Proactive Human-Machine Conversation with Explicit Conversation Goals

Wenquan Wu, Zhen Guo, Xiangyang Zhou, Hua Wu, Xiyuan Zhang, Rongzhong Lian and Haifeng Wang
Baidu Inc., Beijing, China
{wuwenquan01,guozhenguozhen,zhouxiangyang,wu_hua}@baidu.com
{zhangxiyuan01,lianrongzhong,wanghaifeng}@baidu.com

Abstract

Though great progress has been made for human-machine conversation, current dialogue system is still in its infancy: it usually converses passively and utters words more as a matter of response, rather than on its own initiatives. In this paper, we take a radical step towards building a human-like conversational agent: endowing it with the ability of proactively leading the conversation (introducing a new topic or maintaining the current topic). To facilitate the development of such conversation systems, we create a new dataset named DuConv where one acts as a conversation leader and the other acts as the follower. The leader is provided with a knowledge graph and asked to sequentially change the discussion topics, following the given conversation goal, and meanwhile keep the dialogue as natural and engaging as possible. DuConv enables a very challenging task as the model needs to both understand dialogue and plan over the given knowledge graph. We establish baseline results on this dataset (about 270K utterances and 30k dialogues) using several state-of-the-art models. Experimental results show that dialogue models that plan over the knowledge graph can make full use of related knowledge to generate more diverse multi-turn conversations. The baseline systems along with the dataset are publicly available 1.

1 Introduction

Building a human-like conversational agent is one of long-cherished goals in Artificial Intelligence (AI) (Turing, 2009). Typical conversations involve exchanging information (Zhang et al., 2018), recommending something (Li et al., 2018), and completing tasks (Bordes et al., 2016), most of which rely on background knowledge. However, many dialogue systems only rely on utterances and responses as training data, without explicitly exploiting knowledge associated with them, which sometimes results in uninformative and inappropriate responses (Wang et al., 2018). Although there exist some work that use external background knowledge to generate more informative responses (Liu et al., 2018; Yin et al., 2015; Zhu et al., 2017), these systems usually generate responses to answer questions instead of asking questions or leading the conversation. In order to solve the above problems, some new datasets have been created, where external background knowledge is explicitly linked to utterances (Dinan et al., 2019; Moghe et al., 2018), to facilitate the development of knowledge aware conversation models. With these datasets, conversation systems can be built to talk with humans given a topic based on the provided external knowledge. Unlike task-oriented systems (Bordes et al., 2016), these conversation systems don’t have an explicit goal to achieve, thereof not able to plan over the background knowledge.

In this paper, we take a radical step towards building another type of human-like conversational agent: endowing it with the ability of proactively leading the conversation with an explicit conversation goal. To this end, we investigate learning a proactive dialogue system by planning dialogue strategy over a knowledge graph. Our assumption is that reasoning and planning with knowledge are the keystones to achieve proactive conversation. For example, when humans talk about movies, if one person learns more about some movies, he/she usually leads the conversation based on one or more entities in the background knowledge and smoothly changes the topics from one entity to another. In this paper, we mimic this process by setting an explicit goal as a knowledge path “[start] → topic_a → topic_b”,

---

1 https://github.com/PaddlePaddle/models/tree/develop/PaddleNLP/Research/ACL2019-DuConv
which means that one person leads the conversation from any starting point to \textit{topic}_a and then to \textit{topic}_b. Here \textit{topic} represents one entity in the knowledge graph.

