Improving Knowledge-Aware Dialogue Generation via Knowledge Base Question Answering

Jian Wang,[1] **Junhao Liu,**[2] **Wei Bi,**[3] **Xiaojiang Liu,**[3] **Kejing He,**[1]* **Ruifeng Xu,**[4] **Min Yang**[2]*

[1]South China University of Technology, Guangzhou, China
[2]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China
[3]Tencent AI Lab, Shenzhen, China
[4]Harbin Institute of Technology (Shenzhen), China
cs_wangjian@mail.scut.edu.cn, {jh.liu, min.yang}@siat.ac.cn, {victoriabi, kieranliu}@tencent.com,
kjhe@scut.edu.cn, xuruifeng@hit.edu.cn

Abstract

Neural network models usually suffer from the challenge of incorporating commonsense knowledge into the open-domain dialogue systems. In this paper, we propose a novel knowledge-aware dialogue generation model (called TransDG), which transfers question representation and knowledge matching abilities from knowledge base question answering (KBQA) task to facilitate the utterance understanding and factual knowledge selection for dialogue generation. In addition, we propose a response guiding attention and a multi-step decoding strategy to steer our model to focus on relevant features for response generation. Experiments on two benchmark datasets demonstrate that our model has robust superiority over compared methods in generating informative and fluent dialogues. Our code is available at https://github.com/siat-nlp/TransDG.

1 Introduction

Building a dialogue system that is capable of providing informative responses is a long-term goal of artificial intelligence (AI). Recent advances on dialogue systems are overwhelmingly contributed by deep learning techniques (i.e., sequence-to-sequence model) (Sutskever, Vinyals, and Le 2014), which have taken the state-of-the-art of dialogue systems to a new level. However, fully data-driven neural models (Serban et al. 2016; Li et al. 2016b; 2016c) tend to generate responses that are conversationally appropriate but seldom include factual content. Previous studies (Ghazvininejad et al. 2018; Zhou et al. 2018) revealed that infusing commonsense knowledge into dialogue systems could enhance user satisfaction and contribute to highly versatile and applicable open-domain dialogue systems.

Several studies have been proposed to integrate external knowledge into dialogue generation (Zhu et al. 2017; Liu et al. 2018). Despite the effectiveness of previous studies, there are still several challenges for generating informative and appropriate conversation, which are not addressed
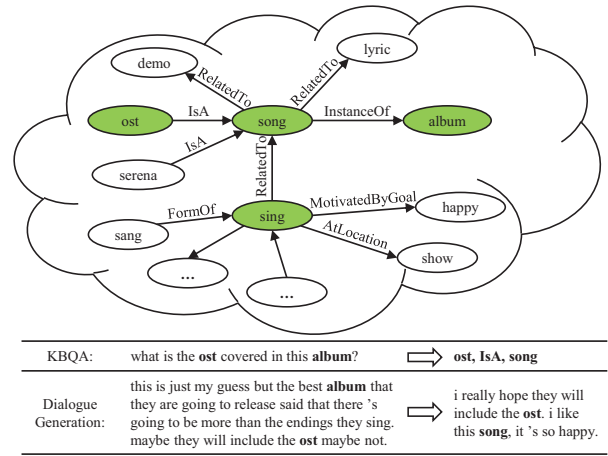

Figure 1: Examples from a real-life dataset show that KBQA can facilitate the utterance understanding and factual knowledge selection for generating informative dialogue, e.g., locating the KB fact ⟨ost, IsA, song⟩.

well in prior works. First, prior methods (Zhou et al. 2018) extract knowledge from KB by using each word in the post as query to retrieve related facts in an explicit manner. However, in dialogue systems, matching posts to exact facts in KB is much harder than explicit factoid inquiries answering. For some posts, the subjects and relations are elusive, e.g., the related entities are far from each other in the post (see Figure 1), which leads to trouble in matching relational facts from KB. Second, the generation-based methods generate the response word by word and lack a global perspective. As a result, the knowledge connection between the post and potential response (entity diffusion) is ignored, making the generated knowledge (entities) in responses not appropriate and reasonable with respect to the post. Third, most previous studies focus on enriching entities or triples for generation by merely incorporating information from KB. However, it is difficult to retrieve related facts and generate meaningful responses relying on solely the insufficient input posts, especially when the input posts are really short.

To deal with the aforementioned challenges, we propose

a novel knowledge-aware dialogue generation model (called TransDG), which effectively fuses external knowledge in KB into sequence-to-sequence model to generate informative dialogues by transferring the question modeling and knowledge matching abilities from KBQA, with the intuition that KBQA can facilitate the utterance understanding and factual knowledge (facts in KB) selection in dialogue generation (see Figure 1). **First**, we pre-train a KBQA model, which consists of an encoding layer for representing both questions and candidate answers, and a knowledge selection layer for selecting the most appropriate answer from KB. **Second**, we encode the input post using gated recurrent unit (GRU) with augmentation of question representation layer learned from the KBQA model as dialogue encoder. **Third**, we integrate commonsense knowledge to generate informative responses by transferring the knowledge selection layer learned from the KBQA model with multi-step decoding strategy. The first-step decoder generates a draft response by attending relevant facts (entities) related to the post, while the second-step decoder generates the final response by referring to the context knowledge learned by the first-step decoder and improves the overall correctness of the generated response. To further improve the informativeness of dialogues, we propose a response guiding attention mechanism, which leverages top-k retrieved responses of similar posts to distill information in the input post.

