

GE-Blender: Graph Based Knowledge Enhancement for Blender

Xiaolei Lian

*Beijing Zhong Ke Lian Yuan Technologies Co., Ltd.
Beijing, Sanlitun Street 100027, China*

LIANXIAOLEI@SAFEIS.CN

Chonglin Tang

*Beijing Zhong Ke Lian Yuan Technologies Co., Ltd.
Beijing, Sanlitun Street 100027, China*

TANGCHONGLIN@SAFEIS.CN

Corresponding Author: Xiaolei Lian

Copyright © 2023 Xiaolei Lian and Chonglin Tang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

In spite of the considerable accomplishments in open-domain dialogue generation, the presence of unseen entities can substantially impact the task, resulting in a performance decline in model-driven dialogue generation. While previous research has utilized retrieved knowledge of seen entities to augment model representation, a comprehensive exploration of unseen entities, including their potential co-occurrence and semantic associations within the entity category, remains largely uncharted. In this study, we propose an approach to address this challenge. We construct a graph by extracting entity nodes, enhancing the contextual representation of unseen entities by leveraging the 1-hop surrounding nodes of associated entities. Furthermore, we introduce a named entity tag prediction task to address the issue of unseen entities not present in the constructed graph. We conduct our experiments on an open dataset Wizard of Wikipedia and the empirical results indicate that our approach outperforms the state-of-the-art approaches on Wizard of Wikipedia.

Keywords: Dialogue generation, knowledge enhancement, Unseen entity, Wizard of wikipedia.

1. INTRODUCTION

Unseen entity problem leads to the failure of the model in recognizing semantics in open-domain dialogue generation [1], and some previous methods improved on this issue. Knowled-GPT uses reinforcement learning for knowledge selection [2], and feeds knowledge as input of the model. Knowledge Grounded Blender focuses on fixing the problem of how to supply the knowledge of seen entities [2]. It performs well in addressing this problem but overlooks scenarios involving unseen entities. Knowledge Enhanced Blender (KE-Blender) aims to enhance the representation of context with seen entities' knowledge [1]. It forces the model to generate the definition and hypernym of seen entities, then updates model parameters for context representation enhancement. KE-Blender gives a state-of-the-art performance on Wizard of Wikipedia. But if we replace the

currently seen entity to a semantically different unseen entity and keep the context unchanged, the context representation enhancement of KE-Blender will not work well.

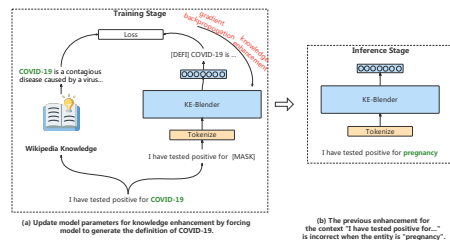


Figure 1: An illustration for wrong enhancement in the inference stage of KE-Blender.

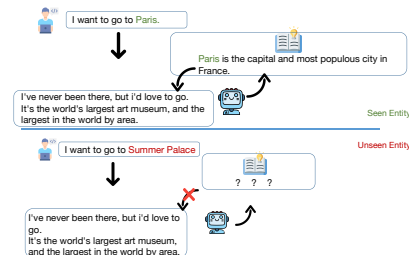


Figure 2: The generation result of KE-Blender on seen entity scenario and unseen entity respectively

Describe with an example in FIGURE 1, given the sentence "I have tested positive for COVID-19" and the definition of "COVID-19", the model will generate the response "I'm sorry to hear that." But if we replace the entity "COVID-19" to "pregnancy". The previous enhancement about the context "I have tested positive" with the definition of "COVID-19" will not positively impact the result. Take another proven example as the illustration in FIGURE 2, the KE-Blender does not work well when there are some unseen entity in the input sentence.

In order to solve this problem, we propose a novel method that enhances knowledge representation with graph knowledge and named entity recognition information in the dialogue generation task. We call our method as **Graph Based Knowledge Enhancement for Blender (GE-Blender)**.

Our approach have two modules: (1) **Graph Knowledge Enhancement**: We integrate the 1-hop node of the entity in the document searched from the graph constructed with dialog and retrieved knowledge into the dialogue generation task as knowledge enhancement; (2) **Named Entity Tags Enhancement**: For an entity that does not exist in the graph, we replace the entity with the entity tag generated by Named Entity Recognition(NER) [3], model and use the sequence replaced by tags as the target sequence to make the model generate.

Specifically, in the Graph Knowledge Enhancement, we first build a graph with dialogue data and retrieved knowledge in the dialogue generation dataset(such as Wizard of Wikipedia). Secondly, we extract 1-hop nodes of proper nouns in dialogue and flatten these nodes as an ordered word sequence by the edge weights. Finally, we regard the word sequence as the target sequence and make the model generate the sequence to enhance the context representation. In this way, we can use the global co-occurrence information of entities to enhance the representation of the context of entities. In the Named Entity Tags Enhancement, we use the NER model to predict the tags of named entities and replace named entities with their NER tags as a target sequence. Then we force the model to generate the replaced sequence. Through this task, the context representation can be further enhanced.