With this in mind, we first build a knowledge graph which combines factoid knowledge and non-factoid knowledge such as comments and synopsis about movies. To construct the knowledge graph, we take a factoid knowledge graph (KG) as its backbone and align unstructured sentences from the non-factoid knowledge with entities. Then we use this KG to facilitate knowledge path planning and response generation, as shown in Figure 1. Based on this knowledge graph, we create a new knowledge-driven conversation dataset, namely the Baidu Conversation Corpus (DuConv) to facilitate the development of proactive conversation models. Specifically, DuConv has around 30k multi-turn conversations and each dialog in the DuConv is created by two crowdsourced workers, where one plays the role of the conversation leader and the other one acts as the conversation follower. At the beginning of each conversation, the leading player is assigned with an explicit goal, i.e., to sequentially change the conversation topic from one to another, meanwhile keeping the conversation as natural and engaging as possible. The conversation goal is a knowledge path comprised of two topics and structured as \texttt{[start] \rightarrow \textit{topic}_a \rightarrow \textit{topic}_b} and the leading player is also provided with related knowledge of these two topics. For each turn in the conversation, the leading player needs to exploit the provided knowledge triplets to plan his/her conversation strategy and construct responses to get closer to the target topic, while the follower only needs to respond according to the contexts without knowing the goal.

Figure 1 illustrates one example dialog in DuConv. It can be seen that DuConv provides a very challenging task: the conversational agents have to fully exploit the provided knowledge to achieve the given goal. To test the usability of DuConv, we propose a knowledge-aware neural dialogue generator and a knowledge-aware retrieval-based dialogue system, and investigate their effectiveness. Experimental results demonstrate that our proposed methods can proactively lead the conversation to complete the goal and make more use of the provided knowledge.

To the best of our knowledge, it is the first work that defines an explicit goal over the knowledge graph to guide the conversation process, making the following contributions:

- A new task is proposed to mimic the action of humans that lead conversations over a knowledge graph combining factoid and non-factoid knowledge, which has a wide application in real-world but is not well studied.
- A new large-scale dataset named DuConv is constructed and released to facilitate the development of knowledge-driven proactive dialogue systems.
- We propose knowledge-aware proactive dialogue models and conduct detailed analysis over the datasets. Experimental results demonstrate that our proposed methods make full use of related knowledge to generate more diverse conversations.

2 Related Work

Our related work is in line with two major research topics, \textit{Proactive Conversation} and \textit{Knowledge Grounded Conversation}.
2.1 Proactive Conversation

The goal of proactive conversation is endowing dialogue systems with the ability of leading the conversation. Existing work on proactive conversation is usually limited to specific dialogue scenarios. Young et al. (2013), Mo et al. (2016) and Bordes et al. (2018) proposed to complete tasks more actively, like restaurant booking, by actively questioning/clarifying the missing/ambiguous slots. Besides the task-oriented dialogue systems, researchers have also investigated building proactive social bots to make the interaction more engaging. Wang et al., (2018) explored to ask good questions in open-domain conversational systems. Li et al., (2018) enabled chatbots to recommend films during chitchatting. Unlike the existing work, we proposed to actively lead the conversation by planning over a knowledge graph with an explicit goal. We also create a new dataset to facilitate the development of such conversation systems.

2.2 Knowledge Grounded Conversation

Leveraging knowledge for better dialogue modeling has drawn lots of research interests in past years and researchers have shown the multi-fold benefits of exploiting knowledge in dialogue modeling. One major research line is using knowledge to generate engaging, meaningful or personalized responses in chitchatting (Ghazvininejad et al., 2018; Vougiouklis et al., 2016; Zhou et al., 2018a; Zhang et al., 2018). In addition to proposing better conversation models, researchers also released several knowledge grounded datasets (Dinan et al., 2019; Moghe et al., 2018). Our work is most related to Mogh et al., (2018) and Dinan et al., (2019), where each utterance in their released datasets is aligned to the related knowledge, including both structured triplets and unstructured sentences. We extend their work, by including the whole knowledge graph into dialogue modeling and propose a new task of proactively leading the conversation via planning over the knowledge graph in this paper.