Our contributions are summarized as follows:

- We propose TransDG, a novel knowledge-aware dialogue generation model, which transfers the abilities of question understanding and fact extraction from the pre-trained KBQA model to facilitate both post understanding and factual knowledge selection from KB.

- We propose a multi-step decoding strategy which captures the knowledge connection between the post and response. Both the post and draft response generated by the first-step decoder is matched with relevant facts from KB, which makes the final response generated by the second-step decoder more appropriate and reasonable with respect to the post.

- We propose a response guiding attention mechanism which steers the model to focus on relevant features with the help of k-best response candidates.

- Extensive experiments on a real dialogue dataset show that our model outperforms the compared methods from both quantitative and qualitative perspectives.

2 Related Work

Open domain dialogue generation (Serban et al. 2016) aims at generating meaningful and coherent dialogue responses given input dialogue history. It is an important but challenging task that has received much attention recently from the natural language processing (NLP) community. Various techniques have been proposed to improve the quality of generated responses from different perspectives, such as diversity promotion (Li et al. 2016a), unknown words handling (Gu et al. 2016), prototype editing (Wu et al. 2018) and retrieval-based ensemble (Song et al. 2018). These models

are end-to-end trainable and have good language modeling ability. However, a well-known problem of these methods is that they are prone to generate universal and even meaningless responses.

Recently, incorporating external knowledge in open-domain dialogue generation is demonstrated to be effective to improve the performance of dialogue models. Some previous studies treated unstructured texts as external knowledge, which applied a convolutional neural network (Long et al. 2017) or a memory network (Ghazvininejad et al. 2018) to extract external knowledge for improving response generation. Many recent work incorporated open-domain knowledge bases in dialogue generation (Zhu et al. 2017; Liu et al. 2018; Zhou et al. 2018; Lian et al. 2019). Specifically, related knowledge was acquired from knowledge bases to build knowledge grounded dialogues by using a copy network (Zhu et al. 2017). A neural knowledge diffusion model (Liu et al. 2018) was proposed to further integrate knowledge bases with dialogues through facts matching and entity diffusion. In addition, large scale commonsense knowledge bases were utilized in dialogue generation through graph attention mechanism (Zhou et al. 2018). The posterior knowledge distribution was also utilized to guide knowledge selection for response generation (Lian et al. 2019).

On the other hand, Knowledge Base Question Answering (KBQA) has also been an active research field in recent years, which aims at selecting an appropriate factual answer from structured knowledge bases, such as DBpedia (Auer et al. 2007) and Freebase (Bollacker et al. 2008), given a query. A variety of methods have been proposed for KBQA, including information retrieval based methods (Yao and Van Durme 2014; Xu et al. 2016), semantic parsing based methods (Yih et al. 2015; Hu et al. 2018) and neural network based methods (Hao et al. 2017; Yu et al. 2017; Luo et al. 2018). In most neural network based approaches, both questions and candidate answers (or knowledge facts) are encoded into distributed representations, and then the similarity calculation is used to select the most appropriate answer. For example, Luo et al. (2018) employed an ensemble method to handle complex questions by leveraging dependency parsing and entity linking to enrich question representation (Luo et al. 2018).

3 Knowledge Base Question Answering

As shown in Figure 2, our model contains two parts: a KBQA model and a dialogue generation model, where knowledge learned from the KBQA task is transferred to dialogue generation in both encoding and decoding phases. In this section, we describe the KBQA model in detail.

Given a question $Q = \{x_1, \ldots, x_n\}$, the task of KBQA is to select an appropriate answer from a set of candidate answers (facts) $A = \{a_1, \ldots, a_m\}$ in structured knowledge bases, where n and m are the lengths of the given question and candidate answer set, respectively. A common idea of KBQA is to encode the questions and facts in KBs into distributed representations, and then perform semantic matching between the question and candidate answer representations to obtain the final answer.
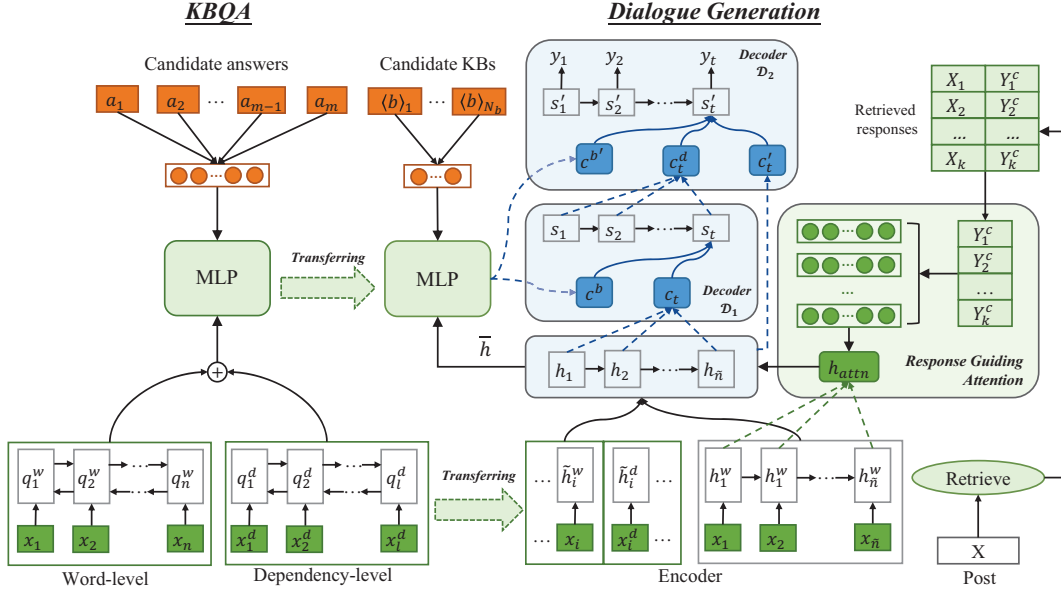
Figure 2: Overview of TransDG model, which consists of a knowledge base question answering (KBQA) part (left) and knowledge-aware dialogue generation part (right), where the KBQA is pre-trained for transferring knowledge.