The contributions of this paper are as follows:

- We use graph knowledge to enhance the representation of the context of the entity, allowing the dialogue generation model to refer to global word co-occurrence information for solving the unseen entity problem.
- We use NER tags to guide the model to enhance the representation of the context of the entity.

Results on the Wizard of Wikipedia benchmark show that the proposed model brings performance improvement. The proposed method achieves **16.7** and **15.3** F1-score on Wizard of Wikipedia Test Seen and Test Unseen and PPL achieves **16.5** and **20.0** respectively. We release our code and dataset at <https://github.com/Fakekid/GE-Blender>.

2. RELATED WORK

Here we review the related works, including open-domain dialogue generation, knowledge enhancement approaches and unseen entity problems.

2.1 Open-domain Dialogue Generation

As the definition of open-domain dialogue generation in [4], the system must output a fluent, engaging, and meaningful natural language response based on the given previous dialogue turns. Different from task-based dialogue generation, open-domain dialogue generation does not need to have a clear goal, and the content of the interlocutor is often very casual hence open-domain dialogue will appear with unseen entities.

A common base for open-domain dialogue generation model is Blender: a standard Seq2Seq Transformer architecture with pre-training. Some recent research works use it as a basebone.

2.2 Knowledge Enhanced for Dialogue System

The knowledge enhancement method is relatively popular in recent years, and it has been applied in various fields, such as language model [5], text classification model [6], chat-bot model [1], etc. The knowledge enhancement method is to integrate the knowledge mastered by humans into the input data to allow the model to learn and enhance the representation ability of the model. Taking ERNIE [1], as an example, it extracts the entities in the input sentence, performs multi-head attention encoding, and fuses them with the hidden state of the token corresponding to the entity encoded by the default multi-head attention. For the text classification sense, STCKA [6], first extracts the concept of the entity in the input sentence from a knowledge base, then run the knowledge encoder layer. Finally, fuses the knowledge encoding and default multi-head attention output.

In the open-domain dialogue generation task, we integrate the 1-hop nodes of the entity in the graph as enhanced knowledge and integrate it into the input, so that the model can pay attention to the global co-occurrence words of the entity.

2.3 Unseen Entity Problems

The unseen entity problem is common in open-domain dialogue generation. What topics we talk about often are new and not in the dataset that the pre-trained model used. Some previous methods do not fix this problem, they didn't work well on unseen entities. The KG-Blender [2] concatenates the dialogue sentence and the seen entity's definition in retrieved knowledge as a knowledge-enhanced method. But this approach does not work when the entity does not appear in the training stage. The KE-Blender [1], uses two tasks for knowledge enhancement, one is to generate the definition of the seen entity in retrieved knowledge, and the other is to generate the hypernym of the seen entity. Both of these tasks can enhance the representation of the model, but when an unseen entity has no hypernym and definition, the KE-Blender also performs poorly in such cases.

For these hard problems, we use graph knowledge to enhance the representation of the dialogue generation model in the unseen entities sense.

3. APPROACH

3.1 Base Model

Named Entity Recognition Named Entity Recognition (NER) aims to extract and classify mentions of rigid designators from text, such as proper names, biological species, and temporal expressions [7]. Named Entity Recognition (NER) not only acts as a standalone tool for information extraction but also plays an essential role in a variety of natural language processing (NLP) applications, such as text understanding [8]. Therefore, the NER model can usually capture context information to complete accurate prediction tasks. In this regard, we can also use the characteristics of the NER task to strengthen the open-domain dialog generation model.

The task of NER can be expressed by the following formula:

$$p(y_i|x_i, y_{1:i-1}; \theta) = \sigma(NER(x_i, y_{1:i-1}; \theta)) \in \mathbb{R}^{\rho^l} \quad (1)$$

Where x_i is the i th token of the input sentence, and y_i is the i th named entity tag generated by the NER model.

We use Bi-LSTM as our base NER model and use the AllenNLP [9], library to complete the NER task. Compared to models based on the Transformer architecture, models built on the Bi-LSTM architecture offer faster inference speeds and nearly identical accuracy. Therefore, considering inference speed, we opted for the Bi-LSTM architecture. However, replacing the Bi-LSTM architecture with a Transformer architecture model may lead to performance improvements.

Blender We use Blender [10], as our basebone, Blender is a pre-trained dialogue generation model based on sequence-to-sequence transformer structure. It uses an encoder to encode the input sequence and uses the decoder to generate a decoded sequence.

