Response Enhanced Semi-supervised Dialogue Query Generation

Jianheng Huang[1,2,3*], **Ante Wang**[1,2,3*], **Linfeng Gao**[1,3], **Linfeng Song**[4], **Jinsong Su**[1,2,3†]

[1]School of Informatics, Xiamen University, China
[2]Shanghai Artificial Intelligence Laboratory, China
[3]Key Laboratory of Digital Protection and Intelligent Processing of Intangible Cultural Heritage
of Fujian and Taiwan (Xiamen University), Ministry of Culture and Tourism, China
[4]Tencent AI Lab
{enatsu, wangante, 22920202200799}@stu.xmu.edu.cn,
lfsong@tencent.com, jssu@xmu.edu.cn

Abstract

Leveraging vast and continually updated knowledge from the Internet has been considered an important ability for a dialogue system. Therefore, the dialogue query generation task is proposed for generating search queries from dialogue histories, which will be submitted to a search engine for retrieving relevant websites on the Internet. In this regard, previous efforts were devoted to collecting conversations with annotated queries and training a query producer (QP) via standard supervised learning. However, these studies still face the challenges of data scarcity and domain adaptation. To address these issues, in this paper, we propose a semi-supervised learning framework – SemiDQG, to improve model performance with unlabeled conversations. Based on the observation that the search query is typically related to the topic of dialogue response, we train a response-augmented query producer (RA) to provide rich and effective training signals for QP. We first apply a similarity-based query selection strategy to select high-quality RA-generated pseudo queries, which are used to construct pseudo instances for training QP and RA. Then, we adopt the REINFORCE algorithm to further enhance QP, with RA-provided rewards as fine-grained training signals. Experimental results and in-depth analysis of three benchmarks show the effectiveness of our framework in cross-domain and low-resource scenarios. Particularly, SemiDQG significantly surpasses ChatGPT and competitive baselines. Our code is available at https://github.com/DeepLearnXMU/SemiDQG.

Introduction

Recent years have witnessed the burgeoning of pre-trained language models (PLMs) (Lewis et al. 2019; Raffel et al. 2020) and large language models (LLMs), which effectively improve the performance of various downstream tasks and pave the way for artificial general intelligence (AGI) (Goertzel and Pennachin 2007). Despite the variation in size, these models can still fail to generate factual content, which is known as *hallucination* (Ji et al. 2023; OpenAI 2023). To tackle this issue, researchers have explored incorporating external knowledge from search engines (Komeili, Shuster,

and Weston 2022). Typically, to bridge a model with a search engine, a query producer is used to generate search queries for retrieving relevant websites. In this work, we focus on *dialogue query generation*, which is more challenging as it has to mine user intents from complex dialogue contexts.

To train such a query producer, previous studies resort to supervised learning, where conversations with annotated search queries are used to fine-tune a pre-trained model (Lewis et al. 2019; Raffel et al. 2020). However, it is costly to construct a dataset with enough human annotations, and the trained model may still have a disappointing performance in out-of-domain conversations. A common practice to tackle these issues is semi-supervised learning (Yarowsky 1995; Blum and Mitchell 1998), which has been widely investigated in both CV (Rosenberg, Hebert, and Schneiderman 2005) and NLP (Zhang and Zong 2016; He et al. 2020) fields. It suits the dialogue query generation task well because abundant conversations without annotated queries are easy to obtain. As implemented in self-training, we expect the model to generate pseudo queries for unlabeled conversations. While in practice, some pseudo queries are often unsatisfying, which may lead to error accumulation and model performance degradation. It can be said that the challenge of effectively collecting high-quality pseudo queries to construct pseudo instances continues to be a hurdle in this task.

Fortunately, we notice that a search query can be highly relevant to the topic of its corresponding dialogue response. When augmenting the input with response information, the model can often generate better search queries. As illustrated in the first case of Table 1, the standard query producer (QP) solely incorporates the dialogue history as input and mistakenly recognizes "*north atlantic*" as the query. In contrast, the response-augmented query producer (RA) accurately predicts the correct query by inferring the mainly discussed topic "*ireland*" (referred to by "*it*") in the response. This demonstrates the potential of RA to generate high-quality pseudo queries which can subsequently be used to construct pseudo instances for training QP.[1] However, we notice that RA may also generate some low-quality queries especially when it is overly influenced by the response. In the second

---

[1]Note that we focus on improving the performance of QP in that the response information is inaccessible in practical application.

Example 1

History	System:	Ever been to **Ireland** in the <u>North Atlantic</u>? Heard it is lovely.
	User:	I have not been there but I'd love to
Response	System:	**It**'s not too big but **it** is the third largest island in Europe so not too small, like a lively and nice place.
Gold query		*ireland*
QP's prediction		*north atlantic* ✗
RA's prediction		*ireland* ✔

Example 2

History	User:	I love to go **bowling** with my family, but I'm a horrible bowler. Do you like it?
	System:	Oh, yes, I love **bowling**. Rolling balls down the lane and knocking down the pins gives me a charge.
	User:	I know! I love it when I just knock one down - lol!! My kids want to win, I just like playing.
Response	System:	Since **it** is one of the major throwing sports, **it** is a lot like the javelin throw.
Gold query		*bowling*
QP's prediction		*bowling* ✔
RA's prediction		*javelin throw* ✗

Table 1: Two examples from Wizard-of-Wikipedia (WoW, Dinan et al. 2018) with corresponding dialogue responses, gold queries, and model predictions. QP and RA denote the standard query producer and the response-augmented model respectively. Here we highlight the main topics or their referring expressions that help predict gold queries in bold and mark the misleading concepts with underlines.

case of Table 1, RA ignores the principal topic "*bowling*" in the history, but mistakenly takes "*javelin throw*", another topic in the response, as the prediction. Therefore, it is worth exploring ways to select high-quality RA-generated pseudo queries.

