

MIGA: A Unified Multi-Task Generation Framework for Conversational Text-to-SQL

Yingwen Fu^{12*†}, Wenjie Ou^{2*‡}, Zhou Yu³, and Yue Lin²

¹ Guangdong University of Foreign Studies, Guangzhou, China

² NetEase Games AI Lab, Guangzhou, China

³ Columbia University

20201010002@gdufs.edu.cn, zy2461@columbia.edu

{ouwenjie, gzlinyue}@corp.netease.com

Abstract

Conversational text-to-SQL is designed to translate multi-turn natural language questions into their corresponding SQL queries. Most advanced conversational text-to-SQL methods are incompatible with generative pre-trained language models (PLMs), such as T5. In this paper, we present a two-stage unified **Multi-task Generation frAamework** (MIGA) that leverages PLMs' ability to tackle conversational text-to-SQL. In the pre-training stage, MIGA first decomposes the main task into several related sub-tasks and then unifies them into the same sequence-to-sequence (Seq2Seq) paradigm with task-specific natural language prompts to boost the main task from multi-task training. Later in the fine-tuning stage, we propose four SQL perturbations to alleviate the error propagation problem. MIGA tends to achieve state-of-the-art performance on two benchmarks (SparC and CoSQL). We also provide extensive analyses and discussions to shed light on some new perspectives for conversational text-to-SQL.

Introduction

As an important branch of semantic parsing, the text-to-SQL task (Zhong, Xiong, and Socher 2017; Xu, Liu, and Song 2017) aims to automatically generate SQL queries according to natural language questions. It could break through the barrier for non-expert users to write SQL queries and interact with database systems. While in practical scenarios, users tend to obtain target information from the system step-by-step through conversations. To meet this demand, conversational text-to-SQL has recently drawn considerable attention. As shown in Figure 1, it extends text-to-SQL into multi-turn settings by generating interactive queries based on conversations.

Recently, generative pre-trained language models (PLMs) represented by T5 (Raffel et al. 2020) achieve competitive performance on multiple sequence-to-sequence (Seq2Seq) tasks. However, as a typical Seq2Seq task, most advanced

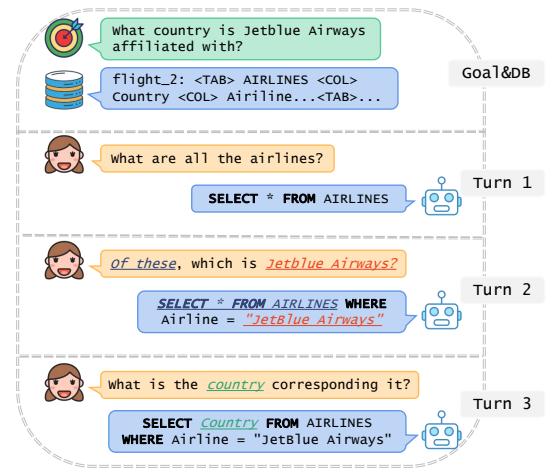


Figure 1: An example of conversational text-to-SQL.

methods (Cai and Wan 2020; Hui et al. 2021; Chen et al. 2021; Yu et al. 2021b; Zheng et al. 2022; Li et al. 2021; Xiao et al. 2022) for conversational text-to-SQL are based on tree-based decoders (Yin and Neubig 2017; Scholak, Schucher, and Bahdanau 2021). While effective, these methods require custom transition grammars which need additional overhead when being extended to new SQL structures. Additionally, the custom decoder structures make them incompatible with the generative PLMs. On the contrary, some recent attempts such as UNIFIEDSKG (Xie et al. 2022), PICARD (Scholak, Schucher, and Bahdanau 2021), and RASAT (Qi et al. 2022) have witnessed the potential of generative PLMs for this task. Another line of research on language pre-training (Phang, Févry, and Bowman 2018; Aghajanyan et al. 2021; Su et al. 2022; Deng et al. 2023) demonstrates that pre-training with intermediate-labeled data of related tasks can help improve the fine-tuned models. This suggests that we could boost the model for conversational text-to-SQL by pre-training on its related tasks.

In this paper, inspired by in-context learning (Brown et al. 2020), we propose a two-stage unified **Multi-task Generation frAamework** (MIGA) for conversational text-to-

*These authors contributed equally.

†This work was conducted when Yingwen Fu was interning at NetEase Games AI Lab.

‡Corresponding author.

Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

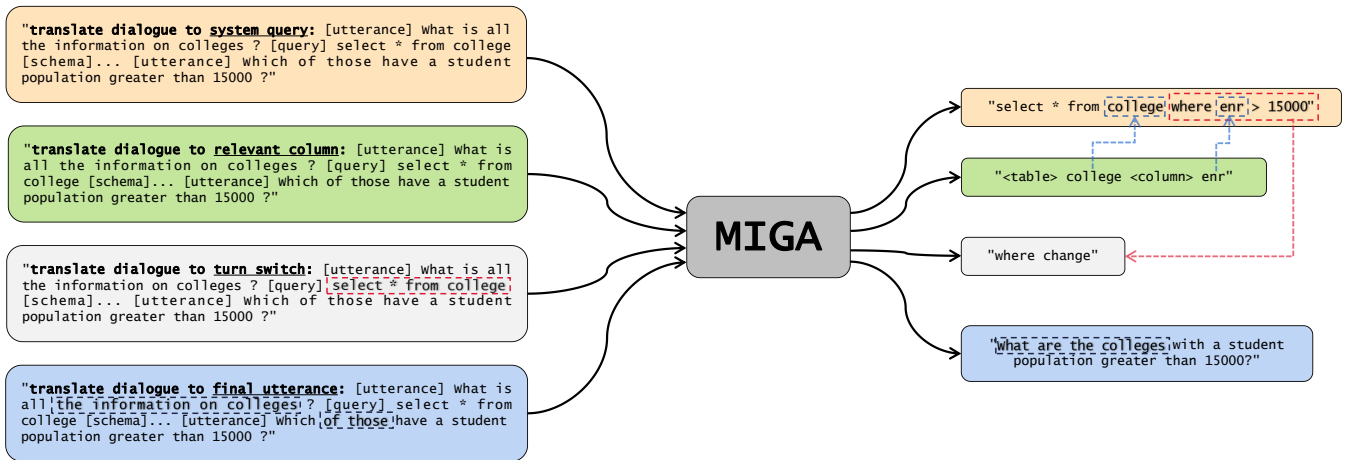


Figure 2: Overview of the proposed framework. In the multi-task pre-training stage, we train the model with four tasks: SQL Generation (SG, orange box), Related Schema Prediction (RSP, green box), Turn Switch Prediction (TWP, gray box), and Final Utterance Prediction (FUP, blue box). We unify them into the same Seq2Seq paradigm and jointly train them with task-specific prompts. "[utterance]", "[query]", and "[schema]" respectively indicate the user utterance, SQL query, and the schema of the target database.

SQL based on the pre-trained T5 models. MIGA consists of multi-task pre-training and single-task fine-tuning stages. In the pre-training stage (Stage 1), we first decompose the main task conversational text-to-SQL (abbreviated as SQL generation (SG) henceforth) into three sub-tasks and then unify them in a Seq2Seq paradigm by inserting task-specific natural language prompts into the input text to indicate each task. The model is jointly trained in the main task as well as three sub-tasks in this stage.

- **Related Schema Prediction (RSP).** A common error in the generated SQL is generating wrong tables and columns. Inspired by SCoRe (Yu et al. 2021b), RSP is defined as generating the tables and columns that appeared in the current SQL query. This sub-task only needs to consider the to-be-appeared columns and tables without taking the generated SQL structure into account, which can effectively alleviate the above issue. As shown in Figure 2, the RSP output for the current turn indicates the appeared table "college" and column "enr" in the SQL query "select * from college where enr > 15000".
- **Turn Switch Prediction (TWP).** Followed by (Li et al. 2021), TWP is introduced to enhance the model by modeling the conversation flow between the adjacent SQL query pair. This task is to predict what modification is made to the current SQL compared with its last SQL. In Figure 2, the current TWP output indicates that compared with the last SQL, the WHERE sub-clause is modified as "where enr > 15000". Thus we simplify this modification as "where change".
- **Final Utterance Prediction (FUP).** As stated in (Xiao et al. 2022), Conversational Question Reformulation (CQR) can effectively improve the model by alleviating the contextual dependency. In this paper, similar to CQR, we propose FUP to generate the final utterance

for the whole interaction. The FUP output in Figure 2 summarizes the information of the two-turn conversation by replacing "which of those" with "what are the colleges".

Such a unified framework provides a general and flexible paradigm that can handle any tasks whose input and output can be recast as a sequence of tokens. It serves some advantages: (1) referring to the steps humans write SQL, the conversational text-to-SQL task is decomposed into multiple sub-tasks, allowing the main task to learn from them; (2) the training samples are constructed as the format of T5 which can maximize the potential of the pre-trained T5 models for the target task; and (3) the unified framework allows plug-and-play of multiple auxiliary tasks. When applying to a specific task, we fine-tune the pre-trained model above (Stage 2) using the same training objective in the task-specific labeled data. When predicting the current-turn SQL, the previously generated SQLs are concatenated in the current model input, which may cause error propagation problem. To alleviate this, we propose to perturb the previous-turn ground truth SQLs according to four common generation errors, aiming to reduce the dependency of previous SQLs.

We evaluate MIGA on two conversational text-to-SQL benchmarks SparC and CoSQL. Comparisons against several advanced approaches show that MIGA tends to achieve state-of-the-art (SOTA) in two datasets based on Interaction Match (IM) accuracy. In summary, our contributions are:

1. We propose a novel framework, namely MIGA, that decomposes conversational text-to-SQL into three sub-tasks and unify them into a Seq2Seq paradigm. This can effectively facilitate the main task learning. The auxiliary tasks are trained in a plug-and-play manner, which makes the framework easily accommodate more auxiliary tasks.
2. We propose four SQL perturbations when fine-tuning the

model, which can effectively alleviate the error propagation problem.

- Experimental results on two benchmark datasets show that MIGA tends to achieve SOTA performance on IM accuracy. Extensive analyses provide some new insights into the proposed framework and some takeaways for future conversational text-to-SQL studies.

Related Work

Text-to-SQL Task

As a representative Seq2Seq task, context-independent text-to-SQL represented by Spider (Yu et al. 2018) benchmark has witnessed multiple meaningful advances in encoder (Wang et al. 2020; Cai et al. 2021; Cao et al. 2021), decoder (Yin and Neubig 2017; Scholak, Schucher, and Bahdanau 2021), and language pre-training (Deng et al. 2021; Yu et al. 2021a,b). With the widespread popularity of dialogue research, conversational text-to-SQL benchmarks SparC (Yu et al. 2019b) and CoSQL (Yu et al. 2019a) have drawn much attention. In particular, EditSQL (Zhang et al. 2019) and IST-SQL (Wang et al. 2021a) propose to leverage previous generated SQL to improve the SQL quality in the current turn. DELTA (Chen et al. 2021) and CQR-SQL (Xiao et al. 2022) focus on the application of CQR (Pan et al. 2019) and context-independent text-to-SQL to alleviate the difficulty of multi-turn SQL generation. RAT-SQL-TC (Li et al. 2021) and HIE-SQL (Zheng et al. 2022) aim to utilize the interaction information between the current and previous text as well as the SQL query to enhance the model performance respectively by two auxiliary objectives and a bimodal pre-trained model of natural language and SQL.