3 DuConv

In this section, we describe the creation of DuConv in details. It contains four steps: knowledge crawling, knowledge graph construction, conversation goal assignment, and conversation crowdsourcing. We limit the dialogue topics in DuConv to movies and film stars, and crawl this related knowledge from the internet. Then we build our knowledge graph with these crawled data. After constructing our knowledge graph, we randomly sample two linked entities to construct the conversation goal, denoted as “[start] → topic_a → topic_b”, and ask two annotators to conduct knowledge-driven conversations, with one playing as the conversation leader and the other one playing as the follower. The leader needs to change the conversation topics following the conversation goal and meanwhile keep the conversation as engaging as possible. All those conversations are recorded and around 30k conversations are finally used in DuConv after filtering dirty/offensive parts. Table 1 summarizes the main information about DuConv.

|                  | # dialogs | # utterances |
|------------------|-----------|--------------|
|                  | 29858     | 270399       |
| average # utterances per dialog | 9.1       |              |
| average # words per utterance    | 10.6      |              |
| average # words per dialog       | 96.2      |              |
| average # knowledge per dialogue | 17.1      |              |

Table 1: Overview of the conversation dataset DuConv.

3.1 Knowledge Crawling

We crawled the related knowledge information from the website MTime.com\(^2\), which records the information of most films, heroes, and heroines in China. We collect both structured knowledge (such as “Harry Potter” is “directed by” “Chris Columbus”) as well as unstructured knowledge including short comments and synopsis. We filter out the dirty or offensive information and further normalize some of the numbers (such as the values of rating) into discrete symbols (good, fair, bad) to facilitate the use of this kind of knowledge. In summary, we crawl more than 91k films and 51k film stars, resulting in about 3.6 million knowledge triplets, the accuracy of which is over 97%\(^3\).

3.2 Knowledge Graph Construction

After the raw data collection, we construct a knowledge graph. Our knowledge graph is comprised of multiple SPO (Subject, Predicate, Ob-
Table 2: Overview of the knowledge graph in DuConv.

| # entities   | 143627 |
|--------------|--------|
| # movies     | 91874  |
| # person names| 51753 |
| # properties | 45     |
| average # spo per entity | 25     |

The workers are collected from a Chinese crowdsourcing platform http://test.baidu.com/. The workers are paid 2.5 Chinese Yuan per conversation.
score measured by our knowledge-aware response ranker is defined as \( p(l = 1 | Y, X, K, G) \). As shown in Figure 2(a), our knowledge-aware response ranker consists of four major parts, i.e., the context-response representation module (Encoder), the knowledge representation module (Knowledge Encoder), the knowledge reasoning module (Knowledge Reasoner) as well as the matching module (Matcher).

The Encoder module has the same architecture as BERT (Devlin et al., 2018), it takes the context \( X \) and candidate response \( Y \) as segment_a and segment_b in BERT, and leverages a stacked self-attention to produce the joint representation of \( X \) and \( Y \), denoted as \( xy \). Each related knowledge \( \text{knowledge}_i \) is also encoded as vector representations in the Knowledge Encoder module using a bi-directional GRU (Chung et al., 2014), which can be formulated as \( k_i = [\overrightarrow{h_T}; \overleftarrow{h_0}] \), where \( T \) denotes the length of knowledge, \( \overrightarrow{h_T} \) and \( \overleftarrow{h_0} \) represent the last and initial hidden states of the two directional GRU respectively. The dialogue goal is also combined with the related knowledge in order to fuse that information into response ranking.

To jointly consider context, dialogue goal and knowledge in response ranking, we make the context-response representation \( xy \) attended to all knowledge vectors \( k_i \) and get the attention distribution. For simplicity, the dialogue goal was treated as part of the knowledge used in the conversation.

\[
p(k_i | x, y) = \frac{\exp(xy \cdot k_i)}{\sum_j \exp(xy \cdot k_j)} \tag{1}
\]

and fuse all related knowledge information into a single vector \( k_c = \sum_i p(k_i | x, y) * k_i \). We view \( k_c \) and \( xy \) as the information from knowledge side and dialogue side respectively, and fuse those two kinds of information into a single vector via concatenation, then finally calculate the matching probability as:

\[
p(l = 1 | X, Y, K, G) = \text{sigmoid}(\text{MLP}(xy; k_c)) \tag{2}
\]

Our knowledge-aware response ranker differs from most existing work in jointly considering the previous dialogue context, the dialogue goal as well as the related knowledge, which enables our model to better exploit knowledge to achieve the conversation goal.