Based on the observations above, we propose a novel framework **Semi**-supervised **D**ialogue **Q**uery **G**eneration (SemiDQG) which effectively improves QP with the guidance of RA. Specifically, we first train QP and RA on a labeled dataset. Subsequently, we leverage the capabilities of RA to generate pseudo queries for an unlabeled dataset and introduce a query selection strategy based on the prediction similarity between QP and RA to select high-quality RA-generated queries (e.g., "*ireland*" in Table 1). In a semi-supervised manner, these selected queries are used to construct pseudo instances, thereby enhancing the performance of both models. Finally, to further enhance QP, we adopt the REINFORCE algorithm (Williams 1992) with RA-provided rewards, serving as fine-grained training signals, based on QP-generated candidate queries. Both pseudo instance construction and the reinforcement learning approach proposed above can jointly consider the output features from both QP and RA. Thus, it can fully utilize the training signals from

RA spanning different levels of granularity and effectively alleviate the negative effect stemming from input discrepancy between the two models.

We conduct experiments in cross-domain and low-resource scenarios respectively. In the cross-domain scenario, we construct Wizard-of-Internet (WoI, Komeili, Shuster, and Weston 2022) → Wizard-of-Wikipedia (WoW, Dinan et al. 2018) in English, and DuSinc (Zhou et al. 2022) → KdConv (Zhou et al. 2020) in Chinese. In the low-resource scenario, we focus on WoI as it provides more data for better evaluation. Experiment results show that SemiDQG significantly outperforms ChatGPT and various baselines. Moreover, in-depth analysis validates the effectiveness of the proposed query selection strategy and reinforcement learning method in our framework.

Related Work

Search Query Generation Using a search engine to exploit knowledge from the Internet is gaining popularity for benefiting various knowledge-intensive tasks, such as open-domain QA (Qi et al. 2019; Nakano et al. 2022), and dialogue response generation (Komeili, Shuster, and Weston 2022; Glaese et al. 2022). Early attempts simply take user questions or keywords as search queries but have been proven to be ineffective when handling distinct domains (Xie et al. 2023) or complex dialogue contexts (Wang et al. 2023a). Recent work (Komeili, Shuster, and Weston 2022; Zhou et al. 2022; Wang et al. 2023a) usually trains a query producer to extract or generate search queries, with query generation more popular due to the limitation of extraction. With the release of various query generation datasets (Komeili, Shuster, and Weston 2022; Zhou et al. 2022), researchers can build their query producers in supervised learning manners. As query annotations are costly to collect, some researchers (Qi et al. 2019; Wang et al. 2023a,b) introduce additional supervision signals to train their query producers.

Very recently, many LLM products (Thoppilan et al. 2022; Glaese et al. 2022) use prompting techniques to generate search queries instead of adopting an independent query producer. However, prompting techniques heavily rely on the ability of LLMs to understand the prompt. After a comparison of these two strategies, our experimental results show that even ChatGPT still shows inferior performance than a smaller task-specific model.

Semi-supervised Learning As a branch of machine learning, semi-supervised learning exploits the knowledge from unlabeled data when labeled data is limited. In this regard, typical methods mainly include self-training (Yarowsky 1995), co-training (Blum and Mitchell 1998; Zhou and Goldman 2004), tri-training (Zhou and Li 2005), and so on. Among them, self-training is one of the earliest approaches and continues to gain popularity in recent years (Amini et al. 2022). For a specific task, it improves a model by iteratively enriching the training data with selected pseudo instances. In NLP fields, several studies have investigated self-training on text generation tasks, such as neural machine translation (He et al. 2020), text summarization (He et al. 2020), and ques-
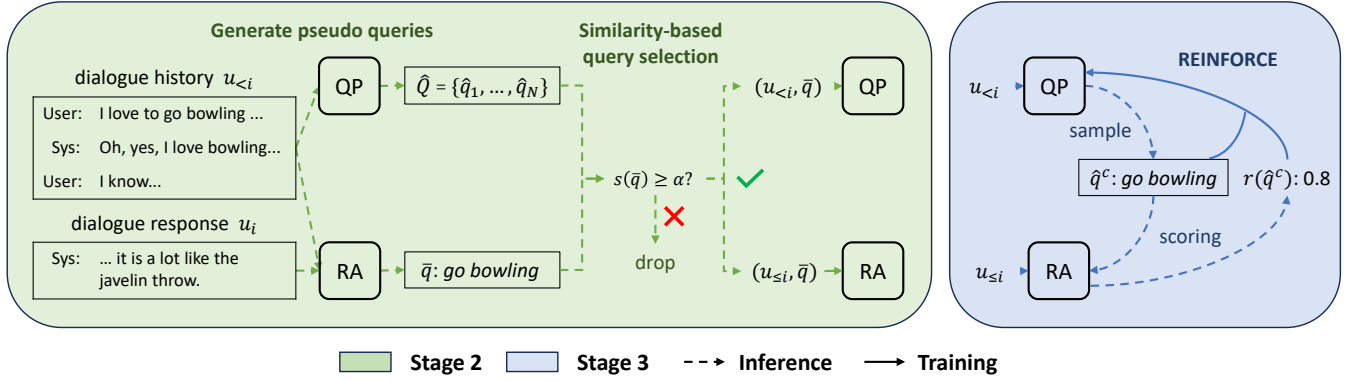
Figure 1: Our proposed Semi-supervised Dialogue Query Generation (SemiDQG) framework. In Stage 1, we train QP and RA via standard supervised training on labeled data (not shown for clarity). In Stage 2, for each unlabeled conversation, we use RA to generate its pseudo queries $\bar{q}$. We only keep the query whose similarity score $s(\bar{q})$ exceeds a given threshold α to construct a pseudo instance. We use these high-quality pseudo instances to train QP and RA. In Stage 3, QP is further enhanced using RA-guided reinforcement learning.

tion generation (Kulshreshtha et al. 2021). Nevertheless, it is challenging to collect appropriate pseudo instances, potentially hindering the progress in building more powerful models. In this work, we focus on leveraging semi-supervised learning to further enhance the query producer, as illustrated in the following section.