Given a complete dialogue $U = \{u_1, \dots, u_{l-1}\}$, we concatenate all previous sequences of each response $U_{:i-1} = \{x_1, x_2, \dots, x_T\}$ as input sequence and the current reply $R = \{y_1, y_2, \dots, y_{T'}\}$ as the target sequence. The model first encodes the input sequence with the encoder:

$$h^{enc} = \text{TRM_ENC}(U) \in \mathbb{R}^{pl} \tag{2}$$

and then decodes the hidden states based on the encoder to generate the response sequence:

$$h_t^{dec} = \text{TRM_DEC}(h_{enc}, y_{1:t-1}) \in \mathbb{R}^{pl} \tag{3}$$

Where the p, l is the length of vocabulary and the sentence length.

In the t th step of the decoder, h_{enc} and previous output tokens $y_{1:t-1}$ are then as inputs, yielding a representation using attention layers [11].

The generative probability distribution of y_t is given by

$$p(y_t|U, y_{1:t-1}) = \text{softmax}(W_o h_t^{dec} + b_o) \in \mathbb{R}^p \tag{4}$$

where W_o and b_o are trainable parameters.

We adopt Blender-90M [10], to initialize our Seq2Seq Transformer model, which has been pre-trained on 1.5 billion training examples from Reddit 2019.

3.2 Our Architecture

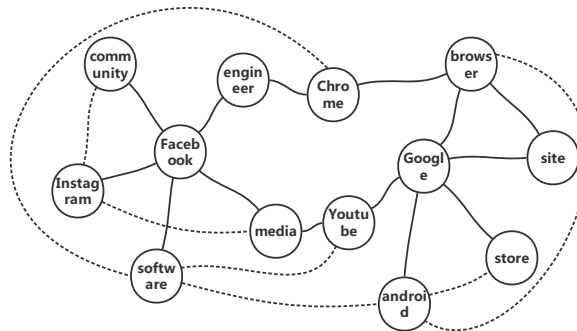


Figure 3: The 1-hop subgraph for entities "Facebook" and "Google".

Graph Knowledge Enhancement We found that constructing a graph network with words as nodes and co-occurrence as edges based on document keywords allows us to associate an entity with similar entities. This, in turn, enables us to use similar entities to represent it in downstream tasks. As shown in the FIGURE 3, illustration, the entity "Facebook" is most related with "social", "media", "picture" and so on. "Google" is most related to words like "retrieve," "email," and "advertise."

As an illustration of FIGURE 4 (a), we first convert each document in the dialogue and the retrieval knowledge in the Wizard of Wikipedia to a keywords sequence, which can extract the more important words in the sentence, thus ignoring the unimportant content. In the next step, we build a graph with keywords sequences, where each word is a node, and there will be an edge between the two nodes if these nodes appear in a document together [12].

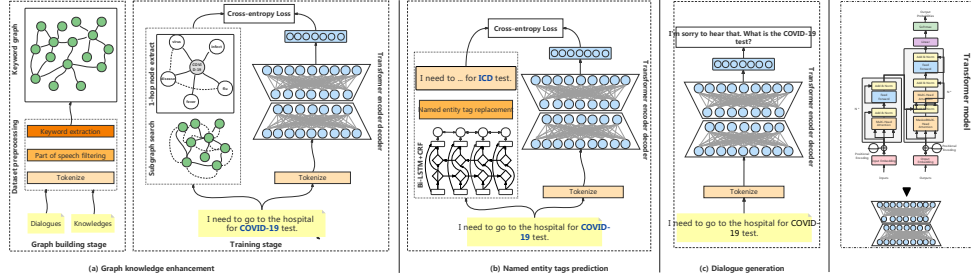


Figure 4: The architecture of GE-Blender. (a) illustrate the graph construction and model train progress of Graph Knowledge Enhancement; (b) shows how the Named Entity Tags Enhancement works. It creates a target sequence using the Bi-LSTM-based model; (c) is a standard structure of Blender, it contains encoder and decoder both based on transformer.

At the model training stage, we extract the 1-hop nodes of the proper nouns in a document from the graph and sort the 1-hop nodes according to the edge weight. Continually, we take the top K nodes as a target sequence to make the model generate it to enhance the context representation.

$$\mathcal{N}(v) = \sum_{u \in \mathcal{U}} \mathbb{1}[(v, u) \in \mathcal{E}] \quad (5)$$

The $\mathcal{N}(\ast)$ is a function for obtaining the neighborhood vertexes. The \mathcal{U} , and \mathcal{E} are vertex set and edge set on the graph respectively. And the 1-hop sequence is:

$$seq(v) = topk(argsort(\{w_{v,u} | u \in \mathcal{N}(v)\})) \quad (6)$$

where $w_{v,u}$ represents the edge weight between vertex v and u :

$$w_{v,u_i} = \frac{exp(\#(v, u_i))}{\sum_k^K exp(\#(v, u_k))} \quad (7)$$

The function $\#(\ast)$ is the statistical function of the number of co-occurrences. We use a specific start token "[GRAPH]" to mark the current task is node sequence generation. The loss of this module is as follows:

$$\mathcal{L}_{graph} = - \sum_{t=1}^{|seq(v)|} \log(p(y_t | x_t, \mathbf{h})) \quad (8)$$

where the y_t is the top-k 1-hop sequence item when the x_t is a noun or proper noun.

