Cross-Lingual Text-to-SQL Semantic Parsing with Representation Mixup

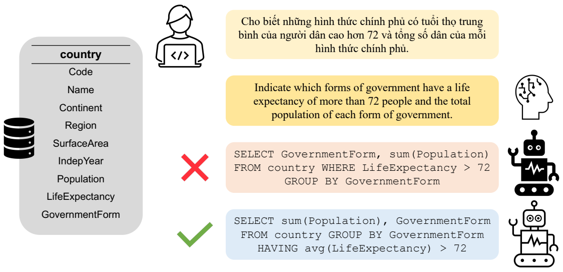
Peng Shi[♠], **Linfeng Song**[♣], **Lifeng Jin**[♣], **Haitao Mi**[♣],
He Bai[♠], **Jimmy Lin**[♠], and **Dong Yu**[♣]

[♠] University of Waterloo [♣] Tencent AI Lab, Bellevue, WA, USA

peng.shi@uwaterloo.ca, lfsong@tencent.com

Abstract

We focus on the cross-lingual Text-to-SQL semantic parsing task, where the parsers are expected to generate SQL for non-English utterances based on English database schemas. Intuitively, English translation as side information is an effective way to bridge the language gap, but noise introduced by the translation system may affect parser effectiveness. In this work, we propose a **Re**presentation Mi**x**up Framework (**REX**) for effectively exploiting translations in the cross-lingual Text-to-SQL task. Particularly, it uses a general encoding layer, a transition layer, and a target-centric layer to properly guide the information flow of the English translation. Experimental results on CSPIDER and VSPIDER show that our framework can benefit from cross-lingual training and improve the effectiveness of semantic parsers, achieving state-of-the-art performance.



Cho biết những hình thức chính phủ có tuổi thọ trung bình của người dân cao hơn 72 và tổng số dân của mỗi hình thức chính phủ.

Indicate which forms of government have a life expectancy of more than 72 people and the total population of each form of government.

```
SELECT GovernmentForm, sum(Population)
FROM country WHERE LifeExpectancy > 72
GROUP BY GovernmentForm
```

```
SELECT sum(Population), GovernmentForm
FROM country GROUP BY GovernmentForm
HAVING avg(LifeExpectancy) > 72
```

Figure 1: An illustration of the cross-lingual Text-to-SQL task. The utterance in English: "Find the government form name and total population for each government form whose average life expectancy is longer than 72.". The automatic translation fails to translate "trung bình" in Vietnamese into "average" in English.

1 Introduction

The task of semantic parsing is to translate natural language utterances into meaning representations, such as lambda calculus (Liang, 2013) or a programming language (Yin et al., 2018; Zhong et al., 2017; Yu et al., 2018). More recently, Text-to-SQL semantic parsing, using SQL queries as the meaning representation, has attracted increasing attention from both academia and industry researchers (Zhong et al., 2017; Wang et al., 2020; Yu et al., 2021; Deng et al., 2021; Shi et al., 2021a; Scholak et al., 2021).

Benefiting from recently annotated large-scale datasets (Zhong et al., 2017; Yu et al., 2018), research in Text-to-SQL has been greatly expedited. Moreover, due to the development of encoder-decoder pre-trained models (Lewis et al., 2020; Raffel et al., 2020), semantic parsers have been improved significantly, benefiting from contextualized representations (Lin et al., 2020; Scholak et al., 2021). However, these advances have been achieved mostly in English, leaving other languages underexplored. Systems that can handle non-English inputs well are in urgent need to enhance the user experience for non-English speakers. Nevertheless, the performance of current cross-lingual Text-to-SQL systems is still far from satisfactory. Taking Figure 1 as an example, a Vietnamese question is asked based on the English database schema, and the system is expected to generate the corresponding SQL query.

Because the utterances are non-English, English parsers cannot be directly applied. For tackling the cross-lingual issue, machine translation based methods can be effective solutions, e.g., first translating non-English utterances into English and then using English parsers to generate SQL queries. Here we denote these as *Translate-Test* methods. However, translation systems may introduce noise that causes further errors from the semantic parsers. For example, in Figure 1, Google Translate produces the English translation "Indicate which forms of government have a life expectancy of more than 72 people and the total population of each form of government." for the input Vietnamese utterance. One important information is missing in the translation process: "trung bình" should be translated

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 5296–5306
December 7-11, 2022 ©2022 Association for Computational Linguistics

into "average" but Google Translate fails to do so, resulting in the wrong `WHERE` condition prediction: `WHERE LifeExpectancy > 72` instead of `HAVING avg(LifeExpectancy) > 72`.

Another direction for solving the cross-lingual Text-to-SQL problem is to build *target language annotated datasets* and train a target language parser directly (Min et al., 2019; Tuan Nguyen et al., 2020). However, these methods fail to leverage knowledge from English parsers (learned from annotated English data), which has the potential to benefit non-English Text-to-SQL parsing.