Our Framework

Figure 1 illustrates the procedure of our proposed SemiDQG, which can be roughly separated into three stages. In **Stage 1**, we train a standard query producer (QP) and a response-augmented query producer (RA) on a labeled dataset via supervised learning. In **Stage 2**, both QP and RA generate pseudo queries for an unlabeled dialogue corpus. Then, based on the prediction similarity between RA and QP, we select high-quality RA-generated queries to construct pseudo instances for training these two models. Nevertheless, due to the discrepancy between QP and RA, these pseudo instances might not effectively guide QP. Thus, in **Stage 3**, we employ reinforcement learning to further improve QP with RA providing rewards as fine-grained training signals. Detailed descriptions will be provided in the following subsections.

Stage 1: Train QP and RA with Supervised Learning

As described above, under our framework, we train a QP and an RA via supervised learning in this stage. Formally, given the dialogue history $u_{<i} = u_1, ..., u_{i-1}$, both QP and RA aim to predict the target query q. The difference between QP and RA lies in that RA takes the dialogue response u_i as additional input, which is inaccessible in practical application.

Following the previous study (Komeili, Shuster, and Weston 2022), we choose pre-trained T5 (Raffel et al. 2020) as the basic model for QP and RA, and fine-tune them on conversations with annotated queries. For each instance, we

take the cross-entropy loss (CE) as the training objective:

$$\begin{aligned}\mathcal{L}_{qp} &= -\log p(q \mid u_{<i}; \theta_{\mathrm{qp}}), \\ \mathcal{L}_{ra} &= -\log p(q \mid u_{\leq i}; \theta_{\mathrm{ra}}),\end{aligned} \tag{1}$$

where θ_{qp} and θ_{ra} denote the parameters of QP and RA respectively.

Stage 2: Semi-supervised Learning with Similarity-based Query Selection

Once the above training is completed, we use RA to generate queries for an unlabeled dialogue corpus and select high-quality queries to construct pseudo instances, which are finally used to enhance QP and RA. Please note that, unlike the standard self-training, we take advantage of RA rather than QP in generating pseudo queries and constructing instances for QP.

One important step of the above process is the quality evaluation of RA-generated queries. Intuitively, the most direct approach is to use their predictive probabilities as the evaluation metric. However, modern neural networks are often poorly calibrated (Guo et al. 2017) and their predictive probabilities may not be reliable. To deal with this issue, we also use QP to generate queries for the unlabeled dialogue corpus and then evaluate the quality of RA-generated queries by the prediction similarity between RA and QP.

Formally, given a dialogue history $u_{<i}$ and response u_i in the unlabeled corpus, we use RA to generate a query $\bar{q}$ and adopt QP to generate N queries $\hat{Q} = \{\hat{q}_1, ..., \hat{q}_N\}$ with only $u_{<i}$ as input.

Then we quantify the quality of RA-generated query $\bar{q}$ by the following similarity score:

$$s(\bar{q}) = \max\{\mathcal{F}_{sim}(\bar{q}, \hat{q}_i)\}_{\hat{q}_i \in \hat{Q}}, \tag{2}$$

where $\mathcal{F}_{sim}(*)$ denotes a text similarity function that returns the score of a specific quantitative metric (e.g., Unigram F1 and ROUGE) or a semantic similarity model such as Sentence-BERT (Reimers and Gurevych 2019). Note that

if $\bar{q}$ is overly influenced by the response information, it will contain unrelated concepts from the response and thus will have a low similarity score.

Afterward, we select high-quality RA-generated queries, whose similarity score exceeds a pre-determined threshold α, to construct pseudo instances with the corresponding dialogue histories. Next, we use these pseudo instances to further train QP using the CE loss again (See Equation 1).

Particularly, during this process, the training strategies we adopt vary slightly in different scenarios. Concretely, in the cross-domain scenario, we directly fine-tune the best checkpoints of QP from Stage 1 on RA-labeled pseudo instances, as implemented in (Meng et al. 2023). While in the low-resource scenario, we follow He et al. (2020) to retrain QP on RA-labeled pseudo instances. Finally, it should be noted that we also further train RA in the above manners to facilitate the subsequent training.

Stage 3: RA-guided Reinforcement Learning

Unfortunately, during the stage above, there are still some low-quality pseudo instances left, which may have negative effects. More importantly, QP still fails to fully utilize useful fine-grained training signals from RA by training on pseudo instances only. Thus, in Stage 3, we adopt the REINFORCE algorithm (Williams 1992) to tackle these problems.

Concretely, for each instance in an unlabeled dialogue corpus, we first sample N_c candidate queries from the predictive distribution of QP. Here we follow Liu et al. (2022) to calculate the length-normalized log probability of QP for each candidate query $\hat{q}^c$:

$$f_{qp}(\hat{q}^c) = \frac{\sum_j \log p(\hat{q}^c_j \mid u_{<i}, \hat{q}^c_{<j}; \theta_{\mathrm{qp}})}{|\hat{q}^c|}, \qquad (3)$$

where $\hat{q}^c_j$ denotes the j-th query token. Furthermore, using a *softmax* normalization, we derive a predictive distribution over all candidate queries, acting as the stochastic policy to sample $\hat{q}^c$.