### 4.2 Generation-based Model

To generate a knowledge-driven dialogue response, we enhance the vanilla seq2seq model with an extra knowledge selection paradigm, Figure 2(b) demonstrates the structure of our knowledge-aware generator, which is comprised of four parts: the Utterance Encoder, the Knowledge Encoder, the Knowledge Manager and the Decoder.

For each given dialogue context \( X \), along with the dialogue goal \( G \) and related knowledge \( K \), our knowledge-aware generator first encodes all input information as vectors in the Utterance Encoder and Knowledge Encoder. The encoding method in those two modules also uses bi-directional GRUs, akin to that in the retrieval-based method. Especially, the dialogue context \( X \) and dialogue goal \( G \) are fused into the same vector \( x \) by sequentially concatenate \( G \) and \( X \) into a single sentence, then feed to the encoder.

After encoding, our knowledge-aware generator starts to plan its dialogue strategy by con-
considering which knowledge would be appropriate next. Practically, the generator can also conduct knowledge selection via attention mechanism as in the retrieval-based method. However, to force the model to mimic human in knowledge selection, we introduce two different distributions: 1) the prior distribution \( p(k_i|x) \) and the posterior distribution \( p(k_i|x,y) \). We take the prior distribution \( p(k_i|x) \) as the knowledge reasoned by machines and the posterior distribution \( p(k_i|x,y) \) as the knowledge reasoned by humans, and then force the machine to mimic human by minimizing the KL-DivLoss between those two distributions, which can be formulated as:

\[
\begin{align*}
    p(k_i|x,y) &= \frac{\exp(k_i \cdot MLP([x; y]))}{\sum_{j=1}^{N} \exp(k_j \cdot MLP([x; y]))} \quad (3) \\
    p(k_i|x) &= \frac{\exp(k_i \cdot x)}{\sum_{j=1}^{N} \exp(k_j \cdot x)} \quad (4) \\
    L_{KL}(\theta) &= \frac{1}{N} \sum_{i=1}^{N} p(k_i|x,y) \log \frac{p(k_i|x,y)}{p(k_i|x)} \quad (5)
\end{align*}
\]

Then, the BOW loss is defined to minimize:

\[
L_{BOW}(\theta) = -\frac{1}{m} \sum_{t=1}^{m} \log p(y_t|k_c) \quad (7)
\]

In summary, the final loss of our generative model is:

\[
L(\theta) = L_{KL}(\theta) + L_{NLL}(\theta) + L_{BOW}(\theta) \quad (8)
\]

5 Experiments

5.1 Setting

Our proposed models are tested under two settings: 1) automatic evaluation and 2) human evaluation. For automatic evaluation, we leverage several common metrics including BLEU, PPL, F1, DISTINCT1/2 to automatically measure the fluency, relevance, diversity etc. In our setting, we ask each model to select the best response from 10 candidates, same as previous work (Zhang et al., 2018). Those 10 candidate responses are comprised of one true response generated by human-beings and nine randomly sampled ones from the training corpus. We measure the performance of all models using Hits@1 and Hits@3, same as Zhang et al., (2018). Furthermore, we also evaluate the ability of exploiting knowledge of each model by calculating knowledge precision/recall/F1 scores.

The human evaluation is conducted at two levels, i.e., the turn-level human evaluation and the dialogue-level human evaluation. The turn-level human evaluation is similar to automatic evaluation. Given the dialogue context, the dialogue goal as well as the related knowledge, we require each model to produce a response according to the dialogue context. The responses are evaluated by three annotators in terms of fluency, coherence, informativeness, and proactiveness. The coherence measures the relevance of the response and the proactiveness measures if the model can successfully introduce new topics without destructing the fluency and coherence.