In this work, we propose **REX**, a **Re**presentation mi**x**up framework for cross-lingual modeling that utilizes both the English data and the annotated data in the target language. First, REX adopts a two-stage training strategy where the target language models are first initialized with the pre-trained English parser and then trained with the target language data. Using this method, basic schema encoding ability and SQL decoding ability of the English parsers can be reused during target language training. Second, to further make use of English parsers' utterance encoding ability, we use English translations as context augmentation for bridging the cross-lingual gap and facilitating non-English model training.

Instead of simply concatenating the English translation and the target utterance, REX takes a general encoder, a transition layer, and a target-centric encoder to properly guide information flow of English translations, to mitigate the aforementioned issue around noisy translations. In detail, the general encoder generates contextual representations for bilingual utterances and database schemas. The transition layer is leveraged to obtain a cross-lingual mixup representation of the target language utterance, aiming to make the best use of English translations while minimizing the noise introduced. Lastly, the target-centric encoder focuses on the interaction between the target language utterance and the source language schema, by ignoring the side effects caused by translations.

We test our REX framework on two non-English Text-to-SQL semantic parsing datasets, CSPIDER and VSPIDER. Experimental results on the benchmarks show that our framework can further improve upon simple yet effective baselines. At the time of publication, REX achieves state-of-the-art performance on the CSPIDER leaderboard based on the results on the hidden test set.

Our contributions are summarized as follows:

- We propose the REX framework for leveraging knowledge from English parsers and information from machine translation by using representation mixup to reduce the negative side effects of automatic translation.
- We conduct a detailed ablation study to show how different configurations of the REX framework affect parser effectiveness.
- Our framework obtains state-of-the-art performance on the CSPIDER and VSPIDER benchmarks. The implementation is available for future work.[1]

2 Problem Definition and Notation

Given an utterance $X = (x_1, x_2, ..., x_n)$ and a database schema S, a Text-to-SQL model is expected to translate the utterance into a valid SQL query. Our framework is based on a standard encoder-decoder architecture. For the task, we assume the existence of an internationalized database where the database schema S is in English. The natural language queries from users are not in English; we denote these non-English languages as target languages.

Formally, we denote $X_t = \{x_1, x_2, ..., x_{n_t}\}$ as an utterance in the target language with n_t tokens. Similarly, the utterance in the *source language*[2] with n_s tokens is denoted $X_s = \{x_1, x_2, ..., x_{n_s}\}$. We assume the database schema S contains several tables $T \in D$ with column names $C = \{c_1, c_2, ..., c_{|T|}\}$, where $|T|$ denotes the number of columns in table T.

3 Baseline: Single-source Input

Here we introduce a sequence-to-sequence based semantic parsing model and a baseline that leverages English translation for non-English Text-to-SQL semantic parsing.

The model for Text-to-SQL semantic parsing has been continuously improved in recent years (Yin and Neubig, 2017; Wang et al., 2019; Guo et al., 2019; Wang et al., 2020; Rubin and Berant, 2021; Shi et al., 2021b; Choi et al., 2021; Hui et al., 2022; Qi et al., 2022). Specifically, Scholak et al. (2021) utilize the pre-trained sequence-to-sequence model

[1] https://github.com/Impavidity/Rex

[2] In this work, the source language refers to English.

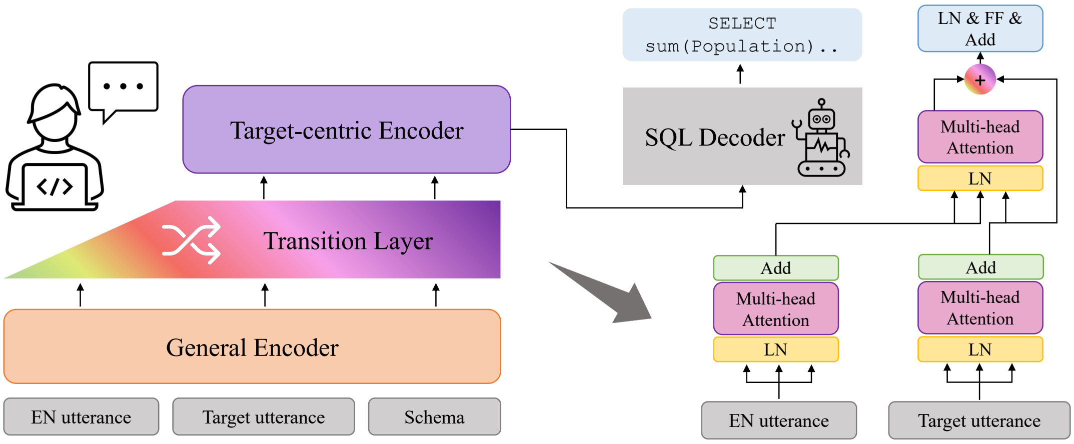


Figure 2: Illustration of the REX framework. The left part is the overall framework comprising a general encoder, a transition layer, and a target-centric encoder. The SQL query is decoded autoregressively from the SQL decoder. The right part demonstrates one of the implementations of the transition layer: Explicit Utterance Mixup.