3.1 Encoding Layer

Question Representation We leverage both word-level and dependency-level information to learn the representation of question Q. For the word-level information, we convert each word x_i into a word vector $\mathbf{v}_i^w \in \mathbb{R}^d$ through a word embedding matrix $\mathbf{E}^w \in \mathbb{R}^{|V_w| \times d}$, where $|V_w|$ is vocabulary size and d is the size of word embedding. Then, we employ a bidirectional gated recurrent unit (BiGRU) (Cho et al. 2014) to obtain the hidden states of words in the question. Formally, the hidden state $\mathbf{q}_i^w$ at time step t can be updated as:

$$\mathbf{q}_i^w = \text{BiGRU}^w(\mathbf{v}_i^w, \mathbf{q}_{i-1}^w) \qquad (1)$$

To better capture the long-range relations between the words in the question, we follow (Xu et al. 2016) to use the dependency path as additional representation, concatenating the words and dependency labels with directions. For example, the dependency path of question "Who is the best actor in the movie" is {who, $\overrightarrow{nsubj}$, actor, $\overrightarrow{prep}$, in, $\overrightarrow{pobj}$, $\langle E \rangle$}, where $\overrightarrow{nsubj}$, $\overrightarrow{prep}$ and $\overrightarrow{pobj}$ denote noun subjective, preposition and predicate objective respectively. $\langle E \rangle$ is a dummy token representing an entity word so as to learn a more general relation representation in syntactic level. We use the widely-used dependency parser Stanford CoreNLP (Manning et al. 2014) to obtain dependency-level tokens, denoted as $x^d = \{x_1^d, \ldots, x_l^d\}$, where l is the length of the dependency-level input, we convert each token x_j^d into a word vector $\mathbf{v}_j^d \in \mathbb{R}^d$ through a dependency embedding matrix $\mathbf{E}^d \in \mathbb{R}^{|V_d| \times d}$. Here, $\mathbf{E}^d$ is randomly initialized and updated during training, $|V_d|$ is the vocabulary size of the dependency tokens. Then, we apply another BiGRU network to obtain the dependency-level question representation. For-

mally, the hidden state $\mathbf{q}_j^d$ is updated as:

$$\mathbf{q}_j^d = \text{BiGRU}^d(\mathbf{v}_j^d, \mathbf{q}_{j-1}^d) \qquad (2)$$

We align the word-level and dependency-level sequences by padding, and combine their hidden states by element-wise addition: $\mathbf{q}_i = \mathbf{q}_i^w + \mathbf{q}_i^d$. Hence, we can get the final question representation as $\mathbf{q} = [\mathbf{q}_1, \ldots, \mathbf{q}_n]$.

Candidate Answer Representation Typically, the candidate answers in KBQA task are denoted as $A = \{a_1, \ldots, a_m\}$, where each answer a_i is a fact from specific KB in the form of ⟨subject entity, relation, object entity⟩. We encode such facts at both word-level and path-level. Given the word sequence of answers, we use the same word embedding matrix $\mathbf{E}^w$ to convert the words into word vectors $(\mathbf{a}_1^w, \ldots, \mathbf{a}_m^w)$, where m is the length of the input answer. Then we calculate an average embedding as the word-level representation of the answer: $\mathbf{a}^w = \frac{1}{m} \sum_{i=1}^m \mathbf{a}_i^w$. For path-level, we treat each relation as a whole unit (e.g., "related_to") and directly translate it into vector representation $\mathbf{a}^p$ through a KB embedding matrix $\mathbf{E}^k \in \mathbb{R}^{|V_k| \times d}$. Here, $\mathbf{E}^k$ is randomly initialized and updated during training, $|V_k|$ is the size of relations in the KB. The final representation of each candidate answer is defined as: $\mathbf{a} = \mathbf{a}^w + \mathbf{a}^p$.

3.2 Semantic Matching and Model Training

We calculate the semantic similarity score between question q_i and candidate answer a_j through a multi-layer perceptron:

$$S(q_i, a_j) = \text{MLP}([\mathbf{q}_i, \mathbf{a}_j]). \qquad (3)$$

During training, we adopt hinge loss to maximize the margin between positive answer set and negative answer set:

$$L_{q,A} = \max\{0, \gamma - S(q, A^+) + S(q, A^-)\} \qquad (4)$$

where A^+ are gold answers, A^- are randomly sampled negative answers from the knowledge base for given questions, γ is a parameter to tune margin between positive and negative samples.

4 Knowledge-aware Dialogue Generation

Given a post $X = \{x_1, \ldots, x_{\tilde{n}}\}$, the goal of dialogue generation is to generate a proper response $Y = \{y_1, \ldots, y_{\tilde{m}}\}$, where $\tilde{n}$ and $\tilde{m}$ are the lengths of the post and response, respectively. As shown in Figure 2, our dialogue generation model transfers knowledge from KBQA task, facilitating the knowledge-level dialogue understanding and fact selection from KB.

