively.

2 Approach

2.1 Problem Statement

Our goal is to incorporate the relevant document knowledge into multi-turn conversations. Formally, let $\mathbf{U} = \mathbf{u}^{(1)}, ..., \mathbf{u}^{(k)}, ..., \mathbf{u}^{(K)}$ be a whole conversation composed of K utterances. We use $\mathbf{u}^{(k)} = u_1^{(k)}, ..., u_i^{(k)}, ..., u_I^{(k)}$ to denote the k-th utterance containing I words, where $u_i^{(k)}$ denotes the i-th word in the k-th utterance. For each utterance $\mathbf{u}^{(k)}$, likewise, there is a specified relevant document $\mathbf{s}^{(k)} = s_1^{(k)}, ..., s_j^{(k)}, ..., s_J^{(k)}$, which represents the document related to the k-th utterance containing J words. We define the document grounded conversations task as generating a response $\mathbf{u}^{(k+1)}$ given its related document $\mathbf{s}^{(k+1)}$ and previous k utterances $\mathbf{U}_{\leq k}$ with related documents $\mathbf{S}_{\leq k}$, where $\mathbf{U}_{\leq k} = \mathbf{u}^{(1)}, ..., \mathbf{u}^{(k)}$ and $\mathbf{S}_{\leq k} = \mathbf{s}^{(1)}, ..., \mathbf{s}^{(k)}$. Note that $\mathbf{s}^{(k)}, \mathbf{s}^{(k+1)}, ..., \mathbf{s}^{(k+n)}$ may be the same.

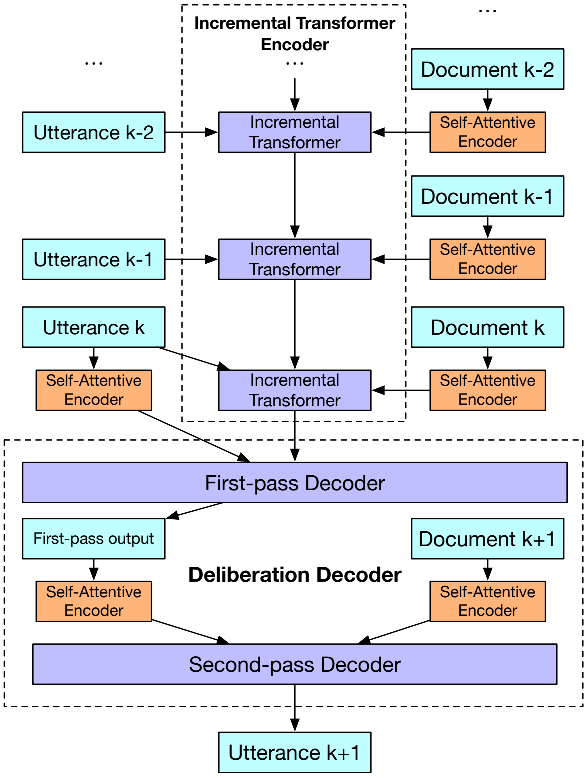

Figure 1: The framework of Incremental Transformer with Deliberation Decoder for Document Grounded Conversations.

Therefore, the probability to generate the response $\mathbf{u}^{(k+1)}$ is computed as:

$$\begin{aligned} &P(\mathbf{u}^{(k+1)}|\mathbf{U}_{\leq k}, \mathbf{S}_{\leq k+1}; \theta) \\ &= \textstyle\prod_{i=1}^{I} P(u_i^{k+1}|\mathbf{U}_{\leq k}, \mathbf{S}_{\leq k+1}, u_{<i}^{(k+1)}; \theta) \end{aligned} \tag{1}$$

where $u_{<i}^{(k+1)} = u_1^{(k+1)}, ..., u_{i-1}^{(k+1)}$.

2.2 Model Description

Figure 1 shows the framework of the proposed Incremental Transformer with Deliberation De-