T5 as the parser to directly generate SQL, simplifying the intermediate representation design for the grammar-based decoder. In detail, the model input is the concatenation of utterance X and linearized database schema S. With a pre-trained encoder, the contextualized utterance representation H_x and the database schema representation H_d can be obtained. The pre-trained decoder leverages these contextualized hidden states for generating SQL in an autoregressive fashion with constrained decoding. This architecture obtains state-of-the-art performance on English benchmarks without complex modeling. By replacing the T5 model with its multilingual version, mT5 (Xue et al., 2021), the model can be applied to non-English training data, to obtain parsers that have the ability to handle non-English utterances. We denoted this as single-source target-language training.

4 Representation Mixup Framework

Here, we propose a representation mixup framework for cross-lingual Text-to-SQL semantic parsing. As shown in Figure 2, the encoder stacks several general encoding layers with a transition layer, and some target-centric encoding layers.

First, utterances from the source and target languages can be encoded separately or jointly in the general encoder layers. Then, a transition layer implements the representation mixup between the input sequences, such as source language utterance, target language utterance, and database schema, in a specific layer. Then, a target-centric encoding layer will try to ignore the noise produced by machine translation systems by only focusing on the target language utterance. Finally, a SQL decoder leverages the information from the encoder component to generate a full SQL query.

4.1 Framework Overview

4.1.1 General Encoder

The general encoder is used to generate basic representations of the utterances, including the source language and target language, and the database schema. Here we discuss two different methods, namely independent encoder (Fang et al., 2021) and joint encoder. The general encoder is parameterized with m-layer transformers.

Independent General Encoder: Formally, the source and target language utterances can be encoded with m-layer transformers to obtain hidden representation H_s and H_t:

$$\begin{aligned} H_s^m &= \mathtt{Transformers_s}(X_s) \\ H_t^m &= \mathtt{Transformers_t}(X_t) \end{aligned} \tag{1}$$

The database schema is first linearized into the token sequence S, following the method used in Scholak et al. (2021). An m-layer transformer is applied on the linearized database schema tokens to obtain H_d:

$$H_d^m = \mathtt{Transformers_d}(S) \tag{2}$$

One benefit of independent encoding is that the representation of the schema can be shared and reused for all queries to the database, speeding up model inference. Note that the m-layer transformer parameters from different components can be either independent or tied.

Joint General Encoder: The interactions between the schema and the utterances are vital for training an effective semantic parser. Instead of encoding the information independently, joint encoding allows full information interaction between the utterances and schema. More specifically, the input of the joint encoder is the concatenation of the source language utterance, the target language utterance, and the schema:

$$H_s^m, H_t^m, H_d^m = \texttt{Transformers}([X_s; X_t; S]) \tag{3}$$

where [;] denotes the concatenation operation. Compared to the independent general encoder, this design requires re-encoding of the schema for each natural language query, where the model can benefit from interactions between the utterances and the schema.

4.1.2 Transition Layer

The transition layer is used to guide the information flow among the different components properly. The output of the transition layer is the representation of the target language utterance and the database schema. Formally, the transition layer is denoted as follows:

$$H_t^{m+1}, H_d^{m+1} = f(H_s^m, H_t^m, H_d^m). \tag{4}$$

We discuss different transition layer designs in detail in Section §4.2.

4.1.3 Target-centric Encoder

Only the hidden states of the target language utterance and the schema are kept for further modeling, eliminating the side effects of noisy translations. Formally, k-layer transformers are applied to the concatenation of target language utterance and schema representations:

$$H_t, H_d = \texttt{Transformers}([H_t^{m+1}; H_d^{m+1}]) \tag{5}$$

The output of the target-centric encoder is then used in the SQL decoder.

4.1.4 SQL Decoder

The transformer decoder is trained to generate SQL queries token by token. The SQL queries are directly tokenized without any preprocessing. The cross-attention of the transformer decoder is applied to the output of the target-centric encoder. Compared to a grammar-based SQL decoder, SQL queries generated token by token may have syntactic errors. For example, the SQL generator may hallucinate column names that are not from the corresponding database schema. To alleviate this issue, we apply the constrained decoding algorithm Picard (Scholak et al., 2021) to improve the SQL generation quality.

4.2 Design of the Transition Layer

Here we introduce the transition layer in detail. The transition layer enhances interactions among the different components (source language utterance, target language utterance, and database schema). The transition layer serves as an information mixer and information flow controller, by fusing information from different components implicitly or explicitly and feeding it to the next layer. The output of the transition layer is the hidden representation of the utterance in the target language and the schema, ignoring the source language information. In this way, the source language utterance only serves as context for the target utterance and/or the schema, without interfering with the decoder behavior explicitly due to unexpected translation noise. Here, we discuss three different transition mechanisms, namely implicit full mixup, implicit utterance mixup, and explicit utterance mixup.

Implicit Full Mixup (IFM): For implicit full mixup, all three components are involved in the modeling. The implicit full mixup layer is parameterized with a single layer transformer:

$$\begin{aligned} &H_t^{m+1}, H_d^{m+1} \\ &= \texttt{Transformer}([H_s^m; H_t^m; H_d^m])[p:q], \end{aligned} \tag{6}$$

where $[p : q]$ is the span of the concatenated sequence of target language utterance and schema tokens. Note that the hidden states of the source language utterance only serve as keys and values, while the hidden states of the target language utterance and the schema serve as queries, keys, and values in the multi-head attention. This is different from a vanilla transformer layer.

