Optimizing Text-to-SQL Models for Chinese Language Translation: A Cross-Evaluation of Single-Language and Multilingual Models

Yizhen Zhang, Dongbing Han, Tao Yan
yz4401@columbia.edu
dh3071@columbia.edu
ty2481@columbia.edu
Columbia University

ABSTRACT

Text-to-SQL converts natural language to structured SQL queries for retrieving desired information, enabling effective communication between users and databases. This is critical for modern database systems as users demand more natural and intuitive access. As regional integration increasingly serves as a development strategy globally, the demand for Text-to-SQL to support multiple languages has grown. However, current Text-to-SQL models are typically trained solely on English or on multiple languages but exhibit lower accuracy when applied to languages other than English. In this research paper, our goal is to optimize existing English models to achieve better performance on the Chinese language, which has the second-largest number of users worldwide. Specifically, we aim to optimize existing Seq2Seq, T5, and mBart-50 models to enhance their performance in Chinese translation tasks. Furthermore, we intend to conduct cross-evaluations to assess the efficiency of both single-language and multilingual models for translating the Chinese language. For further detail: https://github.com/hando189890/COMS6113_Database_Research

KEYWORDS

Adapting Text-to-SQL Models for Non-English Languages: A Cross-Evaluation of Single-Language and Multilingual Models

ACM Reference Format:
Yizhen Zhang, Dongbing Han, Tao Yan. 2023. Optimizing Text-to-SQL Models for Chinese Language Translation: A Cross-Evaluation of Single-Language and Multilingual Models. In *Proceedings of ACM Conference (Conference'17)*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Conference'17, July 2017, Washington, DC, USA
© 2023 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In modern database systems, natural language queries have gained increasing popularity due to the growing need for users to access databases in a more intuitive way. Text-to-SQL is a research area that involves converting natural language queries into structured SQL database queries that can be executed to retrieve the desired information.[19] Such research is critical as it enables effective communication between users and databases.

Despite recent progress in developing Text-to-SQL models, most existing models have two main drawbacks. Firstly, most of the developed models such as T5 and Seq2Seq focus on English language queries only and perform poorly when substituting the dataset into non-English language. [8] Secondly, the large-scale multi-lingual model, mBART-50, has a skewed performance with high accuracy in English but relatively low accuracy in other languages. Research such as "mRAT-SQL+GAP: A Portuguese Text-to-SQL Transformer" has been done to improve mBART-50's performance in the Portuguese language, but none has been done to improve that of Chinese. [6]

To bridge the aforementioned gap in Text-to-SQL models, we undertook several adaptations to improve their performance on the Chinese language. To establish a comparison baseline in the English WikiSQL dataset, we adapted three models, namely Seq2Seq, T5, and mBart-50. Subsequently, we incorporated the T5 and mBart-50 models to support the Chinese WikiSQL dataset. We employed transfer learning techniques to pretrain the models on a large-scale Chinese NLP dataset, and then fine-tuned them on the Chinese WikiSQL dataset. Additionally, we utilized the mBART model trained on a multilingual WikiSQL dataset that we created. During our experimentation, we aimed to evaluate whether training the models using both the original and translated training datasets together, even if the target language was solely Chinese, would result in better performance.

The rest of paper is structured as follows. Section 2 presents a summary of related research and state-of-the-art developments in the field of Seq-2-Seq, T5, and multilingual models for English, Chinese, and other languages. Section 3 provides an overview our Chinese WikiSQL dataset, which we developed by translating the original English queries from the WikiSQL dataset. In Section 4, we provide technical details on how we adapted existing models and optimized mBART-50. Section 5 presents the results of our experiments, including a cross-evaluation of single language models, such as Seq-to-Seq and T5, and multilingual models, such as mBART-50.

Finally, in Section 6, we summarize our research results and discuss their potential impact.

