
Hsuan Su∗⋄ Shachi H Kumar⋄ Sahisnu Mazumder⋄ Wenda Chen⋄ Ramesh Manuvinakurike⋄ Eda Okur⋄ Saurav Sahay⋄ Lama Nachman⋄ Shang-Tse Chen∗ Hung-yi Lee∗ National Taiwan University∗ Intel Labs⋄
hsuansu.96@gmail.com

Abstract

With the power of large pretrained language models, various research works have integrated knowledge into dialogue systems. The traditional techniques treat knowledge as part of the input sequence for the dialogue system, prepending a set of knowledge statements in front of dialogue history. However, such a mechanism forces knowledge sets to be concatenated in an ordered manner, making models implicitly pay imbalanced attention to the sets during training. In this paper, we first investigate how the order of the knowledge set can influence autoregressive dialogue systems’ responses. We conduct experiments on two commonly used dialogue datasets with two types of transformer-based models and find that models view the input knowledge unequally. To this end, we propose a simple and novel technique to alleviate the order effect by modifying the position embeddings of knowledge input in these models. With the proposed position embedding method, the experimental results show that each knowledge statement is uniformly considered to generate responses.

1 Introduction

Transformer-based (Vaswani et al., 2017) pretrained language models are widely used to build dialogue systems (Zhang et al., 2020; Xu et al., 2021; Komeili et al., 2021; Roller et al., 2020; Thoppilan et al., 2022; Rae et al., 2021; Chen et al., 2021; Ham et al., 2020; Hosseini-Asl et al., 2020; Bao et al., 2021). In addition to general-purpose dialogue systems, many specialized dialogue systems have been proposed. Representative examples include personalized dialogue systems (Wolf et al., 2019; Zhang et al., 2018; Wu et al., 2021; Cao et al., 2022; Song et al., 2020), knowledge-grounded dialogue systems (Dinan et al., 2019; Kim et al., 2021; Tao et al., 2021; Cai et al., 2020; Liu et al., 2021), and prompting dialogue systems (Su et al., 2022).

Figure 1: The order effect illustration. Models’ responses are influenced by the order of the input knowledge set.

To build specialized dialogue systems, integrating additional information into the input sequence is necessary. Wolf et al. (2019) prepend persona sentences to personalize the history; while Su et al. (2022); Dinan et al. (2020); Keskar et al. (2019); Xu et al. (2020a) prepending task-specific signals to prompt and control the model.

These methods prepend additional information in front of the history as a sequence for models’ input. Furthermore, the approach generates an unnecessary order among equal knowledge sets since the knowledge is connected in the sequence. Thus models might be influenced by the order and generate imbalanced responses.

Previous works focus on how perturbations in dialog history affect models’ responses (Sankar et al., 2019; O’Connor and Andreas, 2021; Sinha et al., 2021; Lampinen et al., 2022; Webson and Pavlick, 2021; Xu et al., 2020b; Khandelwal et al., 2018). They conduct many experiments and measure the effect of perturbations from the aspect of response quality and information theory to show that these language models are robust and not sensitive to the perturbations in input history. However, dialog history and knowledge are inherently different aspects of a conversation. Dialog history has a temporal property, i.e., the topic and specificity of conversation change as the dialog progresses, whereas knowledge facts are information referenced to generate a response. Although the perturbation in history does not influence the
I live in NYC
I have a dog
I love baseball
I like to go shopping
I'm 22 years old
What's your plan today?

Figure 2: Input format for GPT-series models. The position ids do not treat knowledge equally but as a sequence. The updated position embeddings show our proposed method, where each knowledge statement is encoded with its own position embeddings, hence, models can treat each input sentence equally during training. The same color of blocks indicates using the same layer to generate embeddings.

results generated by the model (Sankar et al., 2019; O’Connor and Andreas, 2021), in our early observation, we found that prepending knowledge influences models’ responses. For example, Figure 1 demonstrates an example where the model exhibits imbalanced attention to input knowledge, and the order of knowledge influences the generated responses. This might cause the model to generate inappropriate responses since it attends to knowledge that might not be relevant to a dialog context. The contributions of this work are as follows:

- We conduct experiments across two typical methods and two models on multiple datasets to show that the order of knowledge sentences does affect generated responses.

- We propose a simple approach to alleviate this sentence-level order effect by manipulating the position embedding layers.

2 Knowledge-grounded Dialogue Methods

In this work, we study the order effect in TransferTransfo (Wolf et al., 2019), which is a state-of-the-art knowledge-grounded method. We train TransferTransfo on two datasets and measure the sentence-level order effect on the test datasets.

2.1 TransferTransfo

The TransferTransfo architecture is built on top of GPT-series models, which simply concatenates the knowledge sets and context in a single sequence, putting the reply at the end. To help models distinguish speakers and position of input tokens, it builds three parallel input sequences for word, position, and segments, and fuses them into a single sequence. For the loss function, in addition to a language modeling loss, a next sentence prediction loss is added. The total loss is the weighted sum of the 1) language modeling loss, which is computed as the cross-entropy loss between the predicted logits and the ground truth response and 2) the next-sentence prediction loss, which is a classification loss to distinguish the ground truth response from distractors that are randomly sampled from the dataset.