Implicit Utterance Mixup (IUM): The implicit utterance mixup implements the information flow transition on the utterance part:

$$\begin{aligned} H_t^{m+1} &= \texttt{Transformer}([H_s^m; H_t^m])[p:q] \\ H_d^{m+1} &= H_d^m, \end{aligned} \tag{7}$$

where $[p : q]$ is the span of the target language utterance. For the schema representation, skip connections are applied. Similar to implicit full mixup, the hidden states of the utterance from the source language are specifically used as keys and values, while the target utterance hidden states are also used for queries in multi-head attention. The main goal is to enhance the representation of the target language utterance by integrating information from the source language counterparts. This can further reduce the cross-lingual representation discrepancy between the target language utterance and the source language (English) schema.

Explicit Utterance Mixup (EUM): Instead of using fully connected self-attention to learn representation mixup, explicitly controlling the information flow is another strategy. Formally, self-attention is applied independently to the source language utterance and the target language utterance:

$$\begin{aligned} H_s^{'} &= \texttt{MultiHead}(H_s^m, H_s^m, H_s^m) \\ H_t^{'} &= \texttt{MultiHead}(H_t^m, H_t^m, H_t^m). \end{aligned} \tag{8}$$

Manifold mixup (Yang et al., 2021) provides a way to obtain intermediate representations by conducting linear interpolation on the hidden states, leveraging a cross-attention layer:

$$H_{t|s}^{m+1} = \texttt{MultiHead}(H_t^{'}, h_s^{'}, h_s^{'}). \tag{9}$$

Following Yang et al. (2021), the cross-attention layer shares parameters with the self-attention layer. With the cross-attention layers, by using the hidden states of the target language tokens as queries and the hidden states of the source language tokens as keys and values, the model can extract target-related signals from the source. With the interpolation operation controlled by a mixup ratio λ, the target representation can be enhanced:

$$H_t^{m+1} = \texttt{LayerNorm}(\lambda H_{t|s}^{m+1} + (1-\lambda)H_t^{m+1}), \tag{10}$$

where the mixup ratio λ is shared by each example in training and inference. Similar to Equation 7, a skip connection layer is applied to the schema representations.

5 Framework Configurations

By configuring parameters such as m and k, some basic model architectures can be obtained.

Multi-source Input with Concatenation: Concatenation is a simple yet effective baseline for leveraging both the source language utterance and the target language utterance at the same time. By setting $k = 0$ and `Transition Layer` = `None`, our framework is configured as a simple concatenation model. In this case, the decoder can leverage bilingual information to generate the SQL query.

Focused Simple Concatenation: By setting $k = 0$ and `Transition Layer` = `Implicit Full Mixup`, our REX framework can obtain a new architecture, denoted as *focused* simple concatenation. Different from simple concatenation, the SQL decoder only focuses on the target utterance component and the database schema, ignoring the source language utterance. This can reduce the negative effects caused by noisy machine translations.

6 Experimental Setup

Datasets: In this work, we evaluate our framework on two Text-to-SQL semantic parsing datasets, CSPIDER (Min et al., 2019) and VSPIDER (Tuan Nguyen et al., 2020), which are **C**hinese and **V**ietnamese cross-domain Text-to-SQL datasets adapted from the SPIDER benchmark (Yu et al., 2018).

For CSPIDER, we use **E**xact Set **M**atch (EM) accuracy as the evaluation metrics. For VSPIDER, we use both EM accuracy and **T**est-**s**uite (TS) (Zhong et al., 2020) accuracy for evaluation. For Exact Set Match accuracy, the prediction is classified as correct only if all of the components (`SELECT` clause, `WHERE` clause, `HAVING` clause, etc.) are correct. The Test-suite accuracy, which is an improved version of execution accuracy (if the execution results of a predicted SQL query are the same as those of the ground truth SQL query, then it is classified as correct), serves as a tight upper bound for semantic accuracy (Zhong et al., 2020).

Model Training: Our REX framework is an adapted sequence-to-sequence transformer. Benefiting from pre-trained sequence-to-sequence language models such as BART (Lewis et al., 2020) or T5 (Raffel et al., 2020), performance is significantly improved by finetuning pre-trained models. Furthermore, because our REX framework is expected to process utterances in multiple languages, our experiments are based on mT5-large, which has 24 layers.

To leverage English annotated data, we conduct two-stage training: We first train mT5-large on the English dataset to obtain a trained parser checkpoint. Note that this trained English parser is based on a standard sequence-to-sequence architecture instead of the REX framework. The input of the parser is the concatenation of the English utterance

Model	**Dev.**	**Test**
DG-SQL (Wang et al. (2021))	50.4	46.9
XL-SQL	54.9	47.8
RAT-SQL + GraPPa + Adv	59.7	56.2
LGESQL + ELECTRA + QT	64.5	58.1
Single-source		
Target-language Training	63.7	-
Multi-source		
Concatenation	65.5	-
Rex	**66.1**	**59.7**

Table 1: Model performance on the CSpider development set and hidden test set with EM accuracy.