2 RELATED WORK

This section discusses related work in three areas: Seq-to-Seq models, T5 models, and multi-lingual models. The Seq-to-Seq models rely on RNNs or transformer-based architectures as decoders and employ attention and copy mechanisms to improve accuracy. [19] T5 models use the pre-trained T5 model, fine-tuned on large-scale text-to-SQL datasets, and have achieved state-of-the-art results on WikiSQL datasets.[8] Multi-lingual models employ a shared encoder and decoder for both languages and cross-lingual attention mechanisms. [6]

2.1 Seq-to-Seq Model

Seq-to-Seq models have succeeded in the text-to-SQL task. However, their generalization ability to unseen data was limited. Therefore, researchers have proposed the use of auxiliary tasks that can serve as supportive models and regularization terms for the generation task. In 2020, Chang proposed a simple yet effective auxiliary task that improved the generalization ability of seq-to-seq models on WikiSQL, demonstrating superior generalizability compared to a strong baseline model. [2]

Another challenge for text-to-SQL models is time normalization and dealing with relatively small training samples. A master's thesis by Costa-Jussà and González Bermúdez in 2021 explored the use of sequences with normalized time information for the seq-to-seq translation task. This research aimed to improve models' ability to translate natural language questions into SQL queries that require time normalization. For example, "What were the average sales for the last three Christmases?"

Additionally, the use of Seq-to-Seq models in text-to-SQL systems has led to many recent advances in deep neural networks. In a 2021 tutorial by Katsogiannis-Meimarakis and Koutrika, the authors discussed the creation of two large datasets specifically designed for training text-to-SQL systems and the state-of-the-art techniques used for natural language representation in neural networks. The tutorial also highlighted recent text-to-SQL systems that utilize deep learning techniques and discussed open problems and research opportunities in this field.[7]

Furthermore, in 2022, Wei, Huang, and Li proposed information sharing and reweight loss techniques to enhance text-to-SQL models' performance. These techniques were tested on the WikiSQL dataset and demonstrated improved accuracy compared to state-of-the-art models. This work shows the potential for incorporating novel techniques into Seq-to-Seq models to further improve their performance in text-to-SQL systems.[15] Overall, Seq-to-Seq models have made significant strides in recent years, opening up new possibilities for natural language processing in various languages and domains.

2.2 T5 Model

The T5 model, a pre-trained text-to-text transformer model, is also used for text-to-SQL parsing. In 2019, Colin Raffel and his team published "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer." In the paper, the authors introduced an innovative pre-trained text-to-text transformer model called T5, which achieved exceptional performance on multiple natural language processing tasks, including machine translation, summarization, and question answering. The T5 model was pre-trained on a massive amount of text data and fine-tuned for specific downstream tasks, demonstrating its ability to generalize well to new tasks.[11]

In 2022, Lu Zeng and colleagues proposed a novel approach to improving T5-based text-to-SQL systems by implementing a reranking model that selects the most suitable hypothesis from a list of 10 hypotheses generated by the T5 model. They also proposed a query plan generation model and a heuristic schema linking algorithm. The combination of these approaches with T5-Large led to new breakthroughs in text-to-SQL parsing on the Spider dataset.[18]

In 2023, Jinyang Li's group proposed an updated model, GRAPHIX-T5, which augments the T5 model with specialized components for text-to-SQL parsing. These components introduce structural inductive bias into text-to-SQL parsers, improving the model's capacity for multi-hop reasoning, which is critical for generating structure-rich SQL queries. GRAPHIX-T5 achieved outstanding results across four text-to-SQL benchmarks, surpassing all other T5-based parsers significantly. [8]

Overall, the T5 model has evolved over time, with researchers proposing various approaches to improve its text-to-SQL parsing performance. These approaches include reranking models and specialized components for multi-hop reasoning, which have led to new state-of-the-art performance on text-to-SQL benchmarks.