Then we explore the following two kinds of reward $r(\hat{q}^c)$:

- **Prob-based Reward** Similar to Equation 3, we feed each candidate query $\hat{q}^c$ into RA and calculate its length-normalized log probability, denoted as $f_{ra}(\hat{q}^c)$. We directly use this probability as the reward:

$$r(\hat{q}^c) = f_{ra}(\hat{q}^c), \qquad (4)$$

- **Rank-based Reward** We sort all candidate queries by $f_{ra}(\hat{q}^c)$ and design the following reward:

$$r(\hat{q}^c) = \frac{1}{1 + g(\hat{q}^c)}, \qquad (5)$$

where $g(*)$ is a function that returns the descending order of input queries according to $f_{ra}(\hat{q}^c)$. Note that We perform normalization across all N_c candidate queries to reduce variance in gradient estimation. This procedure will punish pseudo queries with low rankings.

Finally, QP can be trained with the guidance of reward:

$$\mathcal{L}_{rl} = -r(\hat{q}^c) \log p(\hat{q}^c \mid u_{<i}; \theta_{\mathrm{qp}}). \qquad (6)$$

Intuitively, the reward provided by RA is a fine-grained training signal compared to the pseudo queries in Stage 2.

Experiments

Setup

Datasets We conduct experiments in both cross-domain and low-resource scenarios across three benchmarks. In the cross-domain scenario, we explore Wizard-of-Internet (WoI, Komeili, Shuster, and Weston 2022) → Wizard-of-Wikipedia (WoW, Dinan et al. 2018) in English, and DuSinc (Zhou et al. 2022) → KdConv (Zhou et al. 2020) in Chinese. In the low-resource scenario, we focus on WoI, as it provides more high-quality query annotation data for better evaluation.

- **Wizard-of-Internet (WoI)** A comprehensive dataset providing conversations with search query annotations and websites retrieved from the Bing Search API[2].

- **Wizard-of-Wikipedia (WoW)** A popular dialogue dataset, with each utterance grounded on a Wikipedia page. We follow Wang et al. (2023a) to use Wikipedia Search[3] as the search engine and evaluate the quality of search queries by comparing retrieved Wikipedia page titles with the gold one.

- **DuSinc** A Chinese open-domain dialogue dataset with annotated search queries. We use its publicly available part[4] for experiments.

- **KdConv** A Chinese multi-domain knowledge-driven conversation dataset containing knowledge graph (KG) triplets where dialogue responses may need knowledge from a KG. For each triplet, we use the concatenation of the subject and the predicate as the gold query.

Evaluation Metrics All the metrics we use to evaluate the model performance are listed below:

- **Recall-k (R@k)** We use this metric only on WoW. It is decided by the recall of the target Wikipedia page title when feeding the top-k ($k \in \{1, 3\}$) predicted queries to Wikipedia search.

- **Unigram F1 (Uni. F1)** We use this metric on all the datasets. It measures the unigram overlap between the prediction and gold reference.

- **BLEU** It is a typical metric for text generation tasks that mainly focus on the precision of n-gram for the prediction against the gold reference. We use *sacrebleu* (Post 2018) for *BLEU-1/2* calculation.

- **ROUGE** As another commonly used evaluation metric for text generation, it accounts for both precision and recall, thus providing more comprehensive scores. We report *ROUGE-1/2/L* using Google's implementation[5].

Baselines We compare our proposed SemiDQG with the following baselines:

---

[2]https://www.microsoft.com/en-us/bing/apis/bing-web-search-api

[3]https://www.wikipedia.org

[4]https://aistudio.baidu.com/aistudio/datasetdetail/139431

[5]https://github.com/google-research/google-research/tree/master/rouge

Model	WoW	KdConv
QP(Stage 1)	40.68	56.85
Stage 2		
w/ RA-labeled instances	27.31	63.50
+ Uni. F1 as $\mathcal{F}_{sim}$	42.92	64.68
+ Sentence-BERT as $\mathcal{F}_{sim}$	41.34	64.11

Table 2: Results on development sets of WoW and KdConv with different $\mathcal{F}_{sim}$ in Stage 2.

- **T5-base** A fine-tuned T5-base model (Raffel et al. 2020) on a labeled dataset following Komeili, Shuster, and Weston (2022), same as QP in Stage 1 as mentioned above.

- **Self-training(scratch)** A model initialized from the original T5-base parameters and trained on the QP-labeled pseudo instances following He et al. (2020).

- **Self-training(QP)** A model initialized from trained QP in Stage 1 and then tuned on *self*-labeled pseudo instances.

- **Self-training(joint)** The original T5-base model fine-tuned on the combination of synthetic data and authentic data following He et al. (2020).

- **QP-ext/QP-gen** (Wang et al. 2023a) Different types of QPs, based on extraction and generation respectively. Both are trained with cheap noisy supervision, taking feedback from the Wikipedia search as training signals, and significantly surpass unsupervised keyword extraction methods.

- **KD (RA $\rightarrow$ QP)** A model that adopts vanilla knowledge distillation (Hinton, Vinyals, and Dean 2015; Miao et al. 2023), where the student model (QP) is trained to fit predictions of the teacher model (RA).

- **ChatGPT** We utilize the official `gpt-3.5-turbo` API[6] to perform inference by in-context learning with 3 or 8 demonstrations following Ye et al. (2023).