Model	**EM**	**TS**
Single-source		
Target-language Training	64.2	59.0
Multi-source		
Concatenation	65.6	61.9
Rex	**69.0**	**64.5**

Table 2: Model performance on the VSpider development set with EM accuracy and TS accuracy.

and the linearized database schema. This model obtains state-of-the-art performance on the Spider benchmark. We use the checkpoint to initialize our Rex framework and further finetune on target language datasets. During inference, we translate the target language utterances into English as model inputs. To fix the number of model parameters, we always use $m+k+1 = 24$ in our experiments. For hyper-parameters, we follow Scholak et al. (2021) in all our experiments. By default, we use Google Translate for all model training and inference.

7 Main Results

CSpider: We report the performance of Rex on CSpider, which can be compared with other state-of-the-art systems on the leaderboard.[3] As shown in Table 1, our Rex framework obtains 66.1% EM accuracy on the development set and 59.7% EM accuracy on the hidden test set, exceeding the best-performing system, LGESQL+ELECTRA+QT, by 1.6% on both the development set and the test set. Our model is based on a joint general encoder, explicit utterance mixup with 0.1 mixup ratio, and setting $m = 16$ and $k = 7$. Our system achieves state-of-the-art performance on the CSpider benchmark at the time of writing.

Model	**zh**			**vi**		
	G.	M.	Δ	G.	M.	Δ
Concat.	65.5	61.4	4.1	65.6	63.0	2.6
Rex	66.1	65.0	1.1	69.0	66.8	2.2

Table 3: Robustness with respect to translation error. The performance comparison is conducted based on parsers that use Google Translate (G.) and parsers that use Marian Translate (M.). Concat. denotes multi-source input with the concatenation model. Marian Translate introduces more noise than Google Translate. EM accuracy is reported.

Comparing the multi-source models to single-source ones, we observe that the extra information can improve parser effectiveness. With multi-source concatenation, the parser is already competitive with the state-of-the-art parser. Our Rex framework further improves the model by 0.6% over the multi-source input with concatenation.

VSpider: Because our data split on VSpider is different from Tuan Nguyen et al. (2020), our results are not directly comparable.[4]

The main results for VSpider are shown in Table 2. As a single source baseline, target language training obtains 64.2% EM accuracy and 59.0% TS accuracy. Multi-source concatenation outperforms target-language training, with 1.4% improvement on EM accuracy and 2.9% improvement on TS accuracy. Our Rex framework achieves better effectiveness both on EM accuracy and TS accuracy, with 69.0% EM accuracy and 64.5% TS accuracy. The model is based on a joint general encoder, explicit utterance mixup with 0.3 mixup ratio, and setting $m = 12$ and $k = 11$.

8 Discussion

To investigate issues in cross-lingual semantic parsing and to better understand our Rex framework, we performed several ablation experiments.

8.1 Robustness to Translation Noise

We argue that the models with transition layers are more robust to translation noise than the baseline model using multi-source input with concatenation. To verify this, we conduct an ablation study by testing the models using English translations from different translation systems: Google Trans-

[3] https://taolusi.github.io/CSpider-explorer

[4] Tuan Nguyen et al. (2020) split the dataset into training (6831), dev (954), test (1906) sets. In order to prevent data leak from the trained English parser, we keep our splits consistent with Spider: training set (8659) and dev set (1034).

Model	zh	vi
Target language Training	62.5	62.8
En → Target language Training	63.7	64.2

Table 4: Ablation study of two-stage training. “Target language training” denotes using target language labeled data to train the parser from scratch and “En→ Target language Training” denotes two-stage training. EM accuracy is the evaluation metric.

late and Marian Translate (Tiedemann and Thottingal, 2020). More specifically, the models are first trained with translations obtained from Google Translate, using the simple concatenation model and the REX architecture. These models are tested with different translation systems. The results in the G. and M. columns show the performance of models that are tested with data from Google Translate and Marian Translate, respectively. The Δ column shows the performance gap between using Google Translate and Marian Translate.

The experimental results are shown in Table 3. For Chinese, the concatenation based model is sensitive to translation noise, degrading from 65.5% to 61.4% when the translation system is switched from Google Translate to Marian Translate. However, performance of the REX model only drops 1.1% on accuracy, showing better robustness towards the translation noise. Similarly, on the VSPIDER benchmark, our REX model shows better robustness than the concatenation model or the model obtained from source-language training.

8.2 Effectiveness of Two-stage Training

Here we conduct an ablation study to show that two-stage training is an effective way to leverage annotated English data. With labeled data in the target language, we can train the parser from scratch or apply two-stage training. Results are shown in Table 4. We can observe that two-stage training can benefit the parser consistently under different settings. For example, two-stage training can improve 1.2% EM accuracy for Chinese and 1.4% for Vietnamese. These results rationalize our design choice for the REX framework.

8.3 Transition Layer Index and Design

We explore choices of transition layer index under different transition layer designs, including implicit full mixup, implicit utterance mixup, and explicit utterance mixup. We configure $m \in \{4, 8, 12, 16, 20, 23\}$. Note that when $m = 23$, the `Target-centric Encoder` = `None`. For the explicit utterance mixup, we use a fixed ratio controller by setting $\lambda = 0.3$ (see study of the mixup ratio in Section §8.4). We conduct the ablation study on the VSPIDER dataset and the experimental results are shown in Table 5.