Recently, generative Seq2Seq pre-trained language models such as T5 (Raffel et al. 2020) have unveiled their potential for multiple generation tasks. UNIFIEDSKG (Xie et al. 2022) fine-tunes a pre-trained T5-3B model with text-to-SQL data that can yield results competitive to the then-state-of-the-art. Based on the T5 architecture, RASAT (Qi et al. 2022) uses an augmented relation-aware self-attention to enhance the interaction between the database schema and questions. Different from the methods above, this paper concentrates on improving the conversational text-to-SQL performance without additional annotated training data or specific model structures.

Pre-training on Supplementary Tasks

When further training is applied to tasks containing intermediate-labeled data (Phang, Févry, and Bowman 2018; Aghajanyan et al. 2021; Su et al. 2022; Deng et al. 2023), the fine-tuned models have been shown to perform much better on a variety of tasks, including the GLUE benchmark (Wang et al. 2018) and conversational recommender systems (Lei et al. 2020; Li et al. 2018). Specifically, PPTOD (Su et al. 2022) proposes to enhance the task-oriented dialogue system by supplementarily training on three sub-tasks: end-to-end dialogue modeling, dialogue state tracking, and intent classification. Inspired by PPTOD, we study a similar supplementary training setup with three well-designed sub-tasks for conversational text-to-SQL.

Algorithm 1: Training Process of MIGA

Input: Training set $\mathcal{D}_1 = \{\mathcal{D}_{SG}, \mathcal{D}_{RSP}, \mathcal{D}_{TWP}, \mathcal{D}_{FUP}\} = \{(Z_T, X, y)_i\}_{i=1}^{|\mathcal{D}_1|}$; Pre-trained T5 model θ ; Pre-training epoch e_1 ; Fine-tuning epoch e_2 ;
Output: Trained model θ

- # Multi-task pre-training
- for** $epoch=1, 2, 3, \dots, e_1$ **do**
- for** Batch \mathcal{B} in $Shuffle(\mathcal{D}_1)$ **do**
- Optimize model θ with $\mathcal{B} = \{(Z_T, X, y)_i\}_{i=1}^{|\mathcal{B}|}$ in \mathcal{L}_θ
- end for**
- end for**
-
- # Single-task fine-tuning
- Perturb \mathcal{D}_{SG} with probabilities α and β
- for** $epoch=1, 2, 3, \dots, e_2$ **do**
- for** Batch \mathcal{B} in $Shuffle(\mathcal{D}_{SG})$ **do**
- Optimize model θ with $\mathcal{B} = \{(Z_T, X, y)_i\}_{i=1}^{|\mathcal{B}|}$ in \mathcal{L}_θ
- end for**
- end for**

Methodology

Algorithm 1 gives the overall training process of the proposed MIGA framework in this paper, which consists of multi-task pre-training and single-task fine-tuning stages.

Task Formulation

Given the current user utterance u_t , the interaction history $h_t = [u_1, u_2, \dots, u_{t-1}]$, the schema of the target database $D = \langle T, C \rangle$, our goal is to predict the current SQL query s_t . Specifically, D consists of a set of tables $T = \{t_i\}_{i=1}^{|T|}$ and columns $C = \{C_i\}_{i=1}^{|C|}$, where for table t_i , its corresponding columns C_i are denoted as $C_i = \{c_{ij}\}_{j=1}^{|C_i|}$.

Multi-task Pre-training

Sample Construction In this stage, in addition to SG, we propose three auxiliary sub-tasks to improve the main task performance. As shown in Figure 2, motivated by the T5 training strategy, a training sample input for a specific target task is appended to a task-specific prompt. It is defined as:

$$\mathcal{S} = (Z_T, X, y) \quad (1)$$

where $T \in \{SG, RSP, TWP, FUP\}$ stands for the task to which the sample belongs. The task definition is specified in the next section. Z_T denotes the natural language prompt that employs a guidance sentence to indicate each task as shown in Figure 2.

X is the model input which is the concatenation of the current user utterance, previous user utterances, and SQL queries, as well as the schema. It is arranged as:

$$X = [u_1, s_1, u_2, s_2, \dots, u_t, D] \quad (2)$$

The schema D is formatted as follows:

$$D = [t_1, c_{11}, c_{12}, \dots, t_2, c_{21}, c_{22}, \dots, \dots] \quad (3)$$

And y is the model output text for task T , which is simplified in Figure 2.

Dataset	Interaction Num		Question-SQL Num		RSP Num		FUP Num		TWP Num	
	Train	Dev	Train	Dev	Train	Dev	Train	Dev	Train	Dev
Spider	/	/	7000	1034	7000	1034	/	/	/	/
SparC	3034	422	9025	1203	9025	1203	2960	410	5871	765
CoSQL	2164	292	7485	1008	7485	1008	469	69	5092	704

Table 1: Detailed data statistics on training and evaluation. Since Spider is a single-turn dataset, experimental data involving interactions are treated as ignored for this dataset (marked with ”/”).

Training Objective The model is trained with a unified maximum likelihood objective for each training sample. Given a training sample $\mathcal{S} = (Z_T, X, y)$, its loss is calculated as:

$$\mathcal{L}_\theta = - \sum_{l=1}^L \log P_\theta(y_l | y_{<l}; Z_T, X) \quad (4)$$

where L denotes the length of the target sequence and θ is the model parameters.

Training Tasks The pre-training stage focuses on four tasks: the main task SG as well as the three auxiliary tasks RSP, TWP, and FUP.