Implementation Details For all pre-trained models used in this work, We utilize the checkpoints from Huggingface[7], with different T5-base variants according to languages. For English datasets, we use the `t5-base`. While for Chinese datasets, we employ the `Langboat/mengzi-t5-base` (Zhang et al. 2021). During training, we apply an Adam optimizer, with a linear scheduler and an initial learning rate of 3e-5. We use a batch size of 64 for cross-domain experiments and 16 for low-resource counterparts. For the main experiments, we set $N = 1$ for query selection, and use Unigram F1 as the default $\mathcal{F}_{sim}$. The selection of hyperparameter α for WoW/WoI/KdConv is 1.0/1.0/0.5, respectively. We set $N_c = 10$ for rank-based reward in the cross-domain scenario and $N_c = 3$ for other settings. Next, we analyze the selection of some key hyperparameters as follows.
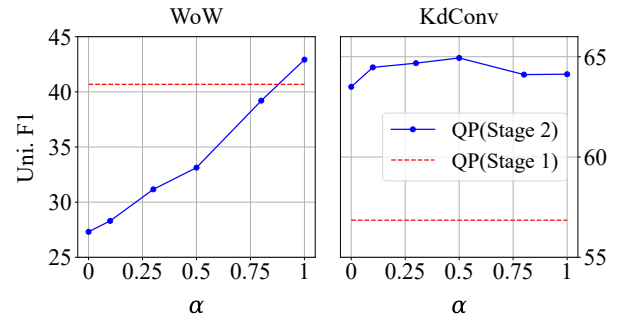
---

[6]https://openai.com/blog/openai-api
[7]https://huggingface.co/models

Figure 2: Effect of α on Unigram F1 for development sets of WoW and KdConv in Stage 2.
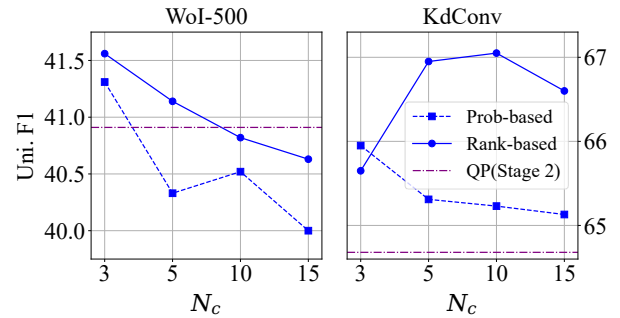

Figure 3: Results on development sets of WoI and KdConv, with different N_c for probability-based and rank-based rewards.

Development Results

Selection of $\mathcal{F}_{sim}$ In Stage 2, we investigate two types of $\mathcal{F}_{sim}$: Unigram F1 as the quantitative metric and Sentence-BERT as the semantic similarity model. As observed in Table 2, both $\mathcal{F}_{sim}$ can effectively enhance QP, and the semantic similarity model does not necessarily yield better results than conventional quantitative metrics. Thus, we take Uni. F1 as the $\mathcal{F}_{sim}$ for later experiments.

Selection of α We explore $\alpha = 0, 0.1, 0.3, 0.5, 0.8, 1.0$ on WoW and KdConv. As illustrated in Figure 2, the selection of the threshold α significantly affects the model performance. The model reaches the best performance when $\alpha = 1.0/0.5$, demonstrating the effectiveness of our similarity-based query selection. Especially, the model performs even worse than QP(Stage 1) when taking a small α on WoW. This may be attributed to the larger domain gap existing between WoI and WoW. Nevertheless, this still emphasizes the necessity of adopting our query selection strategy. For DuSinc $\rightarrow$ KdConv, the gap may be closer, thus setting a relatively lower α can provide a more diverse set of high-quality pseudo instances to boost the model performance.

Selection of N_c and Reward Types We explore the selection of reward types (prob-based and rank-based) with $N_c = 3, 5, 10, 15$ for two scenarios separately, as shown in Figure 3. Generally, prob-based reward only works better when N_c is small, and is inferior to rank-based reward, especially

Model	WoW			KdConv		
	Uni. F1	R@1	R@3	Uni. F1	BLEU-1/2	ROUGE-1/2/L
T5-base	48.11	54.30	68.59	57.13	50.86 / 48.37	56.91 / 50.74 / 56.72
QP-ext†	-	62.41	72.91	-	-	-
QP-gen†	-	56.77	66.08	-	-	-
ChatGPT(3-shot)	37.88	42.63	-	49.21	39.82 / 36.11	49.11 / 41.49 / 48.90
ChatGPT(8-shot)	41.38	46.73	-	49.69	38.00 / 34.11	49.46 / 41.24 / 49.20
Self-training(scratch)	37.94	33.35	45.30	58.81	53.90 / 51.02	58.26 / 52.10 / 58.08
Self-training(QP)	37.83	35.46	46.91	58.43	53.77 / 50.84	58.10 / 51.76 / 57.91
Self-training(joint)	38.61	33.80	47.16	60.10	53.71 / 51.11	59.79 / 53.78 / 59.64
KD (RA → QP)	38.42	40.35	48.31	65.60	61.88 / 58.88	65.40 / 58.95 / 65.32
SemiDQG	**57.12**	**63.50**	**74.89**	**67.21**	**62.89 / 60.19**	**67.33 / 60.62 / 67.30**

Table 3: Test results on WoW and KdConv in the cross-domain scenario. † denotes the results reported in (Wang et al. 2023a). Note that we only request ChatGPT to generate the most relevant query for each instance, so its R@3 is not applicable.

in the cross-domain scenario. This is because poorly calibrated RA cannot provide reasonable confidence scores due to domain discrepancy. Furthermore, larger N_c leads to performance degradation in both scenarios since a large N_c will introduce more diverse but low-quality candidates.