Besides 1-hop nodes, we also explored 2-hop and 3-hop connections. We observed that as the hop distance increases, some less relevant words get included, resulting in higher-order hops (such as 2-hop and 3-hop) representing semantically distant concepts, making them less suitable as similar words. Higher-order hops introduce more noise, adversely affecting result's accuracy. Additionally, 1-hop nodes generally suffice to meet the requirements for similar words, and the additional value of higher-order hops in terms of vocabulary is limited. From the perspective of corpus word frequency, high-frequency words tend to have more specific meanings. 1-hop words, due to their co-occurrence with the central word, also have higher frequencies, whereas multi-hop words have lower frequencies and more ambiguous meanings.

Named Entity Tags Enhancement For entities that are not in the graph, we let the model predict its named entity tags [3], thus performing knowledge enhancement on nodes that are not in the graph.

Intuitively, the correct recognition of named entity tags indicates the model to understand the context of the entity. We replace the entities in the dialogue with their named entity tags predicted by the NER model. In the second step, we take the replaced sequence as the target sequence with a start token "[NER]" and then train the model. The loss of this module is as follows:

$$\mathcal{L}_{ner} = - \sum_{t=1}^T \log(p(z_t|x_t, \mathbf{h}_{1:t-1})) \quad (9)$$

where the z_t is the named entity tag when the x_t is a named entity, otherwise, it is a token same as the origin sequence.

Part (b) of FIGURE 4, illustrates the progress of Named Entity Tags Enhancement. We make an origin dialogue sentence as the input, then we use the NER model to obtain the named entity tags, find the named entities in the original sequence and replace them with named entity tags then train the model.

Training In addition to optimizing the loss of dialog generation tasks, we also optimize the loss of Graph Knowledge Enhancement and Named Entity Tags Enhancement:

$$\mathcal{L}_{total} = \sum_{i=1}^{|O|} \mathcal{L}_{dialogue} + \sum_{i=1}^{|P|} \mathcal{L}_{graph} + \sum_{i=1}^{|Q|} \mathcal{L}_{ner} \quad (10)$$

where $|O|$, $|P|$ and $|Q|$ represent the number of training instances for the three task respectively.

Inference During inference, we only take the dialogue sentence as the input sequence and extract the 1-hop nodes of proper nouns in the sentence from the graph as the auxiliary data. We don't need any knowledge about entities. At the sequence generation stage, we adopt greedy search to select the highest probability token at each time step [1].

4. EXPERIMENTAL SETUP

We first list some research questions to prove the effectiveness of our method. Then, we describe the dataset used for responding to the questions. Finally, we present the evaluation metrics used in our study.

4.1 Research Questions

To validate the proposed approach, we define four main research questions including What's the influence of unseen entities?, Overall performance of GE-Blender, and Case study:

- **RQ-1:** What is the influence of unseen entities?
Compared to the dialogue generation on Seen Entity, the dialogue generation on Unseen Entity will be more difficult. We sample some cases to prove it.
We use the Bilingual Evaluation Understudy(BLEU) and unigram F1-score(F1) metrics to demonstrate that unseen entities affect the results of dialogue generation.
- **RQ-2:** What is the improvement of GE-Blender against the state-of-the-art?
To resolve the unseen entities problem, we try to add the information related to the unseen entities to the model to optimize the model prediction results. We will use some metrics to show the improvement brought by our model.
We can prove that our method has advantages on Unseen Entity compared to the current state-of-the-art. We compare our approach with transformer-based dialogue generation models and Blender series models.
SKT + GPT-2 [2], uses the pre-trained transformer based model SKT to select knowledge and feeds the knowledge to dialogue generation model.
SKT + PIPM + KDBTS [13], uses the posterior knowledge selection and distillation module to assist the prior knowledge selection module in knowledge selection.
KnowledGPT [2], adopts reinforcement learning to optimize the BERT based knowledge selection module and input the selected knowledge and original data into the model. It gets the most advanced performance on the Wizard.
Blender-FT Blender is a large-scale dialogue pre-training model. We fine-tune the Blender on Wizard of Wikipedia training set without utilizing external knowledge.
KG-Blender KG-Blender concatenates the context and the knowledge as input for knowledge enhancement. We fine-tune the KG-Blender model on Wizard of Wikipedia
KE-Blender KE-Blender adds definition generation and hypernym generation tasks, trying to let the model grasp the knowledge of the context of the entity so that additional knowledge is not used as input during inference. We train it with default hyperparameters.
- **RQ-3:** How does our GE-Blender perform in real cases?
We extract some cases to test the learning efficiency of the proposed approach. At the same time, the output results of other methods are compared with ours.