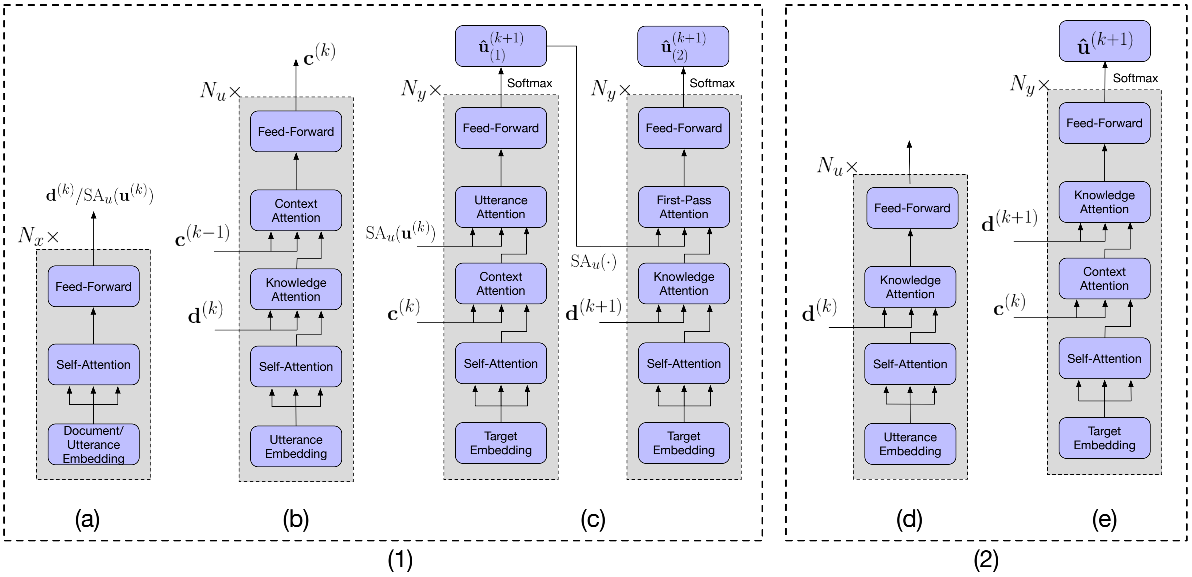

Figure 2: (1) Detailed architecture of model components. (a) The Self-Attentive Encoder(SA). (b) Incremental Transformer (ITE). (c) Deliberation Decoder (DD). (2) Simplified version of our proposed model used to verify the validity of our proposed Incremental Transformer Encoder and Deliberation Decoder. (d) Knowledge-Attention Transformer(KAT). (e) Context-Knowledge-Attention Decoder (CKAD).

coder. Please refer to Figure 2 (1) for more details. It consists of three components:

1) Self-Attentive Encoder (SA) (in orange) is a transformer encoder as described in (Vaswani et al., 2017), which encodes the document knowledge and the current utterance independently.

2) Incremental Transformer Encoder (ITE) (on the top) is a unified transformer encoder which encodes multi-turn utterances with knowledge representation using an incremental encoding scheme. This module takes previous utterances $\mathbf{u}^{(i)}$ and the document $\mathbf{s}^{(i)}$'s SA representation as input, and use attention mechanism to incrementally build up the representation of relevant context and document knowledge.

3) Deliberation Decoder (DD) (on the bottom) is a two-pass unified transformer decoder for better generating the next response. The first-pass decoder takes current utterance $\mathbf{u}^{(k)}$'s SA representation and ITE output as input, and mainly relies on conversation context for response generation. The second-pass decoder takes the SA representation of the first pass result and the relevant document $\mathbf{s}^{(k+1)}$'s SA representation as input, and uses document knowledge to further refine the response.

Self-Attentive Encoder

As document knowledge often includes several sentences, it's important to capture long-range dependencies and identify relevant information. We use multi-head self-attention (Vaswani et al., 2017) to compute the representation of document knowledge.

As shown in Figure 2 (a), we use a self-attentive encoder to compute the representation of the related document knowledge $\mathbf{s}^{(k)}$. The input ($\mathbf{In}_s^{(k)}$) of the encoder is a sequence of document words embedding with positional encoding added.(Vaswani et al., 2017):

$$\mathbf{In}_s^{(k)} = [\mathbf{s}_1^{(k)}, ..., \mathbf{s}_J^{(k)}] \qquad (2)$$

$$\mathbf{s}_j^{(k)} = \mathbf{e}_{s_j} + \mathrm{PE}(j) \qquad (3)$$

where $\mathbf{e}_{s_j}$ is the word embedding of $s_j^{(k)}$ and $\mathrm{PE}(\cdot)$ denotes positional encoding function.