The dialogue-level evaluation is much more challenging. Given a conversation goal and the related knowledge, each model is required to talk with a volunteer and lead the conversation to achieve the goal. For each model, 100 dialogues are generated. The generated conversations are then evaluated by three persons in terms of two aspects: goal completion and coherence. The goal
Table 3: Automatic evaluation results. klg. and norm stands for knowledge and normalized here. S2S stands for the vanilla sequence-to-sequence model.

| Methods                        | Hits@1 | Hits@3 | PPL       | F1/BLEU1/BLEU2 | DISTINCT 1&2 | knowledge P/R/F1 |
|--------------------------------|--------|--------|-----------|---------------|--------------|-----------------|
| retrieval w/o klg.             | 45.84% | 72.86% | 33.08 / 0.280 / 0.147 | 0.121 / 0.376 | 86.90 / 39.30 / 13.73 |
| retrieval w/ klg.              | 46.74% | 75.32% | 33.12 / 0.282 / 0.146 | 0.122 / 0.388 | 85.54 / 37.93 / 13.47 |
| norm retrieval                 | 50.92% | 79.02% | 34.73 / 0.291 / 0.156 | 0.118 / 0.375 | 9.76 / 40.23 / 15.22 |
| S2S w/o klg.                   | 24.88% | 49.64% | 26.43 / 0.187 / 0.100 | 0.032 / 0.088 | 4.59 / 30.00 / 7.73 |
| S2S w/ klg.                    | 30.58% | 57.52% | 32.19 / 0.226 / 0.140 | 0.064 / 0.168 | 8.49 / 36.31 / 15.85 |
| norm S2S                       | 31.26% | 55.12% | 39.94 / 0.283 / 0.186 | 0.093 / 0.222 | 7.52 / 42.74 / 12.34 |
| generation w/o klg.            | 25.52% | 50.14% | 28.52 / 0.29 / 0.154 | 0.032 / 0.075 | 6.18 / 27.48 / 9.86 |
| generation w/ klg.             | 31.90% | 58.44% | 36.21 / 0.32 / 0.169 | 0.049 / 0.144 | 8.67 / 35.90 / 13.62 |
| norm generation                | 32.50% | 58.50% | 41.84 / 0.347 / 0.198 | 0.057 / 0.155 | 9.88 / 38.02 / 15.27 |

Table 4: Turn-level and dialogue-level human evaluation results

| methods                          | fluency (0,1,2) | coherence (0,1,2) | informativeness (0,1,2) | proactivity (-1,0,1) | goal complete (0,1,2) | coherence (0,1,2,3) |
|----------------------------------|----------------|------------------|-------------------------|----------------------|-----------------------|---------------------|
| norm retrieval                   | 1.93           | 1.41             | 0.86                    | 0.80                 | 0.90                  | 1.92                |
| norm generation (S2S)            | **2.00**       | **1.89**         | 0.74                    | 0.86                 | 1.14                  | **2.01**            |
| norm generation                  | 1.87           | 1.61             | **1.10**                | **0.87**             | **1.22**              | **2.03**            |

5.2 Comparison Models

The compared models contain the vanilla seq2seq model, our proposed retrieval-based model as well as our proposed generation-based model. Moreover, we normalize the train/valid/test data by replacing the specific two topics in the knowledge path with “topic_a” and “topic_b” respectively. Models using such normalized corpora are named as normalized models. To test the effectiveness of knowledge, we set up one ablation experiment, which removes all the knowledge triplets by replacing them with “UNK, UNK, UNK”.

5.3 Model Training

All models are implemented using PaddlePaddle and pytorch (Paszke et al., 2017), trained on a single GPU of NVIDIA Tesla K40. We set the vocabulary size to 30k for both retrieval-based and generation based methods. All hidden sizes, as well as embedding size, are set to 300, and the word embedding layer is initialized via word2vec trained on a very large corpus. We apply Adam optimize for model training and the beam size for generative models are set to 10 during decoding.

5.4 Results

Table 3 and Table 4 summarize the experimental results on automatic evaluation and human evaluation. For human evaluation, we only evaluate the normalized models since they achieved better performances on our dataset. All human evaluations are conducted by three persons, where the agreement ratio (Fleiss’ kappa (Fleiss et al., 1971)) is from 0.37 to 0.86, with the lowest agreement on multi-turn coherence and others all above 0.6. More details of these measures are available in the Appendix.

6It is an open source deep learning platform (https://paddlepaddle.org) developed by Baidu. Our code and data are available at https://github.com/PaddlePaddle/models/tree/develop/PaddleNLP/Research/ACL2019-DuConv.

7https://radimrehurek.com/gensim/models/word2vec.html
Table 5: Analysis on goal completion and knowledge exploitation.

| distribution statistics | norm generation | norm seq2seq | norm retrieval |
|-------------------------|-----------------|--------------|----------------|
| goal completion         |                 |              |                |
| 0                       | 21%             | 14%          | 25%            |
| 1                       | 35%             | 26%          | 59%            |
| 2                       | 43%             | 29%          | 15%            |
| knowledge used          |                 |              |                |
| # triplets              | 2.46            | 1.51         | 2.28           |
| # properties            | 27              | 20           | 25             |

It can be seen that the retrieval-based model and the generation-based model have significantly different performances in terms of automatic evaluation and human evaluations. Retrieval-based model works better on Hits@K, however worse on F1 and BLEU compared to the generation-based model. This is perhaps caused by that fact that they are optimized on different metrics. For human evaluation, it can be observed that the retrieval-based method is apparently worse than generation-based models. This is because the retrieved candidates limit the potential of the retrieval-based model. We also found that the methods using knowledge outperform those without using knowledge, which confirms the benefits of using background knowledge. It is very interesting that normalizing the "topic_a" and "topic_b" can significantly improve the performance for all models because of their generalization capability over the knowledge.

From the human evaluation, we found that our proposed generation methods outperform the baseline Seq2Seq model and the retrieval model, especially in terms of turn-level informativeness and proactivity, and dialogue-level goal completion and coherence. In order to further analyze the relationship between informativeness and goal completion, the detailed distribution of goal completion scores and the numbers of used knowledge triplets are shown in Table 5. From this table, it can be seen that our proposed generation model can exploit more knowledge to achieve the conversation goal (much higher rate on score “2”), making the conversation more engaging and coherent. This demonstrates the effectiveness of the knowledge posterior/prior distribution learning. Although the baseline Seq2Seq model can also has good goal completion capability, it usually only uses knowledge directly related to the conversation goal in the conversation process (much higher rate over score “1”), making the conversation usually dull.

However, for the dialogue-level human evaluation, there are still 15% to 20% of conversation goals not achieved. The reason may be that our models (both retrieval and generation) have no explicit multi-turn policy mechanism to control the whole conversation flow, which is left for future research.

6 Case Study

Figure 3 shows the conversations generated by the models via conversing with humans, given the conversation goal and the related knowledge. It can be seen that our knowledge-aware generator can choose appropriate and more knowledge for diverse conversation generation. Even though the retrieval-based method can also produce knowledge-grounded responses, the used knowledge is often wrong. Although the seq2seq model can smoothly achieve the given knowledge goal, it always generates generic responses using safe dialogue strategy, as the mentioned knowledge is much smaller than our proposed knowledge-aware generator, making the generated conversation less diverse and sometimes dull.

7 Conclusion

In this paper, we build a human-like conversational agent by endowing it with the ability of proactively leading the conversation. To achieve this goal, we create a new dataset named DuConv. Each dialog in DuConv is created by two crowd-sourced workers, where one acts as the conversation leader and the other acts as the follower. The leader is provided with a knowledge graph and asked to sequentially change the discussed topics following the given conversation goal, and meanwhile, keep the dialogue as natural and engaging as possible. We establish baseline results on DuConv using several state-of-the-art models. Experimental results show that dialogue models that plan over knowledge graph can make more full use of related knowledge to generate more
diverse conversations. Our dataset and proposed models are publicly available, which can be used as benchmarks for future research on constructing knowledge-driven proactive dialogue systems.

Acknowledgement

We sincerely thank the PaddlePaddle development team for helping us build the baseline models. We also would like to thank Ying Chen and Na Chen for helping us to collect the dataset through crowd-sourcing. This work was supported by the Natural Science Foundation of China (No.61533018).

References

Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2016. Learning end-to-end goal-oriented dialog. arXiv preprint arXiv:1605.07683.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.

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

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of wikipedia: Knowledge-powered conversational agents. In International Conference on Learning Representations.

J.L. Fleiss et al. 1971. Measuring nominal scale agreement among many raters. Psychological Bulletin, 76(5):378–382.

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.

Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards deep conversational recommendations. In Advances in Neural Information Processing Systems, pages 9748–9758.

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), pages 1489–1498.

Kaixiang Mo, Yu Zhang, Shuangyin Li, Jiajun Li, and Qiang Yang. 2018. Personalizing a dialogue system with transfer reinforcement learning. In Thirty-Second AAAI Conference on Artificial Intelligence.

Nikita Moghe, Siddhartha Arora, Suman Banerjee, and Mitesh M. Khapra. 2018. Towards exploiting background knowledge for building conversation systems. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2322–2323.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In NIPS-W.
Alan M Turing. 2009. Computing machinery and intelligence. In *Parsing the Turing Test*, pages 23–65.

Pavlos Vougiouklis, Jonathon Hare, and Elena Simperl. 2016. A neural network approach for knowledge-driven response generation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3370–3380.

Yansen Wang, Chenyi Liu, Minlie Huang, and Liqiang Nie. 2018. Learning to ask questions in open-domain conversational systems with typed decoders. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2193–2203.

Yu Wu, Wei Wu, Ming Zhou, and Zhoujun Li. 2017. Sequential match network: A new architecture for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 372–381.

Lili Yao, Yaoyuan Zhang, Yansong Feng, Dongyan Zhao, and Rui Yan. 2017. Towards implicit content-introducing for generative short-text conversation systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2190–2199.

Jun Yin, Xin Jiang, Zhengdong Lu, Lifeng Shang, Hang Li, and Xiaoming Li. 2015. Neural generative question answering. *CoRR*, abs/1512.01337.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. 2013. Pomd-p-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–664, Vancouver, Canada.

Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018a. Commonsense knowledge aware conversation generation with graph attention. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4623–4629.

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

Wenya Zhu, Kaixiang Mo, Yu Zhang, Zhangbin Zhu, Xuezheng Peng, and Qiang Yang. 2017. Flexible end-to-end dialogue system for knowledge grounded conversation. *CoRR*. 

3803
Appendix

A. Turn-level Human Evaluation Guideline

Fluency measures if the produced response itself is fluent:
- score 0 (bad): unfluent and difficult to understand.
- score 1 (fair): there are some errors in the response text but still can be understood.
- score 2 (good): fluent and easy to understand.

Coherence measures if the response can respond to the context:
- score 0 (bad): not semantically relevant to the context or logically contradictory to the context.
- score 1 (fair): relevant to the context as a whole, but using some irrelevant knowledge, or not answering questions asked by the users.
- score 2 (good): otherwise.

Informativeness measures if the model makes full use of knowledge in the response:
- score 0 (bad): no knowledge is mentioned at all.
- score 1 (fair): only one triplet is mentioned in the response.
- score 2 (good): more than one triplet is mentioned in the response.

Proactivity measures if the model can introduce new knowledge/topics in conversation:
- score -1 (bad): some new topics are introduced but irrelevant to the context.
- score 0 (fair): no new topics/knowledge are used.
- score 1 (good): some new topics relevant to the context are introduced.

B. Dialogue-level Human Evaluation Guideline

Goal Completion measures how good the given conversation goal is finished:
- score 0 (bad): neither “topic_a” nor “topic_b” is mentioned in the conversation.
- score 1 (fair): “topic_a” or “topic_b” is mentioned, but the whole dialogue is very boring and less than 3 different knowledge triplets are used.
- score 2 (good): both “topic_a” or “topic_b” are mentioned and more than 2 different knowledge triplets are used.

Coherence measures the overall fluency of the whole dialogue:
- score 0 (bad): over 2 responses irrelevant or logically contradictory to the previous context.
- score 1 (fair): only 2 responses irrelevant or logically contradictory to the previous context.
- score 2 (good): only 1 response irrelevant or logically contradictory to the previous context.
- score 3 (perfect): no response irrelevant or logically contradictory to the previous context.