2.3 Multi-language Model

Recent progress in text-to-SQL semantic parsing has led to the development of more advanced models that handle multilingual challenges. One example is the work done in "mRAT-SQL+GAP: A Portuguese Text-to-SQL Transformer", where the researchers showed the effectiveness of adapting state-of-the-art techniques to translate text-to-SQL to Portuguese. By using multilingual models and training them with both original and translated datasets, the researchers achieved improved accuracy even in single-language tasks. [6]

In the following year, Peng Shi, Rui Zhang, He Bai, and Jimmy Lin presented "XRICL: Cross-lingual Retrieval-Augmented In-Context Learning for Cross-lingual Text-to-SQL Semantic Parsing". The study centered on leveraging English datasets as a foundation for cross-lingual text-to-SQL semantic parsing. The team introduced the XRICL framework, which employs global translation exemplars to aid large language models in translation. This results in superior performance over existing methods and high accuracy scores for various languages. [12]

In the same year, "MultiSpider: Towards Benchmarking Multilingual Text-to-SQL Semantic Parsing" tackled text-to-SQL semantic parsing challenges in multiple languages by developing the largest multilingual text-to-SQL dataset and proposing a schema augmentation framework, SAVe. The latter significantly improved overall model performance and closed the performance gap across languages. [4] Additionally, "Makadi: A Large-Scale Human-Labeled

Dataset for Hindi Semantic Parsing" presented a large-scale, cross-lingual, cross-domain dataset for semantic parsing in Hindi. The researchers highlighted the asymmetry in progress in NLIDB solutions, which is inherently language-dependent due to diverse semantic markers across languages. This study emphasizes the importance of developing techniques and resources for low-resource languages.

Finally, the paper "mBART: Multidimensional Monotone BART" introduces mBART, a constrained version of the Bayesian Additive Regression Trees model. Unlike the unconstrained BART model, mBART incorporates monotonicity constraints on a subset of predictors using a multivariate basis of monotone trees. This results in smoother and more interpretable function estimates, better predictive performance, and less post-data uncertainty. Although many aspects of the unconstrained BART model carry over to mBART, the introduction of monotonicity constraints requires a fundamental rethinking of how the model is implemented. The paper provides simulated and real examples to showcase mBART's broad potential. [1]

To sum up, research on multilingual text-to-SQL semantic parsing has progressed quickly in recent years, with each paper building upon the previous work. These studies highlight the importance of multi-language models, cross-lingual retrieval, and language-specific schema augmentation frameworks for improving accuracy in multiple languages. Large-scale, cross-lingual datasets have been pivotal in advancing the field, especially in low-resource languages. This research indicates that multilingual text-to-SQL semantic parsing is a promising area for further investigation, with substantial potential for practical applications.

2.4 Chinese Text-to-SQL

The aforementioned developments in text-to-SQL models have been extended to the Chinese language, resulting in the development of Chinese text-to-SQL models. In the field of Chinese text-to-SQL, the initial models were based on the sequence-to-sequence approach, which used an encoder-decoder architecture to map natural language questions to SQL queries. However, researchers soon realized that these models suffered from several limitations, such as generating incomplete or incorrect SQL queries. To overcome these limitations, the research shifted towards using the sequence-to-tree approach, which generates SQL queries in a syntactic tree form, with nodes representing SQL keywords and table columns. This approach has shown promising results and has become the current state-of-the-art in Chinese text-to-SQL modeling.

2.4.1 Seq-to-Seq Model for Chinese Text-to-SQL

Text-to-SQL is a natural language processing (NLP) task that involves translating natural language queries to Structured Query Language (SQL) queries. The development of this task has been advancing rapidly in recent years, with the introduction of new datasets and models that achieve state-of-the-art performance. One of the earliest and most influential works in text-to-SQL is the seq2seq model proposed by Dong and Lapata. This approach aims to transduce a natural language question into a logical form, which is then executed as an SQL query.[3]

There have been several studies that have explored the use of seq2seq models in Chinese NLP to SQL tasks. In a study by Wang's team, the authors proposed a seq2seq model that used execution-guided decoding to generate SQL queries from Chinese natural language questions, which improved the accuracy of the generated SQL queries. The model incorporates a novel execution-guided decoding approach, where the intermediate SQL queries generated during the decoding process are executed to obtain feedback on their correctness, which is then used to guide the generation of subsequent SQL queries.[13] In another study by Guo's group, a Chinese seq2seq model was proposed that incorporated BERT pretraining to enhance the quality of the generated SQL queries. By leveraging the pretraining capabilities of BERT, the model was able to capture more complex linguistic features in the input natural language questions, leading to improved performance in generating accurate SQL queries.[5]