In the original TransferTransfo implementation, the authors have already pointed out that the order of the knowledge set influences the model’s performance. To this end, they augment training data by permuting the knowledge sets several times.

2.2 Experimental Setups

We conduct experiments on two datasets:

- **Persona-Chat** (Zhang et al., 2018): This persona-grounded dialogue dataset consists of crowdsourced dialogues between a pair of annotators provided with 4-5 persona statements each.

- **Topical-Chat** (Gopalakrishnan et al., 2019): This is a knowledge-grounded dialogue dataset, where the dialogs are constructed by a pair of annotators conversing about specific topics. The annotators are provided with wiki data with 4-5 facts as knowledge sources.

In our experimental setup, we shuffle the knowledge set’s order 50 times during testing and implement TransferTransfo on GPT (Radford et al., 2018) and GPT-2 (Radford et al., 2019) models.

3 The Order Effect of the Knowledge Set

Models are said to have an order effect of input if the generated responses are sensitive and influenced by order of input sequence. Previous works (Sankar et al., 2019; O’Connor and Andreas, 2021; Sinha et al., 2021; Lampinen et al., 2022; Webson
and Pavlick, 2021; Xu et al., 2020b; Khandelwal et al., 2018) focus on whether perturbation in dialogue history affect models’ responses. In this work, to be more specific, we investigate if sentence level change in the order of input knowledge sets will result in substantial semantic differences in the generated responses.

3.1 The Order Effect Measurement

To address the sentence-level order effect of the input knowledge set in models, we aim to measure the semantic difference given different orders of knowledge sentences. It is intuitive to measure if the response content is influenced by knowledge sets order. In other words, we measure the distribution of response-knowledge relationship in different positions. We build a Natural Language Inference (NLI) classifier to evaluate the degree of entailment between responses and each knowledge in the set.

The Natural Language Inference Classifier is built with BERT model (Devlin et al., 2019), trained on the Dialogue NLI dataset (Welleck et al., 2019), which is built on top of Persona-Chat dataset (Zhang et al., 2018). The annotators label the relationship between persona and response in Persona-Chat with entail, neutral, and contradict classes.

3.2 Results and Discussions for Order Effect

Figures 3 and 4 show the entailment scores of the response with each position of knowledge. Figure 3 presents the experiments of TransferTransfo with GPT and GPT-2 models across Persona-Chat and Topical-Chat datasets. Figure 4 shows the re-
Table 1: The results of measurements. The Max-Min of entailment are reported in 4 knowledge / 5 knowledge. The mean of quality across 50 runs are reported and standard deviation are reported in Appendix A.3.

In this section, we analyse the reason for the order effect in the GPT-series models and propose a method to alleviate the phenomenon. Figure 2 shows the input format of the classic GPT-series. There are three types of embeddings in the model: word embedding to capture the semantic meaning of each word, token embedding to represent the speaker and absolute position embedding that encodes position information of input sequence. Figure 2 shows that the position ids for each knowledge start from zero with different positional embedding layers. In this naive setting, knowledge of the set are treated equally and not input with the order during training.

4.1 Results and Discussion

In the same Figures 3 and 4, the RED lines demonstrate the entailment result after applying multiple position embedding. We observe that all the red lines, which are the GPT-series applied multiple position embeddings, are much smoother compared to BLUE lines in both figures. Furthermore, we report the difference between maximum and minimum entailment across the positions in Table 1. It shows that the difference is negligible after applying multiple position embeddings. This indicates that we can alleviate the order effect under models trained with with multiple position embedding. However, we also observed that on Figure 4 some red lines are still as steep as before, which means the order effect still exists. We think that the model trained only with LM loss treats knowledge like history and does not ground models on knowledge sets. Under this scenario, the multiple position embedding doesn’t work well.

For the measurement of quality, Table 1 shows the perplexity, coherence, and diversity. The details are included in Appendix A.2. We found tiny drops between origin and multiple position embedding. More specifically, our proposed method does not crash the models and can still make models generate plausible responses.

5 Conclusions

In this paper, we investigate whether the order of knowledge set will influence dialogue models’ responses. Our experiments across several datasets show that the GPT-series models unfairly pay attention to the knowledge set and are influenced by order of knowledge. To solve this problem, we study
the reason for the phenomenon and propose simple method to alleviate the order effect in models. The experimental results show that our approach reduces the order effect and makes the model select the knowledge uniformly.

Limitations

This work has potential limitations:

- We found that on the Figure 3 and 4, The entailment of the methods after applying multiple position embedding (RED lines) are sometimes lower than origin methods (BLUE lines). This is not meet our expectations since we don’t want our method to decrease performance. In our opinion, we think the reason might be the embedding method has never been seen before during the pretraining of models, which requires the model’s additional efforts to adapt the embedding, thus hurts the performance. We leave it as future work to be improved.