Model	4	8	12	16	20	23
IFM	66.4	67.5	68.7	68.8	68.6	68.9
IUM	67.6	67.7	68.8	68.4	68.0	67.5
EUM	67.6	67.4	**69.0**	68.5	68.2	66.8

Table 5: Ablation study of the transition layer index on the VSPIDER dev set. IFM denotes implicit full mixup; IUM denotes implicit utterance mixup; EUM denotes explicit utterance mixup. EM accuracy is reported.

ratio	Easy	Med.	Hard	Extra	All
0.1	88.3	**73.1**	49.4	42.2	67.8
0.2	87.1	72.6	51.7	**47.0**	**68.5**
0.3	**89.1**	71.7	52.9	45.2	**68.5**
0.4	84.7	70.0	**54.6**	45.2	66.9
0.5	86.3	70.9	51.7	45.8	67.3

Table 6: Study of the mixup ratio. Experimental results are based on the VSPIDER benchmark, using EM accuracy as the evaluation metric. Accuracy on different difficulty levels are reported.

The EM accuracy scores from the configurations shown in Table 5 are better than the concatenation baseline (65.6%). For implicit full mixup, we find that the transition layer can contribute to model effectiveness more when $m \geq 12$. Note that when $m = 23$, the framework is configured as focused simple concatenation (see §5). For implicit utterance mixup, when $m = 12$ or $m = 16$, the model can perform better. Similarly, for explicit utterance mixup, implementing in the middle layers benefits the most. Especially when $m = 12$, the parser achieves the best EM accuracy. One possible explanation is that utterance mixup changes the information flow more significantly than full mixup, requiring more target-centric layers to encode the mixed information. Regarding the different settings of transition layer design, there is no clear winner. For example, implicit full mixup usually outperforms the others when $m \in \{16, 20, 23\}$. When $m \in \{4, 8\}$, implicit utterance mixup achieves higher accuracy than full mixup.

8.4 Choice of Mixup Ratio

The mixup ratio for the explicit utterance mixup is an important hyper-parameter that affects the

Model	zh	vi
Independent Encoder	64.1	66.2
Joint Encoder	66.1	69.0

Table 7: Ablation study of the general encoder. EM accuracy is reported.

information flow of English translations. Here, we conduct an analysis to see how the ratio influences parser effectiveness. The experiments are based on the supervised setting with the VSPIDER dataset, by configuring $m = 16$ and $\lambda \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$.

The experimental results are shown in Table 6. Based on the overall results, we can observe that the parser obtains the best overall performance when the mixup ratio is 0.2 or 0.3. For different difficulty levels,[5] there is no single ratio setting that achieves the best performance on the four difficulty levels. For example, 0.3 is the best ratio for easy questions while 0.4 is the best for hard questions.

8.5 Choice of General Encoder Design

We compare different design choices for the general encoder in Table 7. Even though the independent encoder has the merit of efficient inference (the hidden states of the schema can be reused for all queries to the database), the performance drop is noticeable. The independent encoder is similar to the *local transformers* proposed in the FILTER architecture (Fang et al., 2021), which benefits POS tagging and multilingual QA tasks. However, we argue that for the semantic parsing task, full interactions between different components are more beneficial.

8.6 Case Studies

Here we conduct an analysis to see what cases the REX framework can improve over the baseline and what cases it still fails.

The noise introduced by translation may affect the parser performance unexpectedly if the decoder accesses the translated English utterance representations directly, such as with the model using multi-source input with concatenation. For example 1 in Table 8, the English translation does not correctly translate 以数量升序 (“in ascending order of the count”), which causes the concatenation based model to fail to predict the `ORDER BY` clause. In contrast, our REX framework leverages the explicit utterance mixup transition layer and the target-centric encoder to ignore this translation noise and predict the SQL query correctly.

Example 1

Chinese Utterance:
请以数量升序显示管弦乐队的唱片格式。

English Translation:
Please display the orchestra record format in ascending order.

Concat. Prediction:
```
SELECT major_record_format FROM orchestra ORDER
BY major_record_format ASC
```

REX Prediction:
```
SELECT major_record_format FROM orchestra GROUP
BY major_record_format ORDER BY count(*) ASC
```

Gold:
```
SELECT major_record_format FROM orchestra GROUP
BY major_record_format ORDER BY count(*) ASC
```

Example 2

Chinese Utterance:
欧洲哪些国家至少有3家汽车制造商?

English Translation:
Which European countries have at least 3 car manufacturers?

Concat. Prediction:
```
SELECT Country FROM CAR_MAKERS GROUP BY Country
HAVING COUNT(*) >= 3
```

REX Prediction:
```
SELECT T1.CountryName FROM COUNTRIES AS T1 JOIN
CAR_MAKERS AS T2 ON T1.CountryId = T2.Country
WHERE T1.Continent = "欧洲" GROUP BY
T1.CountryName HAVING COUNT(*) >= 3
```

Gold:
```
SELECT T1.CountryName FROM COUNTRIES AS T1 JOIN
CONTINENTS AS T2 ON T1.Continent = T2.ContId
JOIN CAR_MAKERS AS T3 ON T1.CountryId =
T3.Country WHERE T2.Continent = "欧洲" GROUP BY
T1.CountryName HAVING count(*) >= 3
```

Table 8: Case studies comparing REX and the Concatenation baseline. Examples are selected from the CSPIDER dev set.