2.4.1 Seq-to-Tree Model for Chinese Text-to-SQL

The seq-to-tree model is a recent development in text-to-SQL technology that addresses the challenge of accurately parsing complex SQL queries with many schema items and logic operators, which can be difficult for traditional seq2seq models. Unlike seq2seq models that translate natural language statements into a linear sequence of tokens, seq-to-tree models generate an abstract syntax tree (AST) that captures the structural properties of the SQL query. This approach makes it easier to handle complex queries with many schema items and logic operators, particularly in the context of Chinese. [14]

Yin and Neubig introduced a neural model that translates natural language statements into an AST, while Rabinovich's group proposed abstract syntax networks that use recursive modules for decoding. However, these models require predicting many non-terminal rules before predicting the terminal tokens, involving more steps [10, 16]. Tao's team proposed a novel seq-to-tree model that exploits a SQL-specific grammar instead of an AST, enabling direct prediction of SQL tokens and reducing the number of steps required for prediction [17].

In a recent pilot study, Min's team proposed a seq-to-set model that leverages the entire output history as a feature for deciding the next term, outperforming previous models. The model uses different sequence-to-set modules to avoid the "ordering issue," providing each module with important dependence information by passes pre-order traverse of SQL decoding history. [9].

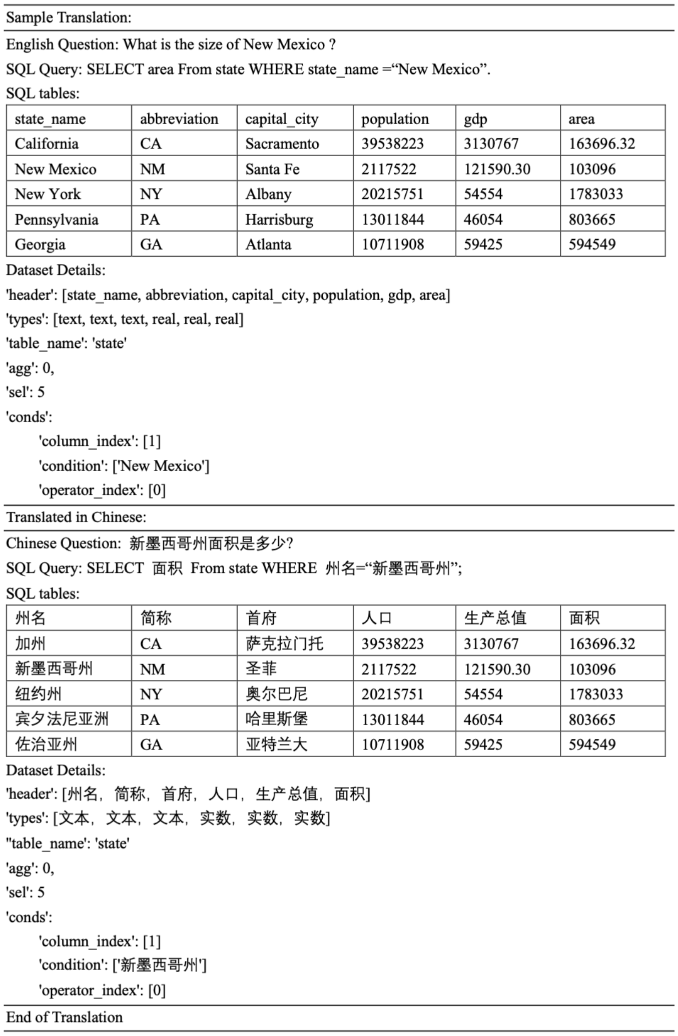
Overall, the development of text-to-SQL models has seen significant progress in recent years. The seq2seq model and WikiSQL dataset have paved the way for more advanced models, such as the seq-to-tree model, and further improvements continue to be made in the field.

3 DATASET TRANSLATION

As we experiment with multilingual environment text to SQL task, specifically in Chinese, the dataset must be translated, and the code must be modified to read, at the very least, in order to adapt to languages other than English.