The Self-Attentive encoder contains a stack of N_x identical layers. Each layer has two sublayers. The first sub-layer is a multi-head self-attention (MultiHead) (Vaswani et al., 2017). $\mathrm{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ is a multi-head attention function that takes a query matrix $\mathbf{Q}$, a key matrix $\mathbf{K}$, and a value matrix $\mathbf{V}$ as input. In current case, $\mathbf{Q} = \mathbf{K} = \mathbf{V}$. That's why it's called self-attention. And the second sub-layer is a simple, position-wise fully connected feed-forward network (FFN). This FFN consists of two linear transformations with a ReLU activation in between. (Vaswani et al., 2017).

$$\mathbf{A}^{(1)} = \mathrm{MultiHead}(\mathbf{In}_s^{(k)}, \mathbf{In}_s^{(k)}, \mathbf{In}_s^{(k)}) \qquad (4)$$

$$\mathbf{D}^{(1)} = \mathrm{FFN}(\mathbf{A}^{(1)}) \qquad (5)$$

$$\mathrm{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \qquad (6)$$

where $\mathbf{A}^{(1)}$ is the hidden state computed by multi-head attention at the first layer, $\mathbf{D}^{(1)}$ denotes the representation of $\mathbf{s}^{(k)}$ after the first layer. Note that residual connection and layer normalization are used in each sub-layer, which are omitted in the presentation for simplicity. Please refer to (Vaswani et al., 2017) for more details.

For each layer, repeat this process:

$$\mathbf{A}^{(n)} = \mathrm{MultiHead}(\mathbf{D}^{(n-1)}, \mathbf{D}^{(n-1)}, \mathbf{D}^{(n-1)}) \tag{7}$$

$$\mathbf{D}^{(n)} = \mathrm{FFN}(\mathbf{A}^{(n)}) \tag{8}$$

where $n = 1, ..., N_s$ and $\mathbf{D}^{(0)} = \mathbf{In}_s^{(k)}$.

We use $\mathrm{SA}_s(\cdot)$ to denote this whole process:

$$\mathbf{d}^{(k)} = \mathbf{D}^{(N_x)} = \mathrm{SA}_s(\mathbf{s}^{(k)}) \tag{9}$$

where $\mathbf{d}^{(k)}$ is the final representation for the document knowledge $\mathbf{s}^{(k)}$.

Similarly, for each utterance $\mathbf{u}^{(k)}$, we use $\mathbf{In}_u^{(k)} = [\mathbf{u}_1^{(k)}, ..., \mathbf{u}_I^{(k)}]$ to represent the sequence of the position-aware word embedding. Then the same Self-Attentive Encoder is used to compute the representation of current utterance $\mathbf{u}^{(k)}$, and we use $\mathrm{SA}_u(\mathbf{u}^{(k)})$ to denote this encoding result. The Self-Attentive Encoder is also used to encode the document $\mathbf{s}^{(k+1)}$ and the first pass decoding results in the second pass of the decoder. Note that SA_s and SA_u have the same architecture but different parameters. More details about this will be mentioned in the following sections.

Incremental Transformer Encoder

To encode multi-turn document grounded utterances effectively, we design an Incremental Transformer Encoder. Incremental Transformer uses multi-head attention to incorporate document knowledge and context into the current utterance's encoding process. This process can be stated recursively as follows:

$$\mathbf{c}^{(k)} = \mathrm{ITE}(\mathbf{c}^{(k-1)}, \mathbf{d}^{(k)}, \mathbf{In}_u^{(k)}) \tag{10}$$

where $\mathrm{ITE}(\cdot)$ denotes the encoding function, $\mathbf{c}^{(k)}$ denotes the context state after encoding utterance $\mathbf{u}^{(k)}$, $\mathbf{c}^{(k-1)}$ is the context state after encoding last utterance $\mathbf{u}^{(k-1)}$, $\mathbf{d}^{(k)}$ is the representation of document $\mathbf{s}^{(k)}$ and $\mathbf{In}_u^{(k)}$ is the embedding of current utterance $\mathbf{u}^{(k)}$.

As shown in Figure 2 (b), we use a stack of N_u identical layers to encode $\mathbf{u}^{(k)}$. Each layer consists of four sub-layers. The first sub-layer is a multi-head self-attention:

$$\mathbf{B}^{(n)} = \mathrm{MultiHead}(\mathbf{C}^{(n-1)}, \mathbf{C}^{(n-1)}, \mathbf{C}^{(n-1)}) \tag{11}$$

where $n = 1, ..., N_u$, $\mathbf{C}^{(n-1)}$ is the output of the last layer and $\mathbf{C}^{(0)} = \mathbf{In}_u^{(k)}$. The second sub-layer is a multi-head knowledge attention:

$$\mathbf{E}^{(n)} = \mathrm{MultiHead}(\mathbf{B}^{(n)}, \mathbf{d}^{(k)}, \mathbf{d}^{(k)}) \tag{12}$$