- **RSP** is defined as generating the tables and columns that appeared in the current SQL query. This sub-task only needs to consider the to-be-appeared columns and tables without taking the generated SQL structure into account, which can effectively prevent the model from confusing some similar tables and columns.
- **TWP** is introduced to enhance the model by modeling the conversation flow between the adjacent SQL query pair. This task is to predict what modification is made to the current SQL compared with its last SQL. Notably, TWP is originally modeled as a binary classification for each type while it is modeled as a generation task in this paper. If there are multiple modification operations, each operation is separated by the special token $\langle /s \rangle$.
- **FUP** is proposed to generate the final utterance for the whole conversation. The final utterance comes from the original dataset that aims to summarize the information of the whole conversation, so it could be regarded as CQR for the last-turn user utterance in this paper.

Single-task Fine-tuning

After the pre-training stage, we further fine-tune the model with the task-specific labeled data with the same learning objective Eq.4.

SQL Perturbation Since we consider the SQL information from previous turns when generating the current SQL, the model would be hampered by the error propagation problem. Therefore, we introduce SQL perturbation to reduce the over-dependency of the model on previously generated SQL queries. Specifically, we would replace the previous ground truth SQLs with their corresponding perturbed SQLs in the input with a probability α . When constructing the perturbed SQL, we randomly sample β tokens of the ground truth SQL and conduct one of the following perturbations:

- Randomly modify or add columns of the same table in SELECT sub-clause.
- Randomly modify the table in the JOIN structure (i.e., swap “t1” and “t2”).
- Randomly modify the “*” column to other columns.
- Swap “asc” and “desc”.

Inference

During the inference process, we utilize the fine-tuned model to predict the SQL queries for each conversation. When predicting the current SQL, different from the training step where the previous ground truth SQLs or perturbed SQLs are concatenated, we only concatenate the previously predicted SQLs to simulate real application scenarios.

Experiments

Datasets

Pre-training Datasets We use context-independent dataset Spider (Yu et al. 2018), conversational datasets SparC (Yu et al. 2019b), CoSQL (Yu et al. 2019a) for the pre-training stage. We create intermediate-labeled data for the proposed three auxiliary tasks on the three datasets above. Specifically, we only perform RSP and SG tasks for Spider because it is context-independent while conducting all three tasks for SparC and CoSQL. Detailed data statistics are shown in Table 1.

Evaluation Datasets Since we focus on the conversational text-to-SQL task, SparC and CoSQL datasets are utilized to evaluate the proposed MIGA performance.

Experimental Setups

Evaluation Metrics Following previous works (Xie et al. 2022; Qi et al. 2022; Xiao et al. 2022), Interaction Match (IM), which requires all output SQL queries in interaction to be valid, is the primary metric used to assess model performance. Every single question’s accuracy is also assessed using Question Match (QM).

Implementation We implement MIGA based on HuggingFace’s Transformers¹ with PyTorch² in 2 NVIDIA A100 GPUs. The models are initialized with the pre-trained T5 model of 3B size³. Epochs and learning rates for 2 training steps are set as (15, 50) and (1e-4, 5e-5). Batch size and

¹<https://github.com/huggingface/transformers>

²<https://pytorch.org/>

³<https://huggingface.co/t5-3B>

Method	QM(%)	IM(%)
CD-seq2seq (Yu et al. 2019b)	21.9	8.1
EditSQL (Zhang et al. 2019)	47.2	29.5
IST-SQL (Wang et al. 2021a)	47.6	29.9
GAZA (Zhong et al. 2020)	48.9	29.7
IGSQL (Cai and Wan 2020)	50.7	32.5
RichContext (Liu et al. 2021)	52.6	29.9
TreeSQL V2 (Wang et al. 2021b)	52.6	34.4
R ² SQL (Hui et al. 2021)	54.1	35.2
DELTA (Chen et al. 2021)	58.6	35.6
SCORE (Yu et al. 2021b)	62.2	42.5
HIE-SQL (Zheng et al. 2022)	64.7	45.0
RAT-SQL+TC (Li et al. 2021)	64.1	44.1
CQR-SQL (Xiao et al. 2022)	67.8	48.1
UNIFIEDSKG (Xie et al. 2022)	61.5	41.9
RASAT (Qi et al. 2022)	64.2	43.8
RASAT-PICARD (Qi et al. 2022)	66.7	47.2
MIGA(Ours)	65.4	48.6
MIGA+PICARD(Ours)	67.3	48.9

Table 2: Main results on the SparC dev set. The methods in the middle box are all T5-3B-based.

optimizer are set as 64, and AdaFactor (Shazeer and Stern 2018) respectively. Following BERT (Kenton and Toutanova 2019), β in SQL perturbation is 15%. As for the other probability α , we set it as 0.15 in this paper after searching.

Main Results

Table 2 and Table 3 report the conversational text-to-SQL results of different models on SparC and CoSQL datasets. Our proposed MIGA achieves the best IM accuracy on two datasets, obtaining 48.9% and 29.8% IM accuracy on the development sets of SparC and CoSQL. The SOTA IM scores indicate the effectiveness of MIGA for the conversational text-to-SQL task. Specifically, our model is based on a single T5-based model architecture that does not require well-designed relation-aware encoders or tree-based decoders, as well as additional human-annotated data. The intermediate-labeled data used in this paper is self-constructed and derived from the original training samples. This demonstrates the benefits of our model.