3.1 Translating the WikiSQL Dataset

We construct our Chinese Dataset on top of WikiSQL, which is a large dataset that was developed through crowdsourcing for the purpose of creating natural language interfaces for relational databases in English. WikiSQL is a dataset that contains 80654 instances of questions and SQL queries that have been hand-annotated and are distributed across 24241 tables from Wikipedia. [19]



Sample Translation:

English Question: What is the size of New Mexico ?

SQL Query: SELECT area From state WHERE state_name ="New Mexico".

SQL tables:

state_name	abbreviation	capital_city	population	gdp	area
California	CA	Sacramento	39538223	3130767	163696.32
New Mexico	NM	Santa Fe	2117522	121590.30	103096
New York	NY	Albany	20215751	54554	1783033
Pennsylvania	PA	Harrisburg	13011844	46054	803665
Georgia	GA	Atlanta	10711908	59425	594549

Dataset Details:

```
'header': [state_name, abbreviation, capital_city, population, gdp, area]
'types': [text, text, text, real, real, real]
'table_name': 'state'
'agg': 0,
'sel': 5
'conds':
    'column_index': [1]
    'condition': ['New Mexico']
    'operator_index': [0]
```

Translated in Chinese:

Chinese Question: 新墨西哥州面积是多少?

SQL Query: SELECT 面积 From state WHERE 州名="新墨西哥州";

SQL tables:

州名	简称	首府	人口	生产总值	面积
加州	CA	萨克拉门托	39538223	3130767	163696.32
新墨西哥州	NM	圣菲	2117522	121590.30	103096
纽约州	NY	奥尔巴尼	20215751	54554	1783033
宾夕法尼亚洲	PA	哈里斯堡	13011844	46054	803665
佐治亚州	GA	亚特兰大	10711908	59425	594549

Dataset Details:

```
'header': [州名，简称，首府，人口，生产总值，面积]
'types': [文本，文本，文本，实数，实数，实数]
''table_name': 'state'
'agg': 0,
'sel': 5
'conds':
    'column_index': [1]
    'condition': ['新墨西哥州']
    'operator_index': [0]
```

End of Translation

Figure 1: Structure and Translation of WikiSQL.

3.2 Structure of WikiSQL dataset

The original WikiSQL dataset represents the table schema, table rows, natural language questions, corresponding SQL queries, and expected query results. The SQL queries are represented as tuples with selected columns, aggregation function, and conditions, while the query results are represented as a list of tuples with values for each column in the resulting table.[19]

Header: A list of column names for a particular table in a SQL database. The Header is used to define the structure of the table, and each column name corresponds to a column of data in the table.

Types: The types of each column in a table are represented as a list of strings, where each string corresponds to the data type of a column in the table. WikiSQL dataset only has text and real, two types.

Rows: The rows of a table are represented as a list of lists, where each inner list contains the values for each column in the corresponding row.

Question (Sel, Agg, Conds): The natural language questions in the dataset are represented as strings. Each question is associated with a SQL query, which is represented as a tuple of three elements:

sel: an integer indicating the selected column(s) in the query, and it is an int32 feature. For example, if sel is [2], it means that the query selects the third column in the table (assuming 0-based indexing).

agg: an integer indicating the aggregation function to be applied on the selected column(s), and it is also an int32 feature. The possible values of agg are: 0: no aggregation (select individual values); 1: MAX; 2: MIN; 3: COUNT; 4: SUM; 5: AVG.

conds(column_index, operator_index, condition): a list of tuples, where each tuple represents a condition in the WHERE clause of the SQL query. Each tuple contains the following elements:

column_index: an integer indicating the index of the column in the table that the condition applies to (assuming 0-based indexing).

operator_index: an integer indicating the operator used in the condition. The possible values are: 0: =; 1: >; 2: <; 3: >=; 4: <=; 5: !=; 6: LIKE.

condition: a string containing the value or pattern to be compared with the column in the condition.

3.3 Translation pipeline:

To translate the WikiSQL dataset, an online JSON translator is used to automatically translate Chinese text, which is then followed by human post-editing to ensure accuracy. The translation pipeline involves three main steps: preprocessing(edit the json file allowable to auto translator), auto-translation, and post-editing. The translation typically includes two training JSON files: train.json and train_table.json. We mostly focused on translating headers, types, rows, and condition values.