4.1 Knowledge-aware Dialogue Encoder

The dialogue generation employs a sequence-to-sequence (Seq2Seq) based method to generate a response for a given post. The encoder of Seq2Seq reads the post X word by word and generates a hidden representation of each word by a GRU. Formally, given the input word embedding $\mathbf{e}_i^w$ for word x_i in the post, the hidden state $\mathbf{h}_i^w$ can be updated by:

$$\mathbf{h}_i^w = \text{GRU}(\mathbf{e}_i^w, \mathbf{h}_{i-1}^w) \tag{5}$$

To facilitate the understanding of a post, we transfer the ability of question representation in KBQA task to obtain multi-level semantic understandings (i.e., word level and dependency level). Formally, we use the pre-trained bidirectional GRUs learned by KBQA task as additional encoders:

$$\tilde{\mathbf{h}}_i^w = \text{BiGRU}^w(\mathbf{e}_i^w, \tilde{\mathbf{h}}_{i-1}^w) \tag{6}$$

$$\tilde{\mathbf{h}}_i^d = \text{BiGRU}^d(\mathbf{e}_i^d, \tilde{\mathbf{h}}_{i-1}^d) \tag{7}$$

We combine $\tilde{\mathbf{h}}_i^w$ and $\tilde{\mathbf{h}}_i^d$ by element-wise addition to obtain the KBQA based post representation: $\tilde{\mathbf{h}}_i = \tilde{\mathbf{h}}_i^w + \tilde{\mathbf{h}}_i^d$.

Response Guiding Attention To enrich the post representation for better comprehension, we propose a response guiding attention mechanism, which uses the retrieved responses of similar posts to steer the model to focus only on relevant information. First, we use the widely-used text retrieval tool Lucene[1] to retrieve top-k similar posts for a given post, the corresponding responses of the k selected posts serve as our candidate responses, denoted as $[Y_1^c, \ldots, Y_k^c]$. For each candidate response Y_m^c ($m = 1, \ldots, k$), we first convert it into word vectors $(\mathbf{y}_1, \ldots, \mathbf{y}_{\tilde{m}})$ through the word embedding matrix $\mathbf{E}^w$, and then represent the response by averaging operation: $e(Y_m^c) = \frac{1}{\tilde{m}} \sum_{j=1}^{\tilde{m}} \mathbf{y}_j$. We then calculate the mutual attention between the candidate response Y_m^c and the hidden representation of the post to obtain the m-th candidate response guided post representation:

$$\mathbf{h}_{attn}^{(m)} = \sum_{i=1}^{\tilde{n}} \alpha_{mi} \mathbf{h}_i, \quad \alpha_{mi} = \text{softmax}(\beta_{mi}) \tag{8}$$

$$\beta_{mi} = \mathbf{V}_h^T \tanh(\mathbf{W}_h e(Y_m^c) + \mathbf{U}_h \mathbf{h}_i) \tag{9}$$

where $\mathbf{V}_h$, $\mathbf{W}_h$ and $\mathbf{U}_h$ are parameters to be learned. Finally, the candidate responses guided post representation is calculated by averaging the k candidate responses guided post representations:

$$\mathbf{h}_{attn} = \frac{1}{k} \sum_{m=1}^{k} \mathbf{h}_{attn}^{(m)} \tag{10}$$

Finally, the integrated post representation is formulated by concatenating $\mathbf{h}_{\tilde{n}}^w$, $\tilde{\mathbf{h}}_{\tilde{n}}$ and $\mathbf{h}_{attn}$, denoted as: $\mathbf{h}_{final} = [\mathbf{h}_{\tilde{n}}^w; \tilde{\mathbf{h}}_{\tilde{n}}; \mathbf{h}_{attn}]$.

4.2 Knowledge-aware Multi-step Decoder

The knowledge-aware decoder generates responses by transferring the knowledge selection ability learned from the pre-trained KBQA model using a multi-step decoding strategy. The first-step decoder generates a draft response by incorporating external knowledge relevant to the post. The second-step decoder generates the final response by referring to the post, the context knowledge and the draft response produced by the first-step decoder. In this way, the multi-step decoder can capture the knowledge connection between the post and response, and therefore generate more coherent and informative response.

First-step decoder Formally, in the first-step decoder $\mathcal{D}_1$, the hidden state $\mathbf{s}_t$ of the decoder GRU at time step t is updated as:

$$\mathbf{s}_t = \text{GRU}(\mathbf{s}_{t-1}, [\mathbf{c}_{t-1}; \mathbf{c}^b; \mathbf{e}(\hat{y}_{t-1})]) \tag{11}$$

where $\mathbf{e}(\hat{y}_{t-1})$ is the embedding of previously generated word $\hat{y}_{t-1}$, $\mathbf{c}_{t-1}$ is the context vector at time step $t-1$, $\mathbf{c}^b$ is the attention vector augmenting the selected knowledge by KBQA model. Similar to previous studies, the context vector $\mathbf{c}_t$ is calculated as follows:

$$\mathbf{c}_t = \sum_{k=1}^{\tilde{n}} \alpha_{tk} \mathbf{h}_k, \quad \alpha_{tk} = \text{softmax}(e_{tk}) \tag{12}$$

$$e_{tk} = \mathbf{V}_1^T \tanh(\mathbf{W}_1 \mathbf{s}_{t-1} + \mathbf{U}_1 \mathbf{h}_k) \tag{13}$$

where $\mathbf{V}_1$, $\mathbf{W}_1$ and $\mathbf{U}_1$ are parameters to be learned.