Compared with T5-based methods (the middle boxes in Table 2 and Table 3), MIGA could obtain significant and consistent improvements in IM scores ranging from 1.7% to 7.0% and QM scores ranging from 0.6% to 5.8%. This indicates that our training method can stimulate the potential of the T5 pre-trained models for the target task to a greater extent. Besides, PICARD can well constrain the decoding steps of T5 models which can further boost the model performance. When augmented with PICARD, MIGA+PICARD brings a significant improvement on QM metric to the original MIGA. For a fair comparison, we compare MIGA against the version of RASAT *w/o* PICARD as reported in the original paper. The results indicate that MIGA significantly outperforms RASAT, especially in IM accuracy of

Method	QM(%)	IM(%)
EditSQL (Zhang et al. 2019)	39.9	12.3
GAZA (Zhong et al. 2020)	42.0	12.3
IGSQL (Cai and Wan 2020)	44.1	15.8
RichContext (Liu et al. 2021)	41.0	14.0
IST-SQL (Wang et al. 2021a)	44.4	14.7
R ² SQL (Hui et al. 2021)	45.7	19.5
DELTA (Chen et al. 2021)	51.7	21.5
SCORE (Yu et al. 2021b)	52.1	22.0
HIE-SQL (Zheng et al. 2022)	56.4	28.7
CQR-SQL (Xiao et al. 2022)	58.4	29.4
T5-3B (cholak et al. 2021)	53.8	21.8
PICARD (cholak et al. 2021)	56.9	24.2
UNIFIEDSKG (Xie et al. 2022)	54.1	22.8
RASAT (Qi et al. 2022)	56.2	24.9
RASAT-PICARD (Qi et al. 2022)	58.8	26.3
MIGA(Ours)	57.9	29.6
MIGA+PICARD(Ours)	59.0	29.8

Table 3: Main results on the CoSQL dev set. The methods in the middle box are all T5-3B-based.

4.8% without PICARD.

To further verify the SQL prediction effectiveness of the proposed MIGA, we compare the performance of some baseline methods with MIGA on different interaction turns and SQL difficulties. As shown in Tables 4 and 5, the accuracy of SQL prediction decreases as the interaction turn and difficulty increases due to the long-range dependency and complex SQL structures. MIGA achieves competitive performance on different turns and difficulties, outperforming most of the baseline models except CQR-SQL. Typically, Since CQR-SQL focuses on CQR, it can effectively alleviate the dependency of information extraction on the conversation and previous SQLs, thus achieving superior performance on QM metrics especially for later turns (Turn 3 and Turn 4) and high difficulties (Hard and Extra). This also indicates the importance of CQR for the target task. Unfortunately, as the original CQR data in CQR-SQL is a newly introduced dataset and it is not publicly released, it is impossible to include this data in our MIGA model. Moreover, the data size of the proposed FUP task in this paper is only 3908, and the data-hungry problem limits the ability of the CQR task to boost MIGA. How to extend the data size of the FUP task is left for future work.

Analysis

Ablation Study We conduct experiments on different variants of the proposed framework to investigate the contributions of different components. There are 5 different variants: 1) *w/o* RSP: we remove the RSP task in the multi-task pre-training stage; 2) *w/o* TWP: we remove the TWP task in the multi-task pre-training stage; 3) *w/o* FUP: we remove the FUP task in the multi-task pre-training stage; 4) *w/o* SQL perturbation: we remove SQL perturbation in the fine-tuning step and fine-tune the model as in the pre-training stage; and

Method	Turn 1	Turn 2	Turn 3	Turn 4+
EditSQL	62.2	40.1	36.1	19.3
IG-SQL	63.2	50.8	39.0	26.1
R ² SQL	67.7	55.3	45.7	33.0
RAT-SQL+TC	75.4	64.0	54.4	40.9
CQR-SQL	75.6	68.7	58.9	53.9
MIGA+PICARD	74.9	68.5	58.9	52.8

Table 4: Detailed QM accuracy performance on different interaction turns of the SparC dev set.

Method	Easy	Medium	Hard	Extra
EditSQL	68.8	40.6	26.9	12.8
IG-SQL	70.9	45.4	29.0	18.8
R ² SQL	75.5	51.5	35.2	21.8
CQR-SQL	80.7	68.3	46.2	43.3
MIGA+PICARD	81.8	66.7	44.8	41.8

Table 5: Detailed QM accuracy performance on different SQL difficulties of the SparC dev set.

5) *w/o Spider/CoSQL*: we remove the training samples of the corresponding dataset in the pre-training stage. All the variants are evaluated on the SparC dev set.

Table 6 shows the results of the variants and MIGA. Generally, all components contribute to the target task. The full model consistently achieves the best performance on the SparC dev set. Some in-depth analyses are shown as follows.

- Removing the RSP task leads to performance drops of 4.0% and 4.4% in QM and IM metrics respectively. This task can effectively model the information of the tables and columns that appeared in the current SQL, which further avoids the confusion of similar columns and tables.
- If removing the TWP task, two metrics on SparC drop by 3.4% and 3.2%. TWP can obtain information about the interaction between the questions and SQLs, thus reducing the difficulty of the current SQL prediction. However, this task models the modifications of the previously generated SQL which may cause error propagation.
- Adding the FUP task can get a performance gain of 2.1% and 1.8% on two metrics. However, the training sample size for this task is relatively small, resulting in a relatively small improvement. Constructing more samples for this task may further improve the model.
- SQL perturbation can significantly alleviate the error propagation problem, which achieves performance gains of (1.0%, 1.0%) on two metrics.
- Removing either of the pre-trained datasets leads to a performance drop. Specifically, the CoSQL dataset seems to contribute more than the Spider dataset on IM metric, for Spider is a context-independent dataset which could only boost the first-turn SQL prediction potentially.