Main Results

Cross-domain Scenario   Table 3 shows the main results in the cross-domain scenario. Overall, SemiDQG achieved the best result, exhibiting remarkable superiority over all baselines across all metrics. While exceeding the typical self-training, it also surpasses other competitive baselines, even the famous LLM product ChatGPT. After an in-depth analysis, we have the following conclusions:

(1) Currently accepted LLMs still fail to handle the dialogue query generation task well, despite the application of in-context learning. As the number of demonstrations increases from 3 to 8, ChatGPT exhibits some performance improvement on WoW, yet it still falls short of expectations compared to a task-specific model. We believe that the capabilities of LLMs should be further explored, as the performance of in-context learning may be constrained.

(2) The two competitive baselines, QP-ext and QP-gen, exhibit performance closest to SemiDQG on WoW. However, their training costs are higher due to the use of search engines as feedback. Besides, both QP-ext and QP-gen are trained to predict continuous entity spans from inputs. This also makes their approaches impractical on distinct datasets.

(3) Traditional self-training may hurt model performance. As shown in Table 3, none of the three self-training variants improve the performance of QP on WoW, and even lead to a decline. Meanwhile, the performance improvement on KdConv is also limited. These results reflect the negative impact of low-quality pseudo instances. We also observe that Self-training(scratch) slightly outperforms Self-training(QP) due to different model initializations, aligning with He et al. (2020)'s findings.

(4) With the guidance of RA, KD (RA → QP) beats all self-training approaches on KdConv, demonstrating the necessity of leveraging response information. However, it also
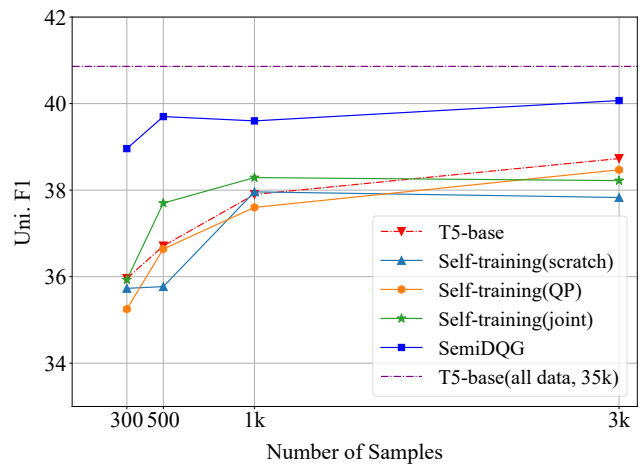

Figure 4: Unigram F1 test results on WoI in the low-resource scenario.

performs worse on WoW compared with T5-base similar to self-training baselines. Our SemiDQG successfully improves results on both datasets and significantly outperforms KD (RA → QP), validating its effectiveness.

Low-resource Scenario   Figure 4 depicts that SemiDQG also demonstrates its effectiveness in the low-resource scenario on WoI, which achieves greater performance improvement under extremely low-resource settings (300/500-shot). Besides, when using 300 labeled instances, SemiDQG outperforms a T5-base trained with 3k instances, which is 10 times data efficiency. In addition, similar to the cross-domain results, the performance of the three traditional self-training variants is suboptimal in the low-resource scenario on WoI. This also highlights the limitations of traditional methods and the effectiveness of SemiDQG.

Analysis

In this subsection, we take DuSinc → KdConv as an example to conduct a detailed analysis of our proposed framework.

Model	Uni. F1	BLEU-1/2	ROUGE-1/2/L
QP(Stage 1)	57.13	50.86 / 48.37	56.91 / 50.74 / 56.72
Stage 2			
w/ QP-labeled inst.	58.43	53.77 / 50.84	58.10 / 51.76 / 57.91
+ QP-prob sel.	59.10	53.15 / 50.75	58.92 / 52.86 / 58.78
w/ RA-labeled inst.	65.60	**61.88 / 58.88**	65.40 / 58.95 / 65.32
+ RA-prob sel.	65.73	60.06 / 57.69	65.71 / 59.60 / 65.62
+ QP-prob sel.	65.01	58.81 / 56.55	65.04 / 58.91 / 64.97
+ sim. sel.	**66.06**	61.50 / 58.77	**66.12 / 59.67 / 66.07**
Stage 3			
w/ prob-based RL	66.39	62.19 / 59.42	66.37 / 59.78 / 66.35
w/ rank-based RL	**67.21**	**62.89 / 60.19**	**67.33 / 60.62 / 67.30**
QP(Stage 1) w/ RL	64.39	59.35 / 56.45	64.14 / 57.39 / 64.08

Table 4: Ablation studies of QP on the KdConv test set. Here "instances", "similarity" and "selection" are abbreviated as "inst.", "sim." and "sel.", respectively.

Model	Uni. F1	BLEU-1/2	ROUGE-1/2/L
RA(Stage 1)	64.64	55.53 / 53.17	64.56 / 58.02 / 64.49
Stage 2			
w/ RA-labeled inst.	68.64	62.73 / 59.99	68.51 / 62.11 / 68.47
+ RA-prob sel.	68.37	62.81 / 59.98	68.33 / 61.72 / 68.26
+ QP-prob sel.	**69.11**	62.68 / 60.16	**69.16 / 62.74 / 69.07**
+ sim. sel.	68.67	**63.77 / 60.74**	68.67 / 61.84 / 68.65

Table 5: Test results of RA variants on KdConv.

Similarity-based Query Selection (Stage 2) We conduct ablation studies as shown in Tables 4 and 5, comparing our method with query selection based on predictive probabilities of either QP or RA. The main conclusions are as follows:

(1) RA-labeled instances benefit QP more. The utilization of QP-labeled pseudo instances can only slightly enhance QP on KdConv, and the improvement of adopting query selection based on its predictive probability is also limited.

(2) The quality of RA-labeled pseudo instances significantly affects the performance of QP. Similarity-based query selection works the best among these selection strategies on KdConv, despite a slight decrease in BLEU-1/2 compared to the vanilla knowledge distillation setting. Besides, both QP and RA have difficulty recognizing better pseudo queries, making probability-based query selection less effective than that of the similarity-based counterpart.