The attention vector $\mathbf{c}^b$ is proposed to transfer the ability of selecting appropriate commonsense knowledge from the pre-trained KBQA model. Concretely, for a given post X, we first retrieve all the relevant triples from the knowledge base using the words in X as queries, where each triple is represented as ⟨subject entity, relation, object entity⟩. All subject entities and object entities serve as our commonsense knowledge candidates, denoted as $\{\langle b_s, b_o \rangle_1, \ldots, \langle b_s, b_o \rangle_{N_b}\}$, where b_s and b_o represent subject and object entities, and N_b is the number of the knowledge candidates. We take the encoded hidden representation of the post as input of the MLP layer defined in Eq. (3) of the pre-trained KBQA model to learn the correlation between the post and the candidate knowledge:

$$\mathbf{c}^b = \frac{\sum_{j=1}^{N_b} r_j \mathbf{e}(b_j)}{\sum_{j=1}^{N_b} r_j}, \quad r_j = \text{MLP}([\bar{\mathbf{h}}, \mathbf{e}(b_j)]) \tag{14}$$

where $\mathbf{e}(b_j)$ is the concatenation of the word embeddings of the j-th subject and object entities, $\overline{\mathbf{h}}$ is the average value of the encoded hidden representation of the post, calculated as $\overline{\mathbf{h}} = \frac{1}{\tilde{n}} \sum_{k=1}^{\tilde{n}} \mathbf{h}_k$.

Therefore, the distribution of the draft response is given by:

$$P(\hat{y}_t|Y_{<t}, X) \propto \text{softmax}(\mathbf{W}_p[\mathbf{s}_t; \mathbf{c}_t; \mathbf{c}^b]) \quad (15)$$

where $\mathbf{W}_p$ is a trainable parameter.

Second-step decoder   For the second-step decoder $\mathcal{D}_2$, we take the hidden information generated by decoder $\mathcal{D}_1$ and the candidate knowledge into consideration. The second-step decoder can generate more appropriate and reasonable responses with respect to the post by matching relevant facts from KB for both the post and the draft response generated by the first-step decoder. Formally, the hidden state of $\mathcal{D}_2$ is computed as:

$$\mathbf{s}_t' = \text{GRU}(\mathbf{s}_{t-1}', [\mathbf{c}_{t-1}'; \mathbf{c}_{t-1}^d; \mathbf{c}^{b'}; \mathbf{e}(y_{t-1})]) \quad (16)$$

where the computation of $\mathbf{c}_t'$ is similar to that of $\mathbf{c}_t$ defined in Eq.(12) and Eq.(13). $\mathbf{c}_t^d$ is the first-step contextual information vector, which is defined as:

$$\mathbf{c}_t^d = \sum_{k=1}^{T_s} \gamma_{tk}\mathbf{s}_k, \quad \gamma_{tk} = \text{softmax}(g_{tk}) \quad (17)$$

$$g_{tk} = \mathbf{V}_2^T \tanh(\mathbf{W}_2\mathbf{s}_{t-1}' + \mathbf{U}_2\mathbf{s}_k) \quad (18)$$

where $\mathbf{s}_k$ is the hidden state of k-th time step in decoder $\mathcal{D}_1$, T_s denotes the length of the time step in decoder $\mathcal{D}_1$. The calculation of $\mathbf{c}^{b'}$ is similar to that of $\mathbf{c}^b$. The difference is that in decoder $\mathcal{D}_2$ we aim to capture the correlation between the draft response and the candidate knowledge:

$$\mathbf{c}^{b'} = \frac{\sum_{j=1}^{N_b} r_j'\mathbf{e}(b_j)}{\sum_{j=1}^{N_b} r_j'}, \quad r_j' = \text{MLP}([\overline{\mathbf{s}}, \mathbf{e}(b_j)]) \quad (19)$$

where $\overline{\mathbf{s}}$ is the average hidden representation in decoder $\mathcal{D}_1$, defined as: $\overline{\mathbf{s}} = \frac{1}{T_s} \sum_{k=1}^{T_s} \mathbf{s}_k$.

Finally, the generation distribution is formulated as follows:

$$P(y_t|Y_{<t}, X) \propto \text{softmax}(\mathbf{W}_o[\mathbf{s}_t'; \mathbf{c}_t'; \mathbf{c}_t^d; \mathbf{c}^{b'}]) \quad (20)$$

where $\mathbf{W}_o$ is a trainable parameter.

4.3   Model Training

Our model is optimized in an end-to-end manner. We use D to represent the training dataset, and Θ_e, Θ_1, Θ_2 to represent the parameters of the encoder, the first-step decoder, the second-step decoder respectively. The training of the first-step decoding is to minimize the following loss:

$$L(\mathcal{D}_1) = -\sum_{X,Y \in D} \sum_{t=1}^{\bar{m}} \log P(\hat{y}_t|Y_{<t}, X; \Theta_e, \Theta_1) \quad (21)$$

Similarly, the second-step decoding is optimized by minimizing the following loss:

$$L(\mathcal{D}_2) = -\sum_{X,Y \in D} \sum_{t=1}^{\bar{m}} \log P(y_t|Y_{<t}, X; \Theta_e, \Theta_2) \quad (22)$$

Finally, the total loss is the sum of $L(\mathcal{D}_1)$ and $L(\mathcal{D}_2)$:

$$L = L(\mathcal{D}_1) + L(\mathcal{D}_2) \quad (23)$$

5   Experimental Setup

5.1   Datasets

We use SimpleQuestions (Bordes et al. 2015) dataset[2] to train the KBQA model which consists of 75,910/10,845/21,687 instances for training/validation/testing, respectively. Each instance is a question paired with a knowledge triple as the gold answer. We use FB2M as the KB for SimpleQuestions, which is a subset of Freebase provided with the SimpleQuestions as default. It contains about 2M entities and 10M triples. Both the SimpleQuestions and the FB2M are used only for pre-training the KBQA model. The pre-trained KBQA model achieves 92.89 of F1 score on validation set.