Model Size Impact To evaluate the proposed framework on pre-trained models with different sizes, we implement MIGA and fine-tune the vanilla T5 pre-trained models on 3

Model	QM	IM
MIGA	65.4	48.6
<i>w/o RSP</i>	61.4 (-4.0)	44.2 (-4.4)
<i>w/o TWP</i>	62.0 (-3.4)	45.4 (-3.2)
<i>w/o FUP</i>	63.3 (-2.1)	46.8 (-1.8)
<i>w/o SQL perturbation</i>	64.4 (-1.0)	47.6 (-1.0)
<i>w/o Spider</i>	63.8 (-1.6)	47.3 (-1.3)
<i>w/o CoSQL</i>	63.8 (-1.6)	45.8 (-2.8)

Table 6: Ablation study.

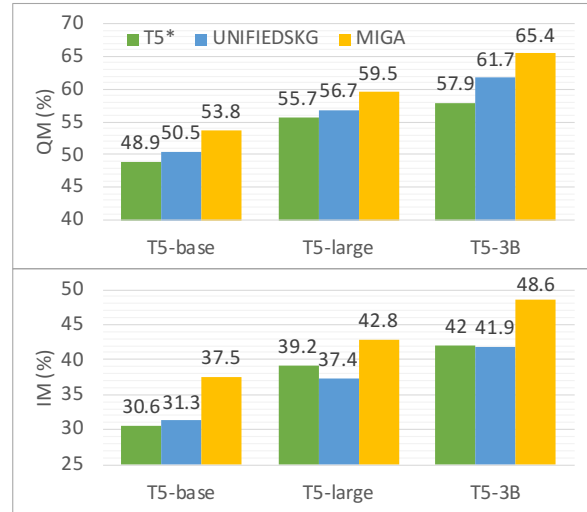


Figure 3: Results on the SparC dev set for different T5 pre-trained model sizes. * indicates our re-implementation.

different model sizes (T5-base, T5-large, T5-3B) and evaluate them on the SparC dev set. The results are shown in Figure 3. Commonly, when the model size increases, the model performance also increases significantly, demonstrating the powerful semantic modeling ability of large PLMs. In addition, the base-size MIGA model outperforms the large-size vanilla fine-tuned T5 model, and the same situation occurs for the large-size MIGA, which again illustrates the effectiveness of MIGA. The three proposed auxiliary tasks effectively stimulate the potential of the T5 pre-trained models and add more training data. This can encourage the pre-trained model to better adapt to the target task.

Case Study Figure 4 presents two cases derived from the SparC dev set and the corresponding predicted SQL queries of the vanilla fine-tuned T5-3B and MIGA. For *Case #1*, in *Question #2*, T5-3B fails to generate valid SQL for relatively complex JOIN structures (two-table join), which in turn leads to the failure of more complex JOIN structures (three-table join) in *Question #3*. While MIGA accurately predicts the JOIN structure and maintains the previous condition `t1.sex="f"` based on the extracted turn switch and related column information. For *Case #2*, T5-3B tends to confuse the columns of different tables and misidentifies `earnings` as a column of table `people`,

Case #1 Goal	How many dog pets are raised by female students?
Question #1	Who are the female students?
T5-3B & MIGA	<code>SELECT * FROM student WHERE sex = "f"</code> ✓
Question #2	Of those, who has a pet?
T5-3B	<code>SELECT * FROM student WHERE sex = "f" INTERSECT SELECT * FROM has_pet AS t1 JOIN student AS t2 ON t1.stuid = t2.stuid WHERE t2.sex = "f"</code> ✗
MIGA	<code>SELECT * FROM student AS t1 JOIN has_pet AS t2 ON t1.stuid = t2.stuid WHERE t1.sex = "f"</code> ✓
Question #3	How many of those have dogs?
T5-3B	<code>SELECT COUNT (*) FROM student WHERE sex = "f" INTERSECT SELECT * FROM has_pet AS t1 JOIN pets AS t2 ON t1.petid = t2.petid WHERE t2.petype = "dog"</code> ✗
MIGA	<code>SELECT COUNT (*) FROM student as t1 join has_pet AS t2 on t1.stuid = t2.stuid JOIN pets AS t3 ON t2.petid = t3.petid WHERE t3.petype = "dog" and t1.sex = "f"</code> ✓
Case #2 Goal	What is the average earnings of poker players with height higher than 200?
Question #1	What is the height of each poker player?
T5-3B & MIGA	<code>SELECT t2.height FROM poker_player AS t1 JOIN people AS t2 ON t1.people_id = t2.people_id</code> ✓
Question #2	Who all have heights greater than 200?
T5-3B & MIGA	<code>SELECT * FROM poker_player AS t1 JOIN people AS t2 ON t1.people_id = t2.people_id WHERE t2.height > 200</code> ✓
Question #3	What is the average earnings among them?
T5-3B	<code>SELECT AVG (t2.earnings) FROM poker_player AS t1 JOIN people AS t2 ON t1.people_id = t2.people_id WHERE t2.height > 200</code> ✗
MIGA	<code>SELECT AVG (t1.earnings) FROM poker_player AS t1 JOIN people AS t2 ON t1.people_id = t2.people_id WHERE t2.height > 200</code> ✓

Figure 4: Case studies on the SparC dev set. MIGA gives correct predictions in two cases while the vanilla T5-3B model fails.

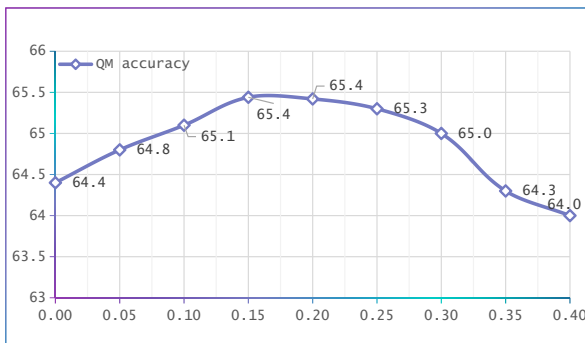


Figure 5: Results on the SparC dev set of different α values. The experiments are conducted on MIGA with T5-3B size.

while MIGA can correctly identify the column belongs to table `poker_player` and link it to `t1` by modeling the table and column information.