(3) RA can also benefit from QP. As depicted in Table 5, it is challenging for RA to identify instances that can result in significant self-improvement, highlighting its limitation in self-calibration. Nevertheless, with the guidance of QP, in terms of either predictive probability or prediction similarity, RA can be further enhanced.

RA as the Reward Model (Stage 3) Table 4 indicates that RA can effectively guide QP to improve model performance, regardless of whether it is adopted directly after Stage 1 or adopted after Stage 2. As the reward model, RA can provide fine-grained training signals during QP's reinforcement learning process, further tapping into the potential of RA. This validates the necessity and effectiveness of Stage 3.

We conduct further analysis to demonstrate that RA can

Model	Pearson	Uni. F1 (top-1)
QP ranking	0.3660	66.17
RA ranking	0.4109	67.46
Gold ranking	1.0000	82.92

Table 6: The effect of different ranking methods for Pearson correlation coefficient and top-1 candidate query performance on the KdConv training set in Stage 3.

provide more reasonable rewards for QP training, which intuitively decides the performance of QP after Stage 3. As RA is asked to assess each query $\hat{q}^c$ from the N_c candidates at this stage, we check whether RA can provide a better ranking to these candidate queries according to their quality.

In detail, the following rankings are compared in Table 6: (1) **QP ranking**. As previously mentioned, we utilize QP to sample the N_c candidate queries via beam search, which naturally results in a descending ranking based on its predictive probability. (2) **RA ranking**. We obtain the ranking by sorting the length-normalized log probability of RA $f_{ra}(\hat{q}^c)$ (See Equation 3) in descending order for each candidate query $\hat{q}^c$. (3) **Gold ranking**. We compute the Unigram F1 scores between each $\hat{q}^c$ and the gold query q, obtaining an oracle ranking by sorting the scores.

To evaluate the quality of each ranking, we calculate **Pearson correlation** coefficients between the QP/RA ranking and the gold ranking and **Uni. F1 (top-1)**, which gives the Unigram F1 score between the candidate query ranked highest and the gold reference q.

As shown in Table 6, the RA ranking has a stronger correlation with the gold ranking and gives higher Uni. F1 (top-1) score. This demonstrates the effectiveness of the RA ranking, as it succeeds in allowing high-quality candidate queries to be ranked higher, thus providing more reasonable rewards when applying reinforcement learning. However, we also notice that there is still a significant performance gap between the RA ranking and the gold ranking. We believe that the potential of RA can be further explored.

Conclusion

In this paper, we propose a semi-supervised learning framework, SemiDQG, to enhance the query producer (QP) with the guidance of the response-augmented query producer (RA). Taking the dialogue response as an additional feature, RA can provide better training signals for QP training. However, we notice that the input discrepancy between QP and RA will stop our model from further improving. To alleviate the negative impact of this discrepancy, we jointly consider the output features from both QP and RA as training signals for QP training. Specifically, we first apply similarity-based query selection to select high-quality RA-generated pseudo queries for training these models and then adopt RA-guided reinforcement learning to exploit fine-grained knowledge from RA to further improve QP. Experimental results and in-depth analysis in cross-domain and low-resource scenarios demonstrate the effectiveness of our SemiDQG.

Acknowledgements

References

Amini, M.-R.; Feofanov, V.; Pauletto, L.; Devijver, E.; and Maximov, Y. 2022. Self-training: A survey. *arXiv preprint arXiv:2202.12040*.

Blum, A.; and Mitchell, T. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, 92–100.

Dinan, E.; Roller, S.; Shuster, K.; Fan, A.; Auli, M.; and Weston, J. 2018. Wizard of wikipedia: Knowledge-powered conversational agents. *arXiv preprint arXiv:1811.01241*.

Glaese, A.; McAleese, N.; Trebacz, M.; Aslanides, J.; Firoiu, V.; Ewalds, T.; Rauh, M.; Weidinger, L.; Chadwick, M.; Thacker, P.; Campbell-Gillingham, L.; Uesato, J.; Huang, P.-S.; Comanescu, R.; Yang, F.; See, A.; Dathathri, S.; Greig, R.; Chen, C.; Fritz, D.; Elias, J. S.; Green, R.; Mokrá, S.; Fernando, N.; Wu, B.; Foley, R.; Young, S.; Gabriel, I.; Isaac, W.; Mellor, J.; Hassabis, D.; Kavukcuoglu, K.; Hendricks, L. A.; and Irving, G. 2022. Improving alignment of dialogue agents via targeted human judgements. arXiv:2209.14375.

Goertzel, B.; and Pennachin, C. 2007. *Artificial general intelligence*, volume 2. Springer.

Guo, C.; Pleiss, G.; Sun, Y.; and Weinberger, K. Q. 2017. On Calibration of Modern Neural Networks. arXiv:1706.04599.

He, J.; Gu, J.; Shen, J.; and Ranzato, M. 2020. Revisiting Self-Training for Neural Sequence Generation. arXiv:1909.13788.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. arXiv:1503.02531.

Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D.; Xu, Y.; Ishii, E.; Bang, Y. J.; Madotto, A.; and Fung, P. 2023. Survey of Hallucination in Natural Language Generation. *ACM Computing Surveys*, 55(12): 1–38.

Komeili, M.; Shuster, K.; and Weston, J. 2022. Internet-Augmented Dialogue Generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 8460–8478. Dublin, Ireland: Association for Computational Linguistics.