As mentioned earlier, the WikiSQL dataset only includes text and real types, and the real types are ignored during translation. Translation of dataset detail may pose challenges related to schema, lexicon, and structure. [4] The evaluation metrics not only focus on the accuracy and fluency of the question or rows, but also require that all features (header, types, agg, sel, conds) in the translated dataset have meaningful representations.

Because of the tremendous size of the WikiSQL datasets, we were only able to translate around 10000 questions with correspondence SQL queries spread across more than 150 database tables (train-set and evaluation set). Since the accuracy of the datasets is crucial for our cross-validation work, we recognize the importance of improving the accuracy of the translated datasets. However, the workload involved in post-editing is considerable, and we cannot guarantee significant improvements in accuracy. Thus, we need to explore better proposals to enhance accuracy and evaluation metrics in future works.

The translated WikiSQL dataset is advantageous compared to multiSpider or DuSQL datasets in Chinese due to its simpler schema, smaller size, more uniform data, and focus on select-project-join queries, making it a useful benchmark for evaluating natural language interfaces to databases.

3.4 Challenges During Translation

According to our exploratory research, the initial process of extracting questions and translating them using the auto translation machine resulted in a significant decrease in the quality of the Chinese dataset due to numerous ambiguities and errors. We identified the following common errors and challenges that are likely to occur when converting a text dataset to SQL, including schema and question errors.

3.4.1 The Difficulties Associated with Schema Translation

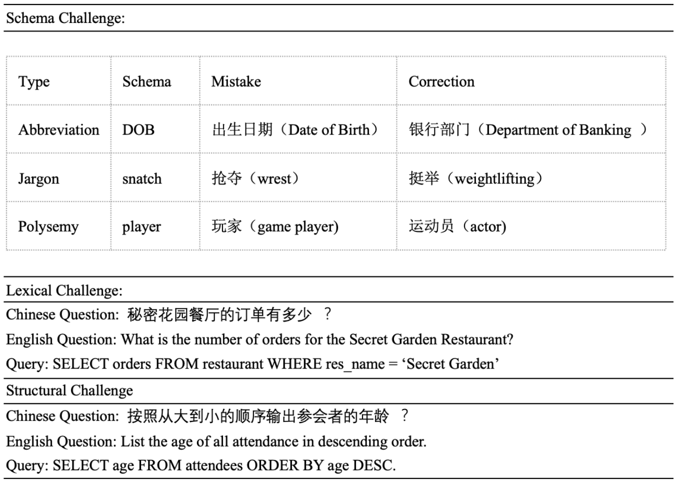
The difficulty of translating the schema is exacerbated by a combination of factors, including a lack of context and a comprehension of the domain. This challenge presents itself as abbreviations, jargon (specific to the subject), polysemy, and unique combinations of non-English place names and figure names. For example, a dataset that has a variable titled "DOB" (which stands for "department of bank"). Without any additional information, the abbreviation "DOB" could most frequently refer to the "date of birth" in auto translation. It is essential to look at the values in the column as well as the context of the dataset in order to clear up any confusion regarding the meaning of "DOB".

Furthermore, jarson and polysemy are common errors that can occur during data translation. For instance, the term "snatch" could refer to the sports of wrestling or weightlifting, alternatively. The term "player" could be translated to "玩家" meaning "player" most used in games, "演员" meaning "actor", or "运动员" meaning "athlete".

In addition, irregular words and norms can often cause significant confusion during translation process. For example, Scandinavian village names can be particularly challenging, as they often have unique spellings and pronunciations that do not follow standard rules of the language being translated. This can result in errors or inconsistencies in the translation, and may require additional research to ensure accuracy. Additionally, norms such as cultural customs or idiomatic expressions may also be difficult to translate directly, and may require adaptation or explanation in order to convey the intended meaning. Literally, when it comes to dealing with content and information that cannot be processed, we usually discard the table, as well as the questions and queries related to that particular table.