Searching of Hyper-parameter α When no SQL perturbation is used, the QM accuracy of MIGA with 3B size on the SparC dev set is 64.4%. To simulate the real error scenario, we set the search range as $[0.0 - 0.4]$ and use QM accuracy as the metric for parameter selection. As shown in Figure 5, when the α value is 0.15, the QM score of the model reaches the peak. In addition, when it exceeds 0.35, the model performance starts to be inferior to the model performance without SQL perturbation.

Limitations and Future Directions

One main concern is that the QM accuracy of our model is potentially inferior to some SOTA methods. One possi-

ble reason is that concatenating previously predicted SQL into the current input may cause error propagation. This results in the model prediction somewhat being dependent on the first-turn predicted SQL, i.e., if the first-turn SQL is predicted correctly, the later-turn SQLs tend to be predicted correctly, and vice versa. To further explore this issue, we input the correct first-turn SQL to evaluate the predicted SQLs of subsequent turns. We found that if the first-turn SQL is correct, the accuracy of the subsequent predicted SQLs (especially the second turn) is greatly improved, thus effectively improving the IM score as well. This indicates that errors in the first several steps have a huge impact on the final results. To resolve this, we can either improve the accuracy of the first-turn SQL prediction or reduce the dependence of the current-turn SQL prediction on the previous SQL prediction. Although our proposed SQL perturbation could alleviate this problem, we plan to further reduce it in the future.

Conclusions

In this paper, we propose MIGA, a two-stage unified multi-task generation framework, for conversational text-to-SQL. Specifically, in the pre-training stage, we decompose the target task into three sub-tasks and unify them into the same sequence-to-sequence (Seq2Seq) paradigm by adding a task-specific natural language prompt into the input text to indicate each task. Later to adapt the model to the target benchmark, we fine-tune it in the task-specific labeled data and introduce four SQL perturbations to alleviate the error propagation problem. Extensive experiments and analyses on two conversational text-to-SQL benchmarks indicate the effectiveness of MIGA. In future work, we would further explore the strategies to tackle the error propagation problem and extend the data size of the FUP task.