Kulshreshtha, D.; Belfer, R.; Serban, I. V.; and Reddy, S. 2021. Back-Training excels Self-Training at Unsupervised Domain Adaptation of Question Generation and Passage Retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 7064–7078. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Liu, Y.; Liu, P.; Radev, D.; and Neubig, G. 2022. BRIO: Bringing Order to Abstractive Summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2890–2903. Dublin, Ireland: Association for Computational Linguistics.

Meng, R.; Wang, T.; Yuan, X.; Zhou, Y.; and He, D. 2023. General-to-Specific Transfer Labeling for Domain Adaptable Keyphrase Generation. In *Findings of the Association for Computational Linguistics: ACL 2023*, 1602–1618. Toronto, Canada: Association for Computational Linguistics.

Miao, Z.; Zhang, W.; Su, J.; Li, X.; Luan, J.; Chen, Y.; Wang, B.; and Zhang, M. 2023. Exploring All-In-One Knowledge Distillation Framework for Neural Machine Translation. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2929–2940. Singapore: Association for Computational Linguistics.

Nakano, R.; Hilton, J.; Balaji, S.; Wu, J.; Ouyang, L.; Kim, C.; Hesse, C.; Jain, S.; Kosaraju, V.; Saunders, W.; Jiang, X.; Cobbe, K.; Eloundou, T.; Krueger, G.; Button, K.; Knight, M.; Chess, B.; and Schulman, J. 2022. WebGPT: Browser-assisted question-answering with human feedback. arXiv:2112.09332.

OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774.

Post, M. 2018. A Call for Clarity in Reporting BLEU Scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, 186–191. Brussels, Belgium: Association for Computational Linguistics.

Qi, P.; Lin, X.; Mehr, L.; Wang, Z.; and Manning, C. D. 2019. Answering Complex Open-domain Questions Through Iterative Query Generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2590–2602. Hong Kong, China: Association for Computational Linguistics.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1): 5485–5551.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. arXiv:1908.10084.

Rosenberg, C.; Hebert, M.; and Schneiderman, H. 2005. Semi-supervised self-training of object detection models.

Thoppilan, R.; Freitas, D. D.; Hall, J.; Shazeer, N.; Kulshreshtha, A.; Cheng, H.-T.; Jin, A.; Bos, T.; Baker, L.; Du, Y.; Li, Y.; Lee, H.; Zheng, H. S.; Ghafouri, A.; Menegali, M.; Huang, Y.; Krikun, M.; Lepikhin, D.; Qin, J.; Chen, D.; Xu, Y.; Chen, Z.; Roberts, A.; Bosma, M.; Zhao, V.; Zhou, Y.; Chang, C.-C.; Krivokon, I.; Rusch, W.; Pickett, M.; Srinivasan, P.; Man, L.; Meier-Hellstern, K.; Morris, M. R.; Doshi, T.; Santos, R. D.; Duke, T.; Soraker, J.; Zevenbergen,

B.; Prabhakaran, V.; Diaz, M.; Hutchinson, B.; Olson, K.; Molina, A.; Hoffman-John, E.; Lee, J.; Aroyo, L.; Rajakumar, R.; Butryna, A.; Lamm, M.; Kuzmina, V.; Fenton, J.; Cohen, A.; Bernstein, R.; Kurzweil, R.; Aguera-Arcas, B.; Cui, C.; Croak, M.; Chi, E.; and Le, Q. 2022. LaMDA: Language Models for Dialog Applications. arXiv:2201.08239.

Wang, A.; Song, L.; Liu, Q.; Mi, H.; Wang, L.; Tu, Z.; Su, J.; and Yu, D. 2023a. Search-engine-augmented dialogue response generation with cheaply supervised query production. *Artificial Intelligence*, 319: 103874.

Wang, A.; Song, L.; Xu, G.; and Su, J. 2023b. Domain Adaptation for Conversational Query Production with the RAG Model Feedback. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Findings of the Association for Computational Linguistics: EMNLP 2023*, 9129–9141. Singapore: Association for Computational Linguistics.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8: 229–256.

Xie, B.; Song, J.; Shao, L.; Wu, S.; Wei, X.; Yang, B.; Lin, H.; Xie, J.; and Su, J. 2023. From statistical methods to deep learning, automatic keyphrase prediction: A survey. *Information Processing & Management*, 60(4): 103382.

Yarowsky, D. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, 189–196.

Ye, S.; Hwang, H.; Yang, S.; Yun, H.; Kim, Y.; and Seo, M. 2023. In-Context Instruction Learning. arXiv:2302.14691.

Zhang, J.; and Zong, C. 2016. Exploiting Source-side Monolingual Data in Neural Machine Translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 1535–1545. Austin, Texas: Association for Computational Linguistics.

Zhang, Z.; Zhang, H.; Chen, K.; Guo, Y.; Hua, J.; Wang, Y.; and Zhou, M. 2021. Mengzi: Towards Lightweight yet Ingenious Pre-trained Models for Chinese. arXiv:2110.06696.

Zhou, H.; Xu, X.; Wu, W.; Niu, Z.; Wu, H.; Bao, S.; Wang, F.; and Wang, H. 2022. Link the world: Improving open-domain conversation with dynamic spatiotemporal-aware knowledge. *arXiv preprint arXiv:2206.14000*.

Zhou, H.; Zheng, C.; Huang, K.; Huang, M.; and Zhu, X. 2020. KdConv: A Chinese multi-domain dialogue dataset towards multi-turn knowledge-driven conversation. *arXiv preprint arXiv:2004.04100*.

Zhou, Y.; and Goldman, S. 2004. Democratic co-learning. In *16th IEEE International Conference on Tools with Artificial Intelligence*, 594–602. IEEE.

Zhou, Z.-H.; and Li, M. 2005. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on knowledge and Data Engineering*, 17(11): 1529–1541.