4.2 Parameter Settings

Our architecture is based on a Transformer encoder-decoder model, leveraging the structural framework and pre-training parameters from Blenderbot-90M. Notably, we fine-tuned several key parameters: the dropout rate was set to 0.1, while both the encoder and decoder utilized 8 layers.

Model training was performed on two Nvidia RTX 3090Ti GPUs with the AdamW optimizer. We initialized the learning rate at 1e-5 and employed a batch size of 16. Due to the use of a relatively small batch size, gradient accumulation was employed with `gradient_accumulation_steps` configured as 4. Additionally, we defined a maximum sequence length of 512 for the model, and sequences exceeding this length were subject to truncation.

4.3 Dataset & Evaluation Metric

We use the Wizard of Wikipedia [14], as our dataset. It is a chit-chat dialogue benchmark. There is some retrieved knowledge from the Wikipedia documents in each utterance for reference. The dataset contains 18,430 training dialogues and 1933 test dialogues. For the test data, it is split into Test Seen and Test Unseen based on whether the topic is appear in the training set [1]. There are 965 dialogues in Test Seen and the other 968 dialogues in Test Unseen. In addition, we also distinguish between "with knowledge during inference" and "without knowledge during inference" based on whether the retrieved knowledge is spliced at the end of the input sentence during inference.

For fair comparison with baselines in different RQ, we unitize following metrics: PPL and F1-score. We also use ROUGE as a secondary evaluation metric.

PPL [14, 15], and [1], use the perplexity of the groundtruth response (PPL) and unigram F1-score (F1) as majority metrics for dialogue generation model. Perplexity (PPL) is one of the most common metrics for evaluating language models. Given a model and an input text sequence, perplexity measures how likely the model is to generate the input text sequence [16]. Therefore, we can evaluate our model with PPL. Generally, the calculation formula of ppl is:

$$\text{PPL}(X) = \exp\left\{-\frac{1}{t} \sum_i^t \log p_{\theta}(x_i|x_{<i})\right\} \quad (11)$$

where $X = \{x_0, x_1, \dots, x_t\}$ is a input sequence of length t .

ROUGE-N [17], is Recall-Oriented Understudy for Gisting Evaluation. It includes measures to automatically determine the quality of a summary by comparing it to other reference summaries created by humans. ROUGE-N is computed as follows:

$$\text{ROUGE-N} = \frac{\sum_{S \in \{Ref_Sum\}} \sum_{gram_n \in S} \text{Count}(gram_n)}{\sum_{S \in \{Ref_Sum\}} \sum_{gram_n \in S} \text{Count}(gram_n)} \quad (12)$$

where *Ref* is short for reference and *Sum* means summary.

5. EXPERIMENTS AND RESULTS

5.1 [RQ-1]: Influence of unseen entity

Our experiments show that the unseen entity restricts the performance of the model. Blender generates the next round of dialogue only based on the content of the previous dialogue. The KG-Blender model concatenates the dialogue and retrieved knowledge as the model's input. It also doesn't work well if we have some unseen entities. If we switch the data from Test Seen to Test Unseen, these approaches will show significant performance degradation. Even though the KE-Blender uses the auxiliary task in the training state for enhancement, it still does not perform well in some cases.

For example, take the sentence A in TABLE 1, Case 1 as the input sentence, and the retrieved knowledge is also given, the KE-Blender outputs a better result. But if we change the input to sentence B of Case 1, The model indicates that it does not know the meaning of the input. In summary, the unseen entity will have an impact on the dialogue generation model. Ignorance of unseen entities leads to a completely unavailable reply.

Table 1: Demonstration of the impact of unseen entity on previous dialogue generation model.

cases	entity type	sentence
Case 1	seen	Sentence A: I have tested positive for the flu so I need to go to the hospital.
		KE outputs: I hope you get it treated soon! It is a viral infectious disease...
		GE outputs: I hope you get well soon! I'm glad you're okay!
	unseen	Sentence B: I have tested positive for the COVID-19 so I need to go to the hospital.
		KE outputs: I have never been tested for that. What is it?
		GE outputs: I hope it's not a serious problem. I hope your symptoms are not too bad.
Case 2	seen	Sentence A: Do you like the movie Harry Potter?
		KE outputs: I love Harry Potter! I love the Harry Potter series...
		GE outputs: I do! I love the movies! Im a huge fan of the Harry Potter books...
	unseen	Sentence B: Do you like the movie X-Men?
		KE outputs: I do! I love the X-Men movies. I love how they use a lot of xfactor characters.
		GE outputs: I do! I love the x X-Men movies. Im a huge fan of wolverine.