References

- Aghajanyan, A.; Gupta, A.; Shrivastava, A.; Chen, X.; Zettlemoyer, L.; and Gupta, S. 2021. Muppet: Massive Multi-task Representations with Pre-Finetuning. In *EMNLP*, 5799–5811.
- Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. In *NIPS*, 1877–1901.
- Cai, R.; Yuan, J.; Xu, B.; and Hao, Z. 2021. SADGA: Structure-Aware Dual Graph Aggregation Network for Text-to-SQL. In *NIPS*, 7664–7676.
- Cai, Y.; and Wan, X. 2020. IGSQ: Database Schema Interaction Graph Based Neural Model for Context-Dependent Text-to-SQL Generation. In *EMNLP*, 6903–6912.
- Cao, R.; Chen, L.; Chen, Z.; Zhao, Y.; Zhu, S.; and Yu, K. 2021. LGESQL: Line Graph Enhanced Text-to-SQL Model with Mixed Local and Non-Local Relations. In *ACL-IJCNLP*, 2541–2555.
- Chen, Z.; Chen, L.; Li, H.; Cao, R.; Ma, D.; Wu, M.; and Yu, K. 2021. Decoupled Dialogue Modeling and Semantic Parsing for Multi-Turn Text-to-SQL. In *Findings of ACL-IJCNLP*, 3063–3074.
- Deng, X.; Hassan, A.; Meek, C.; Polozov, O.; Sun, H.; and Richardson, M. 2021. Structure-Grounded Pretraining for Text-to-SQL. In *NAACL*, 1337–1350.
- Deng, Y.; Zhang, W.; Xu, W.; Lei, W.; Chua, T.-S.; and Lam, W. 2023. A Unified Multi-Task Learning Framework for Multi-Goal Conversational Recommender Systems. *ACM Trans. Inf. Syst.*, 41(3).
- Hui, B.; Geng, R.; Ren, Q.; Li, B.; Li, Y.; Sun, J.; Huang, F.; Si, L.; Zhu, P.; and Zhu, X. 2021. Dynamic hybrid relation exploration network for cross-domain context-dependent semantic parsing. In *AAAI*, 13116–13124.
- Kenton, J. D. M.-W. C.; and Toutanova, L. K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*, 4171–4186.
- Lei, W.; He, X.; Miao, Y.; Wu, Q.; Hong, R.; Kan, M.-Y.; and Chua, T.-S. 2020. Estimation-action-reflection: Towards deep interaction between conversational and recommender systems. In *WSDM*, 304–312.
- Li, R.; Kahou, S. E.; Schulz, H.; Michalski, V.; Charlin, L.; and Pal, C. 2018. Towards Deep Conversational Recommendations. In *NIPS*, 9748–9758.
- Li, Y.; Zhang, H.; Li, Y.; Wang, S.; Wu, W.; and Zhang, Y. 2021. Pay More Attention to History: A Context Modeling Strategy for Conversational Text-to-SQL. *CoRR*, abs/2112.08735.
- Liu, Q.; Chen, B.; Guo, J.; Lou, J.-G.; Zhou, B.; and Zhang, D. 2021. How far are we from effective context modeling? an exploratory study on semantic parsing in context. In *IJCAI*, 3580–3586.
- Pan, Z.; Bai, K.; Wang, Y.; Zhou, L.; and Liu, X. 2019. Improving open-domain dialogue systems via multi-turn incomplete utterance restoration. In *EMNLP-IJCNLP*, 1824–1833.
- Phang, J.; Févry, T.; and Bowman, S. R. 2018. Sentence Encoders on STILTs: Supplementary Training on Intermediate Labeled-data Tasks. *CoRR*, abs/1811.01088.
- Qi, J.; Tang, J.; He, Z.; Wan, X.; Cheng, Y.; Zhou, C.; Wang, X.; Zhang, Q.; and Lin, Z. 2022. RASAT: Integrating Relational Structures into Pretrained Seq2Seq Model for Text-to-SQL. In *EMNLP*, 3215–3229.
- 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. *J. Mach. Learn. Res.*, 21: 140:1–140:67.
- Scholak, T.; Schucher, N.; and Bahdanau, D. 2021. PICARD: Parsing Incrementally for Constrained Auto-Regressive Decoding from Language Models. In *EMNLP*, 9895–9901.
- Shazeer, N.; and Stern, M. 2018. Adafactor: Adaptive Learning Rates with Sublinear Memory Cost. In *ICML*, 4603–4611. PMLR.
- Su, Y.; Shu, L.; Mansimov, E.; Gupta, A.; Cai, D.; Lai, Y.-A.; and Zhang, Y. 2022. Multi-Task Pre-Training for Plug-and-Play Task-Oriented Dialogue System. In *ACL*, 4661–4676.
- Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *EMNLP-WS*, 353–355.
- Wang, B.; Shin, R.; Liu, X.; Polozov, O.; and Richardson, M. 2020. RAT-SQL: Relation-Aware Schema Encoding and Linking for Text-to-SQL Parsers. In *ACL*, 7567–7578.
- Wang, R.-Z.; Ling, Z.-H.; Zhou, J.; and Hu, Y. 2021a. Tracking interaction states for multi-turn text-to-sql semantic parsing. In *AAAI*, 13979–13987.
- Wang, X.; Wu, S.; Shou, L.; and Chen, K. 2021b. An Interactive NL2SQL Approach with Reuse Strategy. In *DASFAA*, 280–288. Springer.
- Xiao, D.; Chai, L.; Zhang, Q.-W.; Yan, Z.; Li, Z.; and Cao, Y. 2022. CQR-SQL: Conversational Question Reformulation Enhanced Context-Dependent Text-to-SQL Parsers. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, 2055–2068.
- Xie, T.; Wu, C. H.; Shi, P.; Zhong, R.; Scholak, T.; Yasunaga, M.; Wu, C.-S.; Zhong, M.; Yin, P.; Wang, S. I.; Zhong, V.; Wang, B.; Li, C.; Boyle, C.; Ni, A.; Yao, Z.; Radev, D.; Xiong, C.; Kong, L.; Zhang, R.; Smith, N. A.; Zettlemoyer, L.; and Yu, T. 2022. UnifiedSKG: Unifying and Multi-Tasking Structured Knowledge Grounding with Text-to-Text Language Models. In *EMNLP*, 602–631.
- Xu, X.; Liu, C.; and Song, D. 2017. SQLNet: Generating Structured Queries From Natural Language Without Reinforcement Learning. *CoRR*, abs/1711.04436.
- Yin, P.; and Neubig, G. 2017. A Syntactic Neural Model for General-Purpose Code Generation. In *ACL*, 440–450.
- Yu, T.; Wu, C.; Lin, X. V.; Wang, B.; Tan, Y. C.; Yang, X.; Radev, D. R.; Socher, R.; and Xiong, C. 2021a. GraPPa: Grammar-Augmented Pre-Training for Table Semantic Parsing. In *ICLR*. OpenReview.net.

Yu, T.; Zhang, R.; Er, H.; Li, S.; Xue, E.; Pang, B.; Lin, X. V.; Tan, Y. C.; Shi, T.; Li, Z.; et al. 2019a. CoSQL: A Conversational Text-to-SQL Challenge Towards Cross-Domain Natural Language Interfaces to Databases. In *EMNLP-IJCNLP*, 1962–1979.

Yu, T.; Zhang, R.; Polozov, A.; Meek, C.; and Awadallah, A. H. 2021b. SCoRe: Pre-Training for Context Representation in Conversational Semantic Parsing. In *ICLR*. OpenReview.net.

Yu, T.; Zhang, R.; Yang, K.; Yasunaga, M.; Wang, D.; Li, Z.; Ma, J.; Li, I.; Yao, Q.; Roman, S.; et al. 2018. Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task. In *EMNLP*, 3911–3921.

Yu, T.; Zhang, R.; Yasunaga, M.; Tan, Y. C.; Lin, X. V.; Li, S.; Er, H.; Li, I.; Pang, B.; Chen, T.; et al. 2019b. SParC: Cross-Domain Semantic Parsing in Context. In *ACL*, 4511–4523.

Zhang, R.; Yu, T.; Er, H.; Shim, S.; Xue, E.; Lin, X. V.; Shi, T.; Xiong, C.; Socher, R.; and Radev, D. 2019. Editing-Based SQL Query Generation for Cross-Domain Context-Dependent Questions. In *EMNLP-IJCNLP*, 5338–5349.

Zheng, Y.; Wang, H.; Dong, B.; Wang, X.; and Li, C. 2022. HIE-SQL: History Information Enhanced Network for Context-Dependent Text-to-SQL Semantic Parsing. In *Findings of ACL*, 2997–3007.

Zhong, V.; Lewis, M.; Wang, S. I.; and Zettlemoyer, L. 2020. Grounded Adaptation for Zero-shot Executable Semantic Parsing. In *EMNLP*, 6869–6882.

Zhong, V.; Xiong, C.; and Socher, R. 2017. Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning. *CoRR*, abs/1709.00103.