3.4.2 The Difficulties Associated with Question Translation

The first challenge in translation question is lexical, which refers to difficulties arising from the use of slang or specialized terminology. For example, consider the question "What is the number of orders for the Secret Garden Restaurant?" In this question, the term "order" is "订单", but it is often mistakenly translated as "命令" which means "command" in Chinese. The second challenge is structural, which arises when a question has complex logic or syntax. In these cases, it may be necessary to refer to the corresponding SQL query to ensure that the logic is accurately translated.



Schema Challenge:

Type	Schema	Mistake	Correction
Abbreviation	DOB	出生日期（Date of Birth）	银行部门（Department of Banking ）
Jargon	snatch	抢夺（wrest）	挺举（weightlifting）
Polysemy	player	玩家（game player)	运动员（actor)

Lexical Challenge:

Chinese Question: 秘密花园餐厅的订单有多少 ？

English Question: What is the number of orders for the Secret Garden Restaurant?

Query: SELECT orders FROM restaurant WHERE res_name = ‘Secret Garden’

Structural Challenge

Chinese Question: 按照从大到小的顺序输出参会者的年龄 ？

English Question: List the age of all attendance in descending order.

Query: SELECT age FROM attendees ORDER BY age DESC.

Figure 2: Translation Challenges in Schema and Question.

4 APPROACH OVERIVEW

In this experiment, we used a variety of language models to evaluate their performance in handling natural language to SQL translation tasks. Specifically, we utilized three different models: the single lingual language models seq2SQL and T5, as well as the multilingual language model mBart-50.

To validate our models on the English WikiSQL dataset, we used pretrained language models along with the corresponding validation dataset. This allowed us to compare the performance of our models to the state-of-the-art results on this dataset. For the Chinese WikiSQL dataset, we fine-tuned the language model and rewrite the training scripts to ensure the best possible performance. This involves reload the datasets, modify model's parameters and optimizing the training process specifically for this dataset.

4.1 Data Preprocessing

Firstly, we loaded the training data and table information from separate JSON files using the jsonlines library and creates pandas DataFrames for each. The training data DataFrame contains the phase, question, and SQL query, while the table information DataFrame contains header, types, and rows. Next, we merged the two DataFrames on the common table ID column to create a new DataFrame that includes all the necessary information for each question with the corresponding table.

We used a dictionary derived from a previous Dataframe to convert information to SQL queries. This dictionary contained column information, which we used to create two lists: aggregate and operator. The aggregate list included functions like MAX, MIN, COUNT, SUM, and AVG, while the operator list included comparison operators like "=", ">", and "<". We extracted necessary SQL query information from the Dataframe, created a dictionary, and converted it to a human-readable SQL text format. We used the aggregate and operator lists to construct the SQL text query, selecting the appropriate functions and operators based on data type and other information from the SQL query dictionary. This streamlined the process of converting Dataframe information to SQL queries.

Finally, we creates a new dataset object from a subset of the merged DataFrame, containing 10,000 randomly sampled rows. Each example in the dataset includes the question, table information, and the SQL query in both human-readable and machine-readable format. We separate these datasets 80% for trainning and 20% for evaluation.

4.2 Seq-to-SQL Model

A Seq-to-SQL model is a neural network-based model that generates SQL queries from natural language questions. It consists of an encoder and a decoder, where the encoder generates a hidden representation of the input question, and the decoder generates the corresponding SQL query using the hidden representation. [19]

Finding an appropriate model for our experiment can be a difficult and time-consuming process. In this case, we has not been able to find a suitable fine-tuned Seq2SQL model that can be used in Google Colab environmental. Additionally, the existing models do not seem suitable for the experiment at hand. As a result, we decided to take matters into their own hands and rewrite the featurization and encoder-decoder architecture from scratch. While this may be a daunting task, it allows us to customize the model to meet the specific needs of the experiment.