For dialogue generation, we use Reddit (Zhou et al. 2018) single-round dialogue dataset[3], which contains 3,384,185 training pairs, 10,000 validation pairs and 20,000 test pairs. Each post-response pair is connected by one or more triple in ConceptNet[4], which is used as the commonsense KB. It is noteworthy that we use the same dialogue dataset and commonsense knowledge base (i.e., ConceptNet) as in previous work (Zhou et al. 2018) for fair comparison. However, we would like to emphasize that the proposed model is general and can easily use other commonsense knowledge base (e.g., Freebase) to generate dialogues. The statistics of the datasets for dialogue generation and KBQA are shown in Table 1.

Task	QA / Dialog pairs		Knowledge Base	
	Training	75,910	Entities	2M
KBQA	Validation	10,845	Relations	31,940
	Testing	21,687	Triples	10M
	Training	3,384,185	Entities	21,471
Dialog	Validation	10,000	Relations	44
	Testing	20,000	Triples	120,850

Table 1: Statistics of datasets for different tasks.

5.2   Implementation Details

For KBQA, we initialize word embeddings using Glove (Pennington, Socher, and Manning 2014) word vectors. The sizes of both word embedding and KB embedding are set to 300. All BiGRUs are 1-layer BiGRU with 256 hidden units. The MLP model is a 2-layer fully-connected network with 512 hidden units and 1 hidden unit respectively. We set the margin γ to be 0.5, and sample 20 negative samples for each gold answer. The model is trained using Adam (Kingma and Ba 2014) optimizer with an initial learning rate 0.001. The batch size is set to 128.

For dialogue generation, we also use 300 dimensional Glove word vectors to initialize the word embeddings. The vocabulary size is set to 30,000. The encoder and decoder

have 2-layer GRUs with 512 hidden units for each layer. The dropout rate is set to 0.2. The number of candidate responses k is set to 3, and we adopt the default configurations provided by the Lucene API. We train the model using Adam optimizer with an initial learning rate of 0.0005, and the batch size is set to 100.

5.3 Baselines

We compare our model with following suitable baselines:

- **Seq2Seq:** a standard sequence-to-sequence model (Sutskever, Vinyals, and Le 2014), which is widely used as a baseline in dialogue generation.

- **CopyNet:** a sequence-to-sequence based model with copy mechanism (Zhu et al. 2017), which acquires knowledge by copying entity words from related knowledge bases.

- **MemNet:** a knowledge-grounded model which uses memory units to process knowledge triples (Ghazvininejad et al. 2018).

- **CCM:** a knowledge-aware dialogue generation model with static and dynamic graph attention mechanisms (Zhou et al. 2018).

- **PostKS:** a knowledge-guided dialogue generation model which employs the posterior knowledge distribution to guide the knowledge selection (Lian et al. 2019).

5.4 Evaluation Metrics

Both automatic and human evaluation metrics are used to measure the performance of our model. For automatic evaluation, we adopt **perplexity** and **entity score** (Zhou et al. 2018) as evaluation metrics, following previous work (Zhou et al. 2018). The perplexity is widely used to quantify a language model, where a model performs better when perplexity is smaller. The entity score is used to measure the ability to generate relevant entities per response from the commonsense KB. Higher entity score generally indicates the generated response is more diverse. To further evaluate the quality of dialogue systems, we also adopt **BLEU** (Papineni et al. 2002) as another automatic metric, which calculates n-gram overlaps between the generated response and the gold response.

We use human evaluation to evaluate the dialogue generation models from three perspectives: **fluency**, **knowledge relevance** and **correctness** (Liu et al. 2018). All the values are scored from 0 to 3, where higher score means better performance. Specifically, 500 posts are randomly selected from the test set, resulting in 3,000 responses generated by TransDG and baseline models in total for human evaluation. Three annotators are recruited to independently assign three scores for each generated response. The agreement ratio computed with Fless' kappa (Fleiss 1971) is 0.58, showing moderate agreement. We report the average rating scores from all annotators as the final human evaluation results.

Model	Overall	High	Medium	Low	OOV
Seq2Seq	47.02	42.41	47.25	48.61	49.96
MemNet	46.85	41.93	47.32	48.86	49.52
CopyNet	40.27	36.26	40.99	42.09	42.24
CCM	39.18	35.36	39.64	40.67	40.87
PostKS	43.56	40.65	44.06	46.36	49.32
TransDG	**37.53**	**32.18**	**36.12**	**38.46**	**40.75**

Table 2: Automatic evaluation with **perplexity**.

Model	Overall	High	Medium	Low	OOV
Seq2Seq	0.717	0.713	0.740	0.721	0.669
MemNet	0.761	0.764	0.788	0.760	0.706
CopyNet	0.960	0.910	0.970	0.960	0.960
CCM	1.180	1.156	1.191	1.196	1.162
PostKS	1.041	1.007	1.028	0.993	0.978
TransDG	**1.207**	**1.195**	**1.204**	**1.232**	**1.182**

Table 3: Automatic evaluation with **entity score**.