- We also found that the multiple position embedding does not work very well to alleviate the order effect in the LM loss-only settings. We have discussed this in previous sections. Since LM loss only does not help the model distinguish which parts in the input sequence are knowledge set and thus treat them the same as history. The multiple position embedding will not be trained finely to help the model distinguish. We also left this as a future work to be improved.

References


Xiang Gao, Yizhe Zhang, Michel Galley, Chris Brockett, and Bill Dolan. 2020. Dialogue response ranking-training with large-scale human feedback data. In EMNLP.


Urvashi Khandelwal, He He, Peng Qi, and Dan Jurafsky. 2018. Sharp nearby, fuzzy far away: How neural language models use context. In ACL.


A Appendix

A.1 Experimental Details

• Hyperparameters: For the Hyperparameters we use to conduct experiments, we follow TransferTransfo link https://github.com/huggingface/transfer-learning-conv-ai. They obtain these Hyperparameters by grid searching. More specifically, They finetuned the model with a batch size of 32 sequences, and finetune the models approximately 2 epochs over training dataset. They used Adam with a learning rate of 6.25e-5, and a coefficient of 2 on the LM loss when summing with the next-sentence prediction loss. The learning rate was linearly decayed to zero over the course of the training.

• Datasets: The link to download Persona-Chat https://parl.ai/docs/tasks.html#persona-chat and the train/valid/test split is 9907/1000/968 dialogues. For the link to download Topical-Chat https://github.com/alexa/Topical-Chat and the train/valid/test split is 8628/1078/1078 dialogues.

• Pretrained Models: For GPT model we use gpt-medium as our pretrained model and use microsoft/DialoGPT-medium as initial checkpoint for GPT-2 model.

A.2 Evaluation Metrics

In addition to entailment, we aimed to employ other metrics that are also important to measure a dialogue system.

Perplexity (Chen et al., 1998): Here we employed the pretrained GPT-2 language model GPT to judge if the output sentence C(x) was an acceptable sentence. The computation of Perplexity (Chen et al., 1998) is shown below.

$$PPL = \prod_{i=1}^{T} \frac{1}{GPT(C(x, D), |x|)^{1/T}}$$  

Coherence: We employed the DialogRPT (Gao et al., 2020) to calculate the coherence between conversation model’s output and the input context. DialogRPT (Gao et al., 2020) is a GPT2-based ranker that finetuned on 133M human feedback data. With the contrastive learning approach that
DialogRPT used. The ranker has better understanding on how relevant the response is for the given context. In our evaluation, we take the probability that output by DialogRPT coherence model \((human_{vs\_rand})\) as our coherence metric.

**Diversity:** BLEU score (Papineni et al., 2002) is a commonly used metric for automatically evaluating machine translation. However, the Self-BLEU (Zhu et al., 2018) score here was applied to measure the diversity of chatbot responses. Regarding one sentence as the prediction and the others as the reference, we can calculate BLEU score for every sentence, and the average is the Self-BLEU score. A lower Self-BLEU score implies more diversity of the chatbot responses.

### A.3 Standard Deviation of Quality Metrics

<table>
<thead>
<tr>
<th>Model</th>
<th>Method</th>
<th>Persona</th>
<th>Topical</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>TT.</td>
<td>LM.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Perplexity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GPT</td>
<td>Origin</td>
<td>0.23</td>
<td>0.27</td>
</tr>
<tr>
<td></td>
<td>Multi Pos</td>
<td>0.22</td>
<td>0.26</td>
</tr>
<tr>
<td>GPT-2</td>
<td>Origin</td>
<td>0.31</td>
<td>0.29</td>
</tr>
<tr>
<td></td>
<td>Multi Pos</td>
<td>0.28</td>
<td>0.23</td>
</tr>
<tr>
<td></td>
<td>Coherence</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GPT</td>
<td>Origin</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td></td>
<td>Multi Pos</td>
<td>0.002</td>
<td>0.001</td>
</tr>
<tr>
<td>GPT-2</td>
<td>Origin</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td></td>
<td>Multi Pos</td>
<td>0.002</td>
<td>0.002</td>
</tr>
<tr>
<td></td>
<td>Diversity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GPT</td>
<td>Origin</td>
<td>0.002</td>
<td>0.002</td>
</tr>
<tr>
<td></td>
<td>Multi Pos</td>
<td>0.002</td>
<td>0.002</td>
</tr>
<tr>
<td>GPT-2</td>
<td>Origin</td>
<td>0.002</td>
<td>0.002</td>
</tr>
<tr>
<td></td>
<td>Multi Pos</td>
<td>0.002</td>
<td>0.002</td>
</tr>
</tbody>
</table>

Table 2: The results of quality measurements. The standard deviation across 50 runs are reported.