For example 2, both the baseline system and REX fail. However, comparing the REX prediction with the Concat. prediction, we see that the REX output is closer to the gold SQL query. Comparing the two, we see that the REX query fails to join `COUNTRIES` with `CONTINENTS` and uses `CONTINENTS.Continent = "欧洲"` in the `WHERE` clause instead of `COUNTRIES.Continent`, because `COUNTRIES.Continent` has the ID instead of the name. This suggests that the model can be further improved by proposing better encoding techniques for the schema information.

[5] The difficulty level of a query is based on the complexity of the corresponding SQL; see Yu et al. (2018) for more details.

9 Related Work

Cross-lingual Semantic Parsing: The goal of cross-lingual semantic parsing is to process user utterances in multiple languages and convert them into some type of logical representation. Much research progress has been made on this task in recent years.

Dataset creation represents a fundamental contribution that is useful for benchmarking progress (Bai et al., 2018; Li et al., 2021a; Cui et al., 2022; Sherborne et al., 2020; Susanto and Lu, 2017; Upadhyay et al., 2018; Xu et al., 2020). On the other hand, for model development, multilingual pre-trained models are widely applied to the task in a supervised fashion or zero-shot fashion (Sherborne et al., 2020; Sherborne and Lapata, 2022; Li et al., 2021a).

For example, Sherborne and Lapata (2022) recently focused on cross-lingual transfer, where the model trained on English data is effectively adapted to other languages. However, their work focuses on single-database semantic parsing and the trained models do not generalize well across different databases. Instead, we focus on cross-database semantic parsing under the supervised learning setting.

Representation Mixup: The term "mixup" was first introduced by Zhang et al. (2018), referring to a data-agnostic data augmentation method for reducing the memorization issue and improving model robustness. Follow-up work tried to mix up the hidden representations (Verma et al., 2019) instead of the input. This technique has been widely applied in different directions (Yang et al., 2021; Fang et al., 2022; Li et al., 2021b). Our approach is the first to introduce the idea of mixup to cross-lingual Text-to-SQL semantic parsing.

10 Conclusions

We propose the REX framework that effectively integrates information from English translations into the modeling of target language utterances. More specifically, we propose three different transition layer implementations that enhance the interactions among different components. We further compare their effectiveness with detailed ablation studies. Experiments show that our framework is robust to translation noise by controlling the information flow properly, outperforming existing baselines on the VSPIDER and CSPIDER benchmarks.

11 Limitations

In this work, our model uses 24-layer transformers, with 1.2B parameters in total. To train such a model requires 8 V100 GPUs for around 24 hours. Thus, this work requires substantial GPU resources. Due to limited data resources, we can only test our methods in Chinese and Vietnamese. In the future, we plan to extend the framework to more language families by creating additional multilingual annotated datasets.

Acknowledgements

This research was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada, Compute Ontario, and Compute Canada.

References

He Bai, Yu Zhou, Jiajun Zhang, Liang Zhao, Mei-Yuh Hwang, and Chengqing Zong. 2018. Source critical reinforcement learning for transferring spoken language understanding to a new language. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3597–3607, Santa Fe, New Mexico, USA.

DongHyun Choi, Myeong Cheol Shin, EungGyun Kim, and Dong Ryeol Shin. 2021. RYANSQL: Recursively applying sketch-based slot fillings for complex text-to-SQL in cross-domain databases. *Computational Linguistics*, 47(2):309–332.

Ruixiang Cui, Rahul Aralikatte, Heather Lent, and Daniel Hershcovich. 2022. Compositional Generalization in Multilingual Semantic Parsing over Wikidata. *Transactions of the Association for Computational Linguistics*, 10:937–955.

Xiang Deng, Ahmed Hassan Awadallah, Christopher Meek, Oleksandr Polozov, Huan Sun, and Matthew Richardson. 2021. Structure-grounded pretraining for text-to-SQL. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1337–1350, Online.

Qingkai Fang, Rong Ye, Lei Li, Yang Feng, and Mingxuan Wang. 2022. STEMM: Self-learning with speech-text manifold mixup for speech translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7050–7062, Dublin, Ireland.

Yuwei Fang, Shuohang Wang, Zhe Gan, Siqi Sun, and Jingjing Liu. 2021. Filter: An enhanced fusion method for cross-lingual language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12776–12784.

Jiaqi Guo, Zecheng Zhan, Yan Gao, Yan Xiao, Jian-Guang Lou, Ting Liu, and Dongmei Zhang. 2019. Towards complex text-to-SQL in cross-domain database with intermediate representation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4524–4535, Florence, Italy.