From TABLE 1, conversely, our approach gave sentence B a good response. It captures that "COVID-19" may be a medical related thing.

Table 2: Comparison of our method with other state-of-the-arts.

Model	Test Seen		Test Unseen	
	PPL	F1-score	PPL	F1-score
w/ knowledge during inference				
SKT + GPT-2[2]	17.6	20.3	23.7	17.8
SKT+PIPM+KDBTS[13]	42.7	19.9	65.7	17.6
KnowledGPT[2]	19.2	22.0	22.3	20.5
KG-Blender[1]	13.8	18.4	16.3	17.8
KE-Blender[1]	14.64	18.1	16.6	17.4
GE-Blender (Ours)	14.9	18.0	17.1	16.9
w/o knowledge during inference				
Blender-FT	17.3	15.8	20.7	15.1
KG-Blender	18.6	15.5	22.7	14.7
KE-Blender	16.8	16.4	20.6	15.1
GE-Blender (Ours)	16.5	16.7	20.0	15.3

It verifies that GE-Blender is effective for unseen entity problems. Through the proposed method, the model captures the global co-occurrence information of entities and embeds it in the token’s embedding, thereby improving the model’s understanding of the current context.

5.2 [RQ-2]: Overall Performance of GE-Blender

PPL & F1-score We conduct experiments on some benchmarks on Wizard of Wikipedia. The details of the comparison are shown in TABLE 2. For scenarios where knowledge is used during inference, KnowledGPT has the most advanced performance. Benefiting from the ability to capture knowledge, our method has better metric results in knowledge-free inference. This proves the assumption we proposed previously that the representation of the context of entities can be enhanced with the global co-occurrence information of entities.

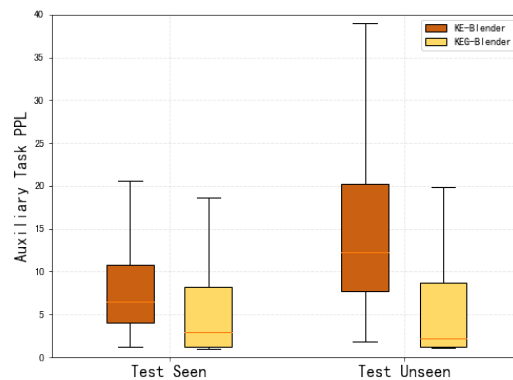


Figure 5: Boxplots of auxiliary task PPL values for KE-Blender and GE-Blender.

What is more, we evaluate the auxiliary tasks of KE-Blender and GE-Blender with the PPL, respectively. As can be seen from FIGURE 5, on Test Seen, both approaches have smaller PPL (KE is 6.7 and GE is 4.0). On the other hand, the two approaches show a large difference in PPL (KE is 12.5 and GE is 4.9) on Test Unseen. Significantly, our auxiliary task is more effective than a definition generation task on Test Unseen.

ROUGE Furthermore, we output the ROUGE metric to augment the comparison between KE-Blender and GE-Blender. As TABLE 3 shows, our method has the best precision and recall rate on Test Unseen.

Table 3: Experimental results on the ROUGE metric.

Rouge Precision				Rouge Recall			
Test Seen							
Model	1-gram	2-gram	longest	Model	1-gram	2-gram	longest
Blender-FT	13.07	1.48	11.71	Blender-FT	12.84	1.57	12.36
KE-Blender	13.53	1.62	12.25	KE-Blender	13.28	1.73	12.48
GE-Blender	13.29	1.60	11.97	GE-Blender	12.35	1.72	11.27
Test Unseen							
Model	1-gram	2-gram	longest	Model	1-gram	2-gram	longest
Blender-FT	12.39	1.44	11.24	Blender-FT	11.31	1.46	10.32
KE-Blender	12.79	1.56	11.63	KE-Blender	11.83	1.63	10.82
GE-Blender	13.13	1.58	11.92	GE-Blender	12.26	1.67	11.19

Ablation Study We studied the contribution of tasks Graph Knowledge Enhancement and Named Entity Tags Enhancement to knowledge embedding, that is, the correlation with the improvement of evaluation metrics. From the result in TABLE 4, it is obvious that Graph Knowledge Enhancement brings more performance improvements, which verifies our assumption that global co-occurrence information can improve the representation of the context of entities. In addition, the Named Entity Tags Enhancement also brings little improvement. It means that letting the model predict named entity tags will also bring little benefit to the model to a certain extent. In general, the combination of the two tasks brings the highest performance improvement.

Table 4: Ablation study result on Test Unseen.

Model	PPL	F1-score
GE-Blender	20.0	15.27
w/o NER	20.2	15.22
w/o Graph	20.6	15.13
Blender-FT	20.7	15.08

5.3 [RQ-3]: Case Study

We have selected some examples to show the results generated by the previous method and our method. TABLE 5, shows the dialogue generation effect of each model in two different scenarios, Seen and Unseen, respectively.