4.2.1 Featurization

The Encoder-Decoder Model is a popular choice for sequence-to-sequence problems, such as predicting a sequence of SQL tokens from a sequence of natural language tokens. In this model, the input sequence is first featurized using a tokenizer and concatenated with the table headers. However, to pass the data in batches, the sequence needs to be of equal length, which is achieved using padding. If the sequence is shorter than the fixed input length, zeros are added to the end. If the sequence is longer, some words are truncated from the end. The resulting sequence is then embedded into a 100-dimensional vector space using a pre-trained GloVe vector as the weight vector.[19]

4.2.2 Building Encoder-Decoder Model

The Encoder component of the model consists of multiple recurrent units that accept one input element at a time, collect information about it, and pass it on to the next unit in the stack. The Decoder component also consists of a stack of recurrent units that generate output tokens one at a time, with each unit taking in a hidden state from the previous unit and generating its own output and hidden state. As mentioned before, in the case of the seq-to-SQL problem, the input sequence is the natural language question concatenated with the table headers, and the output sequence is the corresponding SQL query. [19]

4.3 T5 Model

For our text-to-SQL task, we utilized Google's T5 model fine-tuned on WikiSQL to serve as the foundation for our evaluation work both in English and Chinese. T5 is a unified text-to-text transformer that was initially introduced in "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". [11] Transfer learning is a powerful technique in natural language processing, where a model is first pre-trained on a data-rich task before being fine-tuned on a downstream task. The T5 model takes this a step further by converting every language problem into a text-to-text format, enabling a unified framework for a diverse range of NLP tasks.[8] Our experiment utilized fine-tuned T5 model from Hugging-Face websites and explores the landscape of transfer learning techniques in text-to-SQL task by modification of dataset preparation, tokenization, fine-tuning the model.

4.4.1 Dataset preparation

To train an NLP model for text-to-SQL tasks both in English and Chinese languages, a large dataset of Chinese text and its corresponding SQL queries must be collected and prepared. The data needs to be pre-processed and cleaned to make it suitable for training the model.

To train and evaluate the Chinese text-to-SQL task, we used a dataset consisting of 10,000 rows of translated data, split into 80% for training and 20% for evaluation. Fortunately, the Chinese version of WikiSQL has been created based on the direct translation of the English version of WikiSQL, using the same format and pre-processing method. This Chinese dataset includes all the essential elements needed for the model, such as questions, tables, headers, types, rows, conditions, and the correct versions of SQL queries. Having all these elements in the dataset is essential for accurate training and evaluation of the model's performance on the Chinese text-to-SQL task. By using this dataset and appropriate tokenization methods, we can effectively train and evaluate the performance of the model for the text-to-SQL task in Chinese language.

4.4.2 Tokenization

Tokenization is an important step in preparing the data for the text-to-SQL task in both English and Chinese languages. In the T5 model, tokenization is performed using the built-in tokenizer object.

To tokenize a batch of examples, we use a function called convert_To_Features. This function takes the input and target examples and encodes them using the tokenizer object, padding them to a maximum length of 64. The encoded input and target sequences are then stored in input_encodings and target_encodings respectively.

Next, the function creates a dictionary containing the encoded input and target sequences, as well as their corresponding attention masks and labels. Attention masks are used to indicate which tokens the model should pay attention to during training or inference, and labels are the true target sequences that the model is trying to predict.

In addition to the standard input and target encodings, for the text-to-SQL task in Chinese, the tokenization process needs to take into account the differences in Chinese characters and their representation. Specifically, the tokenizer needs to tokenize Chinese text into individual characters or words using specific segmentation algorithms, such as the word-based and character-based approaches. This ensures that the Chinese text is properly encoded and can be effectively processed by the model.

4.4.3 Fine-tuning and training the model

To fine-tune the pre-trained T5 model for the Chinese text-to-SQL task, the same training procedure as for the English version is used, with some adjustments to the hyperparameters to optimize the model's performance on the Chinese dataset.

In our experiment, we set the per_device_train_batch_size and per_device_eval_batch_size parameters to 16, which determines the number of training and evaluation examples processed in parallel on each device. We also set the number of training iterations to 5 to iterate through the training dataset. The logging_steps parameter is set to 5, which controls how often output is logged during training.

The fine-tuning process involves feeding the pre-trained T5 model with the Chinese text-to-SQL dataset and adjusting the model's weights to optimize its performance on the dataset. During the training process, the model's parameters are iteratively updated using backpropagation to minimize the difference between the model's predicted SQL queries and the