Binyuan Hui, Ruiying Geng, Lihan Wang, Bowen Qin, Yanyang Li, Bowen Li, Jian Sun, and Yongbin Li. 2022. S^{2}SQL: Injecting syntax to question-schema interaction graph encoder for text-to-SQL parsers. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1254–1262, Dublin, Ireland.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online.

Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. 2021a. MTOP: A comprehensive multilingual task-oriented semantic parsing benchmark. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2950–2962, Online.

Jicheng Li, Pengzhi Gao, Xuanfu Wu, Yang Feng, Zhongjun He, Hua Wu, and Haifeng Wang. 2021b. Mixup decoding for diverse machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 312–320, Punta Cana, Dominican Republic.

Percy Liang. 2013. Lambda dependency-based compositional semantics. *arXiv preprint arXiv:1309.4408*.

Xi Victoria Lin, Richard Socher, and Caiming Xiong. 2020. Bridging textual and tabular data for cross-domain text-to-SQL semantic parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4870–4888, Online.

Qingkai Min, Yuefeng Shi, and Yue Zhang. 2019. A pilot study for Chinese SQL semantic parsing. 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)*, pages 3652–3658, Hong Kong, China.

Jiexing Qi, Jingyao Tang, Ziwei He, Xiangpeng Wan, Chenghu Zhou, Xinbing Wang, Quanshi Zhang, and Zhouhan Lin. 2022. RASAT: Integrating relational structures into pretrained seq2seq model for text-to-SQL. *arXiv preprint arXiv:2205.06983*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Ohad Rubin and Jonathan Berant. 2021. SmBoP: Semi-autoregressive bottom-up semantic parsing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 311–324, Online.

Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. 2021. PICARD: Parsing incrementally for constrained auto-regressive decoding from language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9895–9901, Online and Punta Cana, Dominican Republic.

Tom Sherborne and Mirella Lapata. 2022. Zero-shot cross-lingual semantic parsing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4134–4153, Dublin, Ireland. Association for Computational Linguistics.

Tom Sherborne, Yumo Xu, and Mirella Lapata. 2020. Bootstrapping a crosslingual semantic parser. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 499–517, Online.

Peng Shi, Patrick Ng, Zhiguo Wang, Henghui Zhu, Alexander Hanbo Li, Jun Wang, Cicero Nogueira dos Santos, and Bing Xiang. 2021a. Learning contextual representations for semantic parsing with generation-augmented pre-training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 13806–13814.

Peng Shi, Tao Yu, Patrick Ng, and Zhiguo Wang. 2021b. End-to-end cross-domain text-to-SQL semantic parsing with auxiliary task. *arXiv preprint arXiv:2106.09588*.

Raymond Hendy Susanto and Wei Lu. 2017. Neural architectures for multilingual semantic parsing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 38–44, Vancouver, Canada.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT – building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480, Lisboa, Portugal.

Anh Tuan Nguyen, Mai Hoang Dao, and Dat Quoc Nguyen. 2020. A pilot study of text-to-SQL semantic parsing for Vietnamese. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4079–4085, Online.

Shyam Upadhyay, Manaal Faruqui, Gokhan Tür, Hakkani-Tür Dilek, and Larry Heck. 2018. (Almost) zero-shot cross-lingual spoken language understanding. In *2018 IEEE International Conference on*

Acoustics, Speech and Signal Processing (ICASSP), pages 6034–6038.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. 2019. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pages 6438–6447.

Bailin Wang, Mirella Lapata, and Ivan Titov. 2021. Meta-learning for domain generalization in semantic parsing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 366–379, Online.

Bailin Wang, Richard Shin, Xiaodong Liu, Oleksandr Polozov, and Matthew Richardson. 2020. RAT-SQL: Relation-aware schema encoding and linking for text-to-SQL parsers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7567–7578, Online.

Bailin Wang, Ivan Titov, and Mirella Lapata. 2019. Learning semantic parsers from denotations with latent structured alignments and abstract programs. 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)*, pages 3774–3785, Hong Kong, China.

Weijia Xu, Batool Haider, and Saab Mansour. 2020. End-to-end slot alignment and recognition for cross-lingual NLU. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5052–5063, Online.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online.

Huiyun Yang, Huadong Chen, Hao Zhou, and Lei Li. 2021. Enhancing cross-lingual transfer by manifold mixup. In *International Conference on Learning Representations*.

Pengcheng Yin and Graham Neubig. 2017. A syntactic neural model for general-purpose code generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 440–450, Vancouver, Canada.

Pengcheng Yin, Chunting Zhou, Junxian He, and Graham Neubig. 2018. StructVAE: Tree-structured latent variable models for semi-supervised semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia.

Tao Yu, Chien-Sheng Wu, Xi Victoria Lin, Bailin Wang, Yi Chern Tan, Xinyi Yang, Dragomir Radev, Richard Socher, and Caiming Xiong. 2021. Grappa: grammar-augmented pre-training for table semantic parsing. In *International Conference on Learning Representations*.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921, Brussels, Belgium.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.

Ruiqi Zhong, Tao Yu, and Dan Klein. 2020. Semantic evaluation for text-to-SQL with distilled test suites. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 396–411, Online.

Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2sql: Generating structured queries from natural language using reinforcement learning. *arXiv preprint arXiv:1709.00103*.