Take the first case of TABLE 5, as an example, the entity "*Harry Potter*" is general in the training set, so when generating dialogue, the model has learned enough knowledge about the context of "*Harry Potter*". In the fourth case of TABLE 5, the entity "*Grimm's Fairy Tales*" is infrequent, so previous methods do not adequately incorporate knowledge of this entity into contextual sentences. However, our model generated a relatively good response. It shows that our approach can effectively perform contextual representation enhancement based on 1-hop nodes of entities.

Table 5: The output of our approach and other state-of-the-arts on selected samples.

Sentence	Output
	Seen
Who wrote Harry Potter?	Blender-FT: I'm not sure, but I do know that the series has been around since the late 18th century.
	KE-blender: I don't know who the author is, but I do know that the Harry Potter series is the most popular series in the world.
	GE-Blender: I'm not sure, but I know that the first book was published in 1996. It was written by J K. Rowling.
I want to go to the Paris.	Blender-FT: I've never been to Paris, but I'd love to go to the Museum of Art.
	KE-blender: I've never been there, but I'd love to go. it's the world's largest art museum, and the largest in the world by area.
	GE-Blender: I have never been there, but I would love to go to Paris, the capital of France.
	Unseen
Who wrote Grimm's Fairy Tales?	Blender-FT: I'm not sure, but I do know that it was written by Georger. Martin.
	KE-Blender: I'm not sure, but I do know that he wrote the first volume of the series, the winds of winter
	GE-Blender: I'm not sure, but I know that the story was written by William C. Grimm.
I want to go to the Summer Palace.	Blender-FT: I've never been to the palace, but I'd love to go to the Museum of Natural History.
	KE-Blender: I love the palace, it is a palace of the Summer Palace, a palace that is built to serve as a palace for the people of the United Kingdom and Ireland.
	GE-Blender: Ours & I've never been to the Summer Palace, but I'd love to go to the Palace of Wales.

In TABLE 6, we show the training data of two modules. First of all, in Graph Knowledge Enhancement, we can see that the 1-hop node of "*fiction*" taken from the graph has a strong correlation with "*fiction*". In TABLE 6, we show the training data of two modules, including input sequence and target sequence.

6. DISCUSSION

Open-domain dialogue generation still has huge opportunities for exploration. Our model only uses 1-hop nodes for knowledge enhancement. Of course, GNN [18], models with stronger representation capabilities can also be used for knowledge enhancement, such as GAT [19], and GCN [20]. We only use the 1-hop node as a similar node. In fact, we can also use the node2vec method to train the node embedding, so as to use the embedding to calculate the node. Alternatively,

Table 6: Enhancement data in training stage.

Graph Knowledge Enhancement	
input	It blends science \$fiction\$ and paranormal/psychological/MK Ultra type stuff together, but it’s science \$fiction\$ at its core.
target	[GRAPH] genre stories fantasy author science horror novels novel star variety writers pulp doctor franchise trek elements passaic publishing plot sci [END]
input	What is your favorite thing to do with internet access? I like being able to use my computer and \$smartphone\$ to use my email and browse the world wide web.
target	[GRAPH] smartwatch browser tablet watch iphone pro ilife homepod xcode iwork suites safari macos ipad ipod multinational itunes productivity logic creativity [END]
Named Entity Tags Enhancement	
sentence	I think the most famous tennis match was between Bobby Riggs & Billie Jean King.
target	[NER] I think the most famous tennis match was between PERSON & PERSON. [END]
sentence	PepsiCo is located in NY - I live in Georgia so we live on Coke down here.
target	[NER] ORG is located in NY - I live in GPE so we live on ORG down here. [END]

we can also use PageRank [21], to get node similarity. In the Named Entity Tags Enhancement stage, we only use the fine-grained Bi-LSTM+CRF model as our NER model, and we similarly use more advanced transformer-based methods such as RoBERTa [22], to improve the model effect.

Applicability of graph knowledge enhancement in other NLP tasks. We can apply graph knowledge enhancement to other applications of NLP. For example, in weakly supervised text classification[23], we have also done a simple experiment. It has been proved by experiments that the nodes similar to the label name extracted through the graph will indeed promote the classification effect.

Difficulty of Open-domain dialogue generation. Open domain dialogue dataset in languages besides English and Chinese are difficult to find [4]. Users usually do not have any specific goals, and the content of the dialogue is more casual, which makes the model need to guess the meaning of the user’s dialogue. At the same time, due to the novelty of the conversation topic, there will be a large number of unseen entities, and some unseen entities are not associated with any entities in the historical data.