The third sub-layer is a multi-head context attention:

$$\mathbf{F}^{(n)} = \mathrm{MultiHead}(\mathbf{E}^{(n)}, \mathbf{c}^{(k-1)}, \mathbf{c}^{(k-1)}) \tag{13}$$

where $\mathbf{c}^{(k-1)}$ is the representation of the previous utterances. That's why we called the encoder ”Incremental Transformer”. The fourth sub-layer is a position-wise fully connected feed-forward network:

$$\mathbf{C}^{(n)} = \mathrm{FFN}(\mathbf{F}^{(n)}) \tag{14}$$

We use $\mathbf{c}^{(k)}$ to denote the final representation at N_u-th layer:

$$\mathbf{c}^{(k)} = \mathbf{C}^{(N_u)} \tag{15}$$

Deliberation Decoder

Motivated by the real-world human cognitive process, we design a Deliberation Decoder containing two decoding passes to improve the knowledge relevance and context coherence. The first-pass decoder takes the representation of current utterance $\mathrm{SA}_u(\mathbf{u}^{(\mathbf{k})})$ and context $\mathbf{c}^{(k)}$ as input and focuses on how to generate responses contextual coherently. The second-pass decoder takes the representation of the first-pass decoding results and related document $\mathbf{s}^{(k+1)}$ as input and focuses on increasing knowledge usage and guiding the following conversations within the scope of the given document.

When generating the i-th response word $u_i^{(k+1)}$, we have the generated words $\mathbf{u}_{<i}^{(k+1)}$ as input (Vaswani et al., 2017). We use $\mathbf{In}_r^{(k+1)}$ to denote the matrix representation of $\mathbf{u}_{<i}^{(k+1)}$ as following:

$$\mathbf{In}_r^{(k+1)} = [\mathbf{u}_0^{(k+1)}, \mathbf{u}_1^{(k+1)}, ..., \mathbf{u}_{i-1}^{(k+1)}] \tag{16}$$

where $\mathbf{u}_0^{(k+1)}$ is the vector representation of sentence-start token.

As shown in Figure 2 (c), the Deliberation Decoder consists of a first-pass decoder and a second-pass decoder. These two decoders have the same architecture but different input for sub-layers. Both decoders are composed of a stack of N_y identical layers. Each layer has four sub-layers. For the first-pass decoder, the first sub-layer is a multi-head self-attention:

$$\mathbf{G}_1^{(n)} = \mathrm{MultiHead}(\mathbf{R}_1^{(n-1)}, \mathbf{R}_1^{(n-1)}, \mathbf{R}_1^{(n-1)}) \tag{17}$$

where $n = 1, ..., N_y$, $\mathbf{R}_1^{(n-1)}$ is the output of the previous layer, and $\mathbf{R}_1^{(0)} = \mathbf{In}_r^{(k+1)}$. The second sub-layer is a multi-head context attention:

$$\mathbf{H}_1^{(n)} = \text{MultiHead}(\mathbf{G}_1^{(n)}, \mathbf{c}^{(k)}, \mathbf{c}^{(k)}) \tag{18}$$

where $\mathbf{c}^{(k)}$ is the representation of context $\mathbf{u}_{\leq k}$. The third sub-layer is a multi-head utterance attention:

$$\mathbf{M}_1^{(n)} = \text{MultiHead}(\mathbf{H}_1^{(n)}, \quad \text{SA}_u(\mathbf{u}^{(k)}), \text{SA}_u(\mathbf{u}^{(k)})) \tag{19}$$

where $SA_u(\cdot)$ is a Self-Attentive Encoder which encodes latest utterance $\mathbf{u}^{(k)}$. Eq. (18) mainly encodes the context and document knowledge relevant to the latest utterance, while Eq. (19) encodes the latest utterance directly. We hope optimal performance can be achieved by combining both.

The fourth sub-layer is a position-wise fully connected feed-forward network:

$$\mathbf{R}_1^{(n)} = \text{FFN}(\mathbf{M}_1^{(n)}) \tag{20}$$

After N_y layers, we use softmax to get the words probabilities decoded by first-pass decoder:

$$P(\hat{\mathbf{u}}_{(1)}^{(k+1)}) = \text{softmax}(\mathbf{R}_1^{(N_y)}) \tag{21}$$

where $\hat{\mathbf{u}}_{(1)}^{(k+1)}$ is the response decoded by the first-pass decoder. For second-pass decoder:

$$\mathbf{G}_2^{(n)} = \text{MultiHead}(\mathbf{R}_2^{(n-1)}, \mathbf{R}_2^{(n-1)}, \mathbf{R}_2^{(n-1)}) \tag{22}$$

$$\mathbf{H}_2^{(n)} = \text{MultiHead}(\mathbf{G}_2^{(n)}, \mathbf{d}^{(k+1)}, \mathbf{d}^{(k+1)}) \tag{23}$$

$$\mathbf{M}_2^{(n)} = \text{MultiHead}(\mathbf{H}_2^{(n)}, \quad \text{SA}_u(\hat{\mathbf{u}}_{(1)}^{(k+1)}), \text{SA}_u(\hat{\mathbf{u}}_{(1)}^{(k+1)})) \tag{24}$$

$$\mathbf{R}_2^{(n)} = \text{FFN}(\mathbf{M}_2^{(n)}) \tag{25}$$

$$P(\hat{\mathbf{u}}_{(2)}^{(k+1)}) = \text{softmax}(\mathbf{R}_2^{(N_y)}) \tag{26}$$

where $\mathbf{R}_2^{(n-1)}$ is the counterpart to $\mathbf{R}_1^{(n-1)}$ in pass two decoder, referring to the output of the previous layer. $\mathbf{d}^{(k+1)}$ is the representation of document $\mathbf{s}^{(k+1)}$ using Self-Attentive Encoder, $\hat{\mathbf{u}}_{(2)}^{(k+1)}$ is the output words after the second-pass decoder.

Training

In contrast to the original Deliberation Network (Xia et al., 2017), where they propose a complex joint learning framework using Monte Carlo Method, we minimize the following loss as Xiong et al. (2018) do:

$$L_{mle} = L_{mle1} + L_{mle2} \tag{27}$$

$$L_{mle1} = -\sum_{k=1}^{K}\sum_{i=1}^{I} \log P(\hat{\mathbf{u}}_{(1)i}^{(k+1)}) \tag{28}$$

$$L_{mle2} = -\sum_{k=1}^{K}\sum_{i=1}^{I} \log P(\hat{\mathbf{u}}_{(2)i}^{(k+1)}) \tag{29}$$

3 Experiments

3.1 Dataset

We evaluate our model using the Document Grounded Conversations Dataset (Zhou et al., 2018). There are 72922 utterances for training, 3626 utterances for validation and 11577 utterances for testing. The utterances can be either casual chats or document grounded. Note that we consider consequent utterances of the same person as one utterance. For example, we consider *A: Hello! B: Hi! B: How's it going?* as *A: Hello! B: Hi! How's it going?*. And there is a related document given for every several consequent utterances, which may contain movie name, casts, introduction, ratings, and some scenes. The average length of documents is about 200. Please refer to (Zhou et al., 2018) for more details.

3.2 Baselines

We compare our proposed model with the following state-of-the-art baselines:

Models not using document knowledge:

Seq2Seq: A simple encoder-decoder model (Shang et al., 2015; Vinyals and Le, 2015) with global attention (Luong et al., 2015). We concatenate utterances context to a long sentence as input.

HRED: A hierarchical encoder-decoder model (Serban et al., 2016), which is composed of a word-level LSTM for each sentence and a sentence-level LSTM connecting utterances.

Transformer: The state-of-the-art NMT model based on multi-head attention (Vaswani et al., 2017). We concatenate utterances context to a long sentence as its input.

Models using document knowledge:

Seq2Seq (+knowledge) and **HRED (+knowledge)** are based on Seq2Seq and HRED respectively. They both concatenate document knowledge representation and last decoding output embedding as input when decoding. Please refer to (Zhou et al., 2018) for more details.

Wizard Transformer: A Transformer-based model for multi-turn open-domain dialogue with unstructured text facts (Dinan et al., 2018). It concatenates context utterances and text facts to a long

Model	PPL	BLEU(%)	Fluency	Knowledge Relevance	Context Coherence
Seq2Seq without knowledge	80.93	0.38	1.62	0.18	0.54
HRED without knowledge	80.84	0.43	1.25	0.18	0.30
Transformer without knowledge	87.32	0.36	1.60	0.29	0.67
Seq2Seq (+knowledge)	78.47	0.39	1.50	0.22	0.61
HRED (+knowledge)	79.12	0.77	1.56	0.35	0.47
Wizard Transformer	70.30	0.66	1.62	0.47	0.56
ITE+DD (ours)	**15.11**	**0.95**	1.67	**0.56**	**0.90**
ITE+CKAD (ours)	64.97	0.86	**1.68**	0.50	0.82
KAT (ours)	65.36	0.58	1.58	0.33	0.78

Table 1: Automatic evaluation and manual evaluation results for baselines and our proposed models.

Model	Knowledge Relevance(%)	Context Coherence(%)
Wizard	64/25/11	58/28/14
ITE+CKAD	67/16/17	40/37/23
ITE+DD	64/16/20	38/34/28

Table 2: The percent(%) of score (0/1/2) of Knowledge Relevance and Context Coherence for Wizard Transformer, ITE+CKAD and ITE+DD.

sequence as input. We replace the text facts with document knowledge.

Here, we also conduct an ablation study to illustrate the validity of our proposed Incremental Transformer Encoder and Deliberation Decoder.

ITE+CKAD: It uses Incremental Transformer Encoder (ITE) as encoder and Context-Knowledge-Attention Decoder (CKAD) as shown in Figure 2 (e). This setup is to test the validity of the deliberation decoder.

Knowledge-Attention Transformer (KAT): As shown in Figure 2 (d), the encoder of this model is a simplified version of Incremental Transformer Encoder (ITE), which doesn't have context-attention sub-layer. We concatenate utterances context to a long sentence as its input. The decoder of the model is a simplified Context-Knowledge-Attention Decoder (CKAD). It doesn't have context-attention sub-layer either. This setup is to test how effective the context has been exploited in the full model.

3.3 Experiment Setup

We use OpenNMT-py[1] (Klein et al., 2017) as the code framework[2]. For all models, the hidden size is set to 512. For rnn-based models (Seq2Seq, HRED), 3-layer bidirectional LSTM (Hochreiter and Schmidhuber, 1997) and 1-layer LSTM is applied for encoder and decoder respectively. For transformer-based models, the layers of both encoder and decoder are set to 3. The number of attention heads in multi-head attention is 8 and the filter size is 2048. The word embedding is shared by utterances, knowledge and generated responses. The dimension of word embedding is set to 512 empirically. We use Adam (Kingma and Ba, 2014) for optimization. When decoding, beam size is set to 5. We use the previous three utterances and its related documents as input.

3.4 Evaluation Metrics

Automatic Evaluation: We adopt perplexity (PPL) and BLEU (Papineni et al., 2002) to automatically evaluate the response generation performance. Models are evaluated using perplexity of the gold response as described in (Dinan et al., 2018). Lower perplexity indicates better performance. BLEU measures n-gram overlap between a generated response and a gold response. However, since there is only one reference for each response and there may exist multiple feasible responses, BLEU scores are extremely low. We compute BLEU score by the *multi-bleu.perl*[3]

Manual Evaluation: Manual evaluations are essential for dialogue generation. We randomly sampled 30 conversations containing 606 utterances from the test set and obtained 5454 utterances from the nine models. We have annotators score these utterances given its previous utterances and related documents. We defined three metrics - **fluency**, **knowledge relevance** (Liu et al., 2018) and **context coherence** for manual evaluation. All these metrics are scored 0/1/2.

fluency: Whether the response is natural and

[1] https://github.com/OpenNMT/OpenNMT-py

[2] The code and models are available at https://github.com/lizekang/ITDD

[3] https://github.com/google/seq2seq/blob/master/bin/tools/multi-bleu.perl

Document	
moviename **despicable me** ... **cast: steve carell as gru** ... **rotten tomatoes: 81%** ... it is the debut film of **illumination entertainment**. ... gru, a **supervillain**, is disheartened when an unknown supervillain steals the great pyramid of giza. gru, with the assistance of his colleague dr. Nefario and his **minions**, resolves to one-up this mystery rival by shrinking and stealing the moon. as it would be costly to steal the moon, **gru seeks a loan from the bank of evil**. ... upon learning that an up-and-coming **villain** named vector was responsible for the pyramid theft ...	
Utterance 1	what do you think about **despicable me**?
ITE+DD	i think it is an awesome movie, **what about you?**
Wizard	i think it's a great movie!
Utterance 2	i rather liked it. it wasn't what i was expecting but it was still good.
ITE+DD	yeah, **it makes me laugh**, even when they make a movie about it.
Wizard	it's a great movie for kids and adults.
Utterance 3	i like it too. **funny and heart warming at the same time**.
ITE+DD	i think it's interesting that **gru decided to take a loan from the bank of evil**.
Wizard	i think they should have made a sequel.
Utterance 4	**gru was a really interesting character**, too. **i liked steve carell and the voice he used for gru**.
ITE+DD	yeah, now that you mention it, what did you think of the **minions**?
Wizard	i think the entire cast was great.
Utterance 5	yes, he did an excellent job portraying the character. **81% in ratings** i think is great.
ITE+DD	yes, **81%** is a great score!
Wizard	i also liked the **villain** of the movie.

Table 3: Responses generated by Incremental Transformer with Deliberation Decoder(**ITE+DD**) and the Wizard Transformer(**Wizard**). These utterances are from continuous dialogues. **ITE+DD** and **Wizard** generate responses using context utterances and a given document. Note that **Utterance** $i + 1$ is the gold response of **Utterance** i.

fluent. Score 0 represents not fluent and incomprehensible; 1 represents partially fluent but still comprehensible; 2 represents totally fluent.

knowledge relevance: Whether the response uses relevant and correct knowledge. Score 0 represents no relevant knowledge; 1 represents containing relevant knowledge but not correct; 2 represents containing relevant knowledge and correct.

context coherence: Whether the response is coherent with the context and guides the following utterances. Score 0 represents not coherent or leading the dialogue to an end; 1 represents coherent with the utterance history but not guiding the following utterances; 2 represents coherent with utterance history and guiding the next utterance.

3.5 Experimental Results

Table 1 shows the automatic and manual evaluation results for both the baseline and our models.

In manual evaluation, among baselines, Wizard Transformer and RNN without knowledge have the highest fluency of 1.62 and Wizard obtains the highest knowledge relevance of 0.47 while Transformer without knowledge gets the highest context coherence of 0.67. For all models, ITE+CKAD obtains the highest fluency of 1.68 and ITE+DD has the highest Knowledge Relevance of 0.56 and highest Context Coherence of 0.90.

In automatic evaluation, our proposed model has lower perplexity and higher BLEU scores than baselines. For BLEU, HRED with knowledge obtains the highest BLEU score of 0.77 among the baselines. And ITE+DD gets 0.95 BLEU score, which is the highest among all the models. For perplexity, Wizard Transformer obtains the lowest perplexity of 70.30 among baseline models and ITE+DD has remarkably lower perplexity of 15.11 than all the other models. A detailed analysis is in Section 3.6.

3.6 Analysis and Discussion

To our surprise, ITE+DD reaches an extremely low ground truth perplexity. We find that the ground truth perplexity after the first-pass decoding is only similar to the ITE+CKAD. It shows that the second-pass decoder utilizes the document knowledge well, and dramatically reduced the ground truth perplexity.

As shown in Table 2, ITE+DD has a higher percent of score 2 both on Knowledge Relevance and

ID	Utterance	Two-pass Responses
1	I think rachel mcadams had an even better role as regina george however! would you agree?	i'm not a fan of **kristen bell**, but i think **she** did a great job.
		i'm not a huge fan of **rachel mcadams**, but **he** did a great job.
2	yeah, I guess that's always worth it, and a truce was made as well.	yeah, **not only** does she reconcile with the **plastics**.
		yeah, she reconciles with **janis** , **damien** and **aaron**.
3	i liked the scene where buzz thinks he's a big shot hero but then the camera reveals him to be a tiny toy.	i think that's one of the best scenes in the movie.
		oh, i think that is what makes the movie unique as well. **have you seen any of the other pixar movies**?

Table 4: Examples of the two pass decoding. Underlined texts are the differences between two results. For each case, the first-pass response is on the top.

Context Coherence than ITE+CKAD. This result also demonstrates that Deliberation Decoder can improve the knowledge correctness and guide the following conversations better.

Although the perplexity of ITE+CKAD is only slightly better than KAT, the BLEU score, Fluency, Knowledge Relevance and Context Coherence of ITE+CKAD all significantly outperform those of KAT model, which indicates that Incremental Transformer can deal with multi-turn document grounded conversations better.

Wizard Transformer has a great performance on Knowledge Relevance only second to our proposed Incremental Transformer. However, its score on Context Coherence is lower than some other baselines. As shown in Table 2, Wizard Transformer has Knowledge Relevance score 1 results twice more than score 2 results, which indicates that the model tends to generate responses with related knowledge but not correct. And the poor performance on Context Coherence also shows Wizard Transformer does not respond to the previous utterance well. This shows the limitation of representing context and document knowledge by simple concatenation.

3.7 Case Study

In this section, we list some examples to show the effectiveness of our proposed model.

Table 3 lists some responses generated by our proposed Incremental Transformer with Deliberation Decoder (ITE+DD) and Wizard Transformer (which achieves overall best performance among baseline models). Our proposed model can generate better responses than Wizard Transformer on knowledge relevance and context coherence.

To demonstrate the effectiveness of the two-pass decoder, we compare the results from the first-pass decoding and the second-pass decoding. Table 4 shows the improvement after the second-pass decoding. For Case 1, the second-pass decoding result revises the knowledge error in the first-pass decoding result. For Case 2, the second-pass decoder uses more detailed knowledge than the first-pass one. For Case 3, the second-pass decoder cannot only respond to the previous utterance but also guide the following conversations by asking some knowledge related questions.

4 Related Work

The closest work to ours lies in the area of open-domain dialogue system incorporating unstructured knowledge. Ghazvininejad et al. (2018) uses an extended Encoder-Decoder where the decoder is provided with an encoding of both the context and the external knowledge. Parthasarathi and Pineau (2018) uses an architecture containing a Bag-of-Words Memory Network fact encoder and an RNN decoder. Dinan et al. (2018) combines Memory Network architectures to retrieve, read and condition on knowledge, and Transformer architectures to provide text representation and generate outputs. Different from these works, we greatly enhance the Transformer architectures to handle the document knowledge in multi-turn dialogue from two aspects: 1) using attention mechanism to combine document knowledge and context utterances; and 2) exploiting incremental encoding scheme to encode multi-turn knowledge aware conversations.

Our work is also inspired by several works in other areas. Zhang et al. (2018) introduces document context into Transformer on document-level Neural Machine Translation (NMT) task. Guan et al. (2018) devises the incremental encoding scheme based on rnn for story ending generation task. In our work, we design an Incremental Transformer to achieve a knowledge-aware context representation using an incremental encoding scheme. Xia et al. (2017) first proposes Deliberation Network based on rnn on NMT task. Our Deliberation Decoder is different in two aspects: 1) We clearly devise the two decoders targeting context and knowledge respectively; 2) Our second pass decoder directly fine tunes the first pass result, while theirs uses both the hidden states and results from the first pass.

5 Conclusion and Future Work

In this paper, we propose an Incremental Transformer with Deliberation Decoder for the task of Document Grounded Conversations. Through an incremental encoding scheme, the model achieves a knowledge-aware and context-aware conversation representation. By imitating the real-world human cognitive process, we propose a Deliberation Decoder to optimize knowledge relevance and context coherence. Empirical results show that the proposed model can generate responses with much more relevance, correctness, and coherence compared with the state-of-the-art baselines. In the future, we plan to apply reinforcement learning to further improve the performance.

6 Acknowledgments

This work is supported by 2018 Tencent Rhino-Bird Elite Training Program, National Natural Science Foundation of China (NO. 61662077, NO.61876174) and National Key R&D Program of China (NO.YS2017YFGH001428). We sincerely thank the anonymous reviewers for their thorough reviewing and valuable suggestions.

References

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1870–1879.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of wikipedia: Knowledge-powered conversational agents. *arXiv preprint arXiv:1811.01241*.

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 *Thirty-Second AAAI Conference on Artificial Intelligence*.

Jian Guan, Yansen Wang, and Minlie Huang. 2018. Story ending generation with incremental encoding and commonsense knowledge. *arXiv preprint arXiv:1808.10113*.

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

Hsin-Yuan Huang, Eunsol Choi, and Wen-tau Yih. 2018. Flowqa: Grasping flow in history for conversational machine comprehension. *arXiv preprint arXiv:1810.06683*.

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

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proc. ACL*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119.

Shuman Liu, Hongshen Chen, Zhaochun Ren, Yang Feng, Qun Liu, and Dawei Yin. 2018. Knowledge diffusion for neural dialogue generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1489–1498.

Hao-Tong Ye Kai-Ling Lo and Shang-Yu Su Yun-Nung Chen. 2019. Knowledge-grounded response generation with deep attentional latent-variable model. *Thirty-Third AAAI Conference on Artificial Intelligence*.

Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1468–1478.

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

Prasanna Parthasarathi and Joelle Pineau. 2018. Extending neural generative conversational model using external knowledge sources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 690–695.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you dont know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 784–789.

Siva Reddy, Danqi Chen, and Christopher D Manning. 2018. Coqa: A conversational question answering challenge. *arXiv preprint arXiv:1808.07042*.

Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1577–1586.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

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.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.

Yingce Xia, Fei Tian, Lijun Wu, Jianxin Lin, Tao Qin, Nenghai Yu, and Tie-Yan Liu. 2017. Deliberation networks: Sequence generation beyond one-pass decoding. In *Advances in Neural Information Processing Systems*, pages 1784–1794.

Hao Xiong, Zhongjun He, Hua Wu, and Haifeng Wang. 2018. Modeling coherence for discourse neural machine translation. *arXiv preprint arXiv:1811.05683*.

Semih Yavuz, Abhinav Rastogi, Guan-lin Chao, Dilek Hakkani-Tür, and Amazon Alexa AI. 2018. Deepcopy: Grounded response generation with hierarchical pointer networks. *Advances in Neural Information Processing Systems*.

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. *arXiv preprint arXiv:1804.09541*.

Jiacheng Zhang, Huanbo Luan, Maosong Sun, Feifei Zhai, Jingfang Xu, Min Zhang, and Yang Liu. 2018. Improving the transformer translation model with document-level context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 533–542.

Kangyan Zhou, Shrimai Prabhumoye, and Alan W Black. 2018. A dataset for document grounded conversations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 708–713.