6 Experimental Results

6.1 Quantitative Results

As shown in Table 2, TransDG achieves the lowest perplexity on all the datasets, indicating that the generated responses are more grammatical. Table 3 demonstrates that the models leveraging external knowledge achieve better performance than the standard Seq2Seq model in generating meaningful entity words and diverse responses. In particular, our model outperforms all the baselines significantly with highest entity score. This verifies the effectiveness of transferring knowledge from KBQA task for factual knowledge selection. The BLEU values shown in Table 4 demonstrates the comparison results from word-level overlaps. TransDG tends to generate responses that are more similar to the gold responses than baselines in most cases. This may be because that our model utilizes retrieved candidate responses to provide guidance. In addition, we observe that CopyNet also performs well in terms of BLEU score, since it incorporates copying mechanism into the decoding process, which can copy words or subsequences from the input post and KB.

The human evaluation results are reported in Table 5. As shown in Table 5, TransDG tends to generate more appropriate and informative responses in terms of human annotation. Specifically, the responses generated by TransDG have higher knowledge relevance than other models, indicating that TransDG is effective to incorporate appropriate commonsense knowledge.

6.2 Case Study

Table 6 lists some responses generated by TransDG and the baselines. The Seq2Seq model is unable to comprehend the post since it does not incorporate the external commonsense knowledge, which further verifies the effectiveness of commonsense knowledge for open-domain dialogue generation. MemNet and CopyNet can generate fluent responses. However, these responses are not appropriate with respect to the conversation context. CCM and PostKS are capable of generating some informative words while the whole re-

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Seq2Seq	0.0977	0.0098	0.0012	0.0002
MemNet	0.1652	0.0174	0.0028	0.0004
CopyNet	0.1715	**0.0181**	0.0029	0.0005
CCM	0.1625	0.0175	0.0030	0.0005
PostKS	0.1683	0.0165	0.0029	0.0004
TransDG	**0.1807**	0.0178	**0.0031**	**0.0006**

Table 4: Automatic evaluation with **BLEU**.

Model	Fluency	Relevance	Correctness
Seq2Seq	1.67	0.68	0.80
MemNet	1.83	0.89	1.32
CopyNet	2.36	1.13	1.08
CCM	2.27	1.35	1.22
PostKS	2.32	1.36	1.31
TransDG	**2.41**	**1.52**	**1.34**

Table 5: Human evaluation result.

sponses are lack of continuity in logic. Instead, the proposed TransDG can generate a more reasonable response with appropriate entity words such as "fan" and "game".

Post: Did you play Mario Party 2?
Reference: Yes! I played this game recently with my friends on the N64, and it was amazing. I forgot how much fun I had with this as a kid.
Seq2Seq: It's a game, but it's not.
MemNet: I'm just going to have to wait for the next party to be in the game.
CopyNet: I'm not sure I'd be happy with that party.
CCM: I'm not a big Mario Party, but I'm a little disappointed.
PostKS: I have not played Mario Party yet, but I have not played it yet.
TransDG: I'm a big fan of Mario Party, but I'm not sure if I can get a copy of the game or play it on my pc.

Table 6: Case study of generated responses. The reference means the ground-truth response in the dataset.

6.3   Ablation Study

To investigate the effectiveness of each module proposed in our model, we conduct ablation test by removing the following modules: (1) the pre-trained question representation module transferred from KBQA (w/o QRT), which encodes the posts with a standard BiGRU defined by question encoding layer with random initialization, (2) the knowledge selection module transferred from KBQA (w/o KST), (3) the response guiding attention module (w/o RGA), and (4) the second-step decoder module (w/o SSD). The ablation test results are reported in Table 7. From the results, we can observe that the performance of TransDG drops sharply

when we discard the question representation module and the knowledge selection module transferred from KBQA. This is within our expectation since the question representation module transferred from KBQA helps the encoder to capture essential information (e.g., informative entities) from the post, and the knowledge selection module encourages the decoder to select appropriate facts from external KB. Response guiding attention also has noticeable impact on the performance of TransDG, especially on BLEU scores. The candidate responses highlight relevant context and suppress unimportant ones, enabling to generate more accurate responses. In addition, the second-step decoder can improve the ability of TransDG to generate relevant entities per response. It is no surprise that combining all the factors achieves the best performance for all evaluation metrics.

Model	Perplexity	Entity	BLEU-1	BLEU-2
TransDG	37.53	1.207	0.1807	0.0178
w/o QRT	42.17	1.076	0.1604	0.0171
w/o KST	43.05	0.774	0.1643	0.0158
w/o QRT+KST	44.15	0.772	0.1612	0.0170
w/o RGA	38.62	1.106	0.1712	0.0170
w/o SSD	38.18	1.114	0.1804	0.0178

Table 7: Ablation results of TransDG on the test set. Here, Entity represents entity score.

6.4   Error Analysis

To better understand the limitations of TransDG, we additionally carry out an analysis of the errors made by TransDG. We randomly select 100 responses generated by TransDG that achieve low human evaluation scores. We reveal several reasons for low human evaluation scores which can be divided into the following categories.

Illogical (36%): The top error category is illogical, including the responses that are contradictory or conflict with the input posts. For example, the response "I'm not sure he's a good player, especially he's a good player" has a score of 0 since it lacks continuity in logic. This type of errors are difficult to handle with current techniques, especially when trying to build an end-to-end model.

Miscellaneous (32%): The second most common error category is miscellaneous, which includes responses that are less informative or not reasonable caused by polysemic words. For example, given a post "You live under a rock? That saying is older than dirt", the generated response is "I'm not a fan of the rock music".