7. CONCLUSION

In this paper, we propose to use graph knowledge as a method of knowledge enhancement. We use the graph node generation task to enhance the model learning knowledge, and use the named entity tags to expand unseen entities. Experiments show that the proposed approach works well on unseen

entity problems. Our method does not require additional input of knowledge information during the inference stage, which offers significant convenience and usability for inference work.

References

- [1] Cui Leyang et al. Knowledge Enhanced Fine-Tuning for Better Handling Unseen Entities in Dialogue Generation.2021. Arxiv Preprint: <https://arxiv.org/pdf/2109.05487.pdf>
- [2] Zhao X et al. Knowledge-grounded dialogue generation with pretrained language models.2020. Arxiv Preprint: <https://arxiv.org/pdf/2010.08824.pdf>
- [3] Lample G, Ballesteros M, Subramanian S, Kawakami K, Dyer C. Neural architectures for named entity recognition. 2016. Arxiv Preprint: <https://arxiv.org/pdf/1603.01360.pdf>
- [4] Kann K et al. Open-Domain Dialogue Generation: What We Can Do, Cannot Do, and Should Do Next. In: Proceedings of the 4th Workshop on Nlp for Conversational AI. 2022:148-165.
- [5] Zhang Z, Han X, Liu Z, Jiang X, Sun M, Liu Q. Ernie: Enhanced Language Representation With Informative Entities. 2019. Arxiv Preprint: <https://arxiv.org/abs/1905.07129>
- [6] Chen J, Hu Y, Liu J, Xiao Y, Jiang H. Deep Short Text Classification With Knowledge Powered Attention. In Proceedings of the AAAI conference on Artificial Intelligence. 2019;33:6252-6259.
- [7] Nadeau D, Sekine S. A Survey of Named Entity Recognition and Classification. *Linguisticae Investigationes*. 2007;30:3-26.
- [8] Li J, Sun A, Han J, Li C. A Survey on Deep Learning for Named Entity Recognition. *IEEE Trans Knowl Data Eng*. 2020;34:50-70.
- [9] Gardner M, Grus J, Neumann M, Tafjord O, Dasigi P, et al. Allennlp: A Deep Semantic Natural Language Processing Platform. 2018. Arxiv Preprint: <https://arxiv.org/abs/1803.07640>
- [10] Roller S, Dinan E, Goyal N, Ju D, Williamson M, et al. Recipes for Building an Open-Domain Chatbot. 2020. Arxiv Preprint: <https://arxiv.org/pdf/2004.13637.pdf>
- [11] Vaswani A et al. Attention Is All You Need. *Adv Neural Inf Process Syst*. 2017;30.
- [12] Nikolentzos G, Dasoulas G, Vazirgiannis M. K-Hop Graph Neural Networks.*Neural Netw*. 2020;130:195-205.
- [13] Chen Xiuyi et al. Bridging the Gap Between Prior and Posterior Knowledge Selection for Knowledge-Grounded Dialogue Generation.In: Proceedings of the 2020 conference on empirical methods in natural language processing. 2020:3426-3437.
- [14] Dinan E et al. Wizard of wikipedia: knowledge-powered conversational agents. 2018. Arxiv Preprint: <https://arxiv.org/pdf/1811.01241.pdf>
- [15] Kim B, Ahn JH, Kim Gunhee. Sequential Latent Knowledge Selection for Knowledge-Grounded Dialogue. 2020. Arxiv Preprint: <https://arxiv.org/pdf/2002.07510.pdf>
- [16] Jelinek F, Mercer RL, Bahl LR, Baker JK. Perplexitya Measure of the Difficulty of Speech Recognition Tasks. *J Acoust Soc Am*. 1977;62.

- [17] Lin C-Y. Rouge: A Package for Automatic Evaluation of Summaries. In: Text Summarization Branches Out. 2004:74-81.
- [18] Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G. The Graph Neural Network Model. IEEE Trans Neural Netw. 2008;20:61-80.
- [19] Velikovi P, Cucurull G, Casanova A, Romero A, Liò P, et al. Graph Attention Networks. 2017. ArXiv preprint: <https://arxiv.org/pdf/1710.10903.pdf>
- [20] Kipf TN, Welling M. Semi-supervised Classification With Graph Convolutional Networks. 2016. ArXiv preprint: <https://arxiv.org/pdf/1609.02907.pdf>
- [21] <https://www.cis.upenn.edu/~mkearns/teaching/NetworkedLife/pagerank.pdf>
- [22] Liu Y, Ott M, Goyal N, Du J, Joshi M, al. Roberta: A Robustly Optimized Bert Pretraining Approach. 2019. ArXiv preprint: <https://arxiv.org/pdf/1907.11692.pdf>
- [23] Meng Y, Shen J, Zhang C, Han J. Weakly Supervised Neural Text Classification. In: Proceedings of the 27th ACM international conference on information and knowledge management. 2018:983-992.