Irrelevant (20%): The third error category includes the responses that are fluent but not relevant to the post or too general to correctly answer the post, which occurs when the model fails to incorporate appropriate knowledge. For instance, a general response "I don't know, I'm glad I could build a good one!" is generated for a help-seeking post "Is there a tutorial to the design you used anywhere? Or maybe anything to help me build it?"

Ungrammatical (12%): Another error category includes the responses that are grammatically incorrect (e.g., "it's not a single color color"). This may be because that the model fails to prevent the generation of repeated words.

7 Conclusion

In this paper, we propose a novel knowledge-aware dialogue generation model TransDG, which is the first neural dialogue model that incorporates commonsense knowledge via transferring the abilities of utterance representation and knowledge selection from KBQA task. In addition, we propose a response guiding attention mechanism to enhance the input post understanding in encoder, and refine the knowledge selection by multi-step decoding to generate more appropriate and informative responses. Extensive experiments demonstrate the effectiveness of our model.

Acknowledgments

References

Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; and Ives, Z. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*. Springer. 722–735.

Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*, 1247–1250. AcM.

Bordes, A.; Usunier, N.; Chopra, S.; and Weston, J. 2015. Large-scale simple question answering with memory networks. *arXiv preprint arXiv:1506.02075*.

Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*, 1724–1734.

Fleiss, J. L. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin* 76(5):378.

Ghazvininejad, M.; Brockett, C.; Chang, M.-W.; Dolan, B.; Gao, J.; Yih, W.-t.; and Galley, M. 2018. A knowledge-grounded neural conversation model. In *AAAI*.

Gu, J.; Lu, Z.; Li, H.; and Li, V. O. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *ACL*, 1631–1640.

Hao, Y.; Zhang, Y.; Liu, K.; He, S.; Liu, Z.; Wu, H.; and Zhao, J. 2017. An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge. In *ACL*, 221–231.

Hu, S.; Zou, L.; Yu, J. X.; Wang, H.; and Zhao, D. 2018. Answering natural language questions by subgraph matching over knowledge graphs. *IEEE Transactions on Knowledge and Data Engineering* 30(5):824–837.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Li, J.; Galley, M.; Brockett, C.; Gao, J.; and Dolan, B. 2016a. A diversity-promoting objective function for neural conversation models. In *NAACL*, 110–119.

Li, J.; Galley, M.; Brockett, C.; Spithourakis, G. P.; Gao, J.; and Dolan, B. 2016b. A persona-based neural conversation model. In *ACL*, 994–1003.

Li, J.; Monroe, W.; Ritter, A.; Galley, M.; Gao, J.; and Jurafsky, D. 2016c. Deep reinforcement learning for dialogue generation. In *EMNLP*, 1192–1202.

Lian, R.; Xie, M.; Wang, F.; Peng, J.; and Wu, H. 2019. Learning to select knowledge for response generation in dialog systems. In *IJCAI*, 5081–5087.

Liu, S.; Chen, H.; Ren, Z.; Feng, Y.; Liu, Q.; and Yin, D. 2018. Knowledge diffusion for neural dialogue generation. In *ACL*, 1489–1498.

Long, Y.; Wang, J.; Xu, Z.; Wang, Z.; Wang, B.; and Wang, Z. 2017. A knowledge enhanced generative conversational service agent. In *Proceedings of the 6th Dialog System Technology Challenges (DSTC6) Workshop*.

Luo, K.; Lin, F.; Luo, X.; and Zhu, K. 2018. Knowledge base question answering via encoding of complex query graphs. In *EMNLP*, 2185–2194.

Manning, C.; Surdeanu, M.; Bauer, J.; Finkel, J.; Bethard, S.; and McClosky, D. 2014. The stanford corenlp natural language processing toolkit. In *ACL: System Demonstrations*, 55–60.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 311–318.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *EMNLP*, 1532–1543.

Serban, I. V.; Sordoni, A.; Bengio, Y.; Courville, A.; and Pineau, J. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*, 3776–3783.

Song, Y.; Yan, R.; Li, C.-T.; Nie, J.-Y.; Zhang, M.; and Zhao, D. 2018. An ensemble of retrieval-based and generation-based human-computer conversation systems. In *IJCAI*, 4382–4388.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Process-ing Systems*, 3104–3112.

Wu, Y.; Wei, F.; Huang, S.; Li, Z.; and Zhou, M. 2018. Response generation by context-aware prototype editing. *arXiv preprint arXiv:1806.07042*.

Xu, K.; Reddy, S.; Feng, Y.; Huang, S.; and Zhao, D. 2016. Question answering on freebase via relation extraction and textual evidence. In *ACL*, 2326–2336.

Yao, X., and Van Durme, B. 2014. Information extraction over structured data: Question answering with freebase. In *ACL*, volume 1, 956–966.

Yih, S. W.-t.; Chang, M.-W.; He, X.; and Gao, J. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *ACL*, 1321–1331.

Yu, M.; Yin, W.; Hasan, K. S.; Santos, C. d.; Xiang, B.; and Zhou, B. 2017. Improved neural relation detection for knowledge base question answering. In *ACL*, 571–581.

Zhou, H.; Young, T.; Huang, M.; Zhao, H.; Xu, J.; and Zhu, X. 2018. Commonsense knowledge aware conversation generation with graph attention. In *IJCAI*, 4623–4629.

Zhu, W.; Mo, K.; Zhang, Y.; Zhu, Z.; Peng, X.; and Yang, Q. 2017. Flexible end-to-end dialogue system for knowledge grounded conversation. *arXiv preprint arXiv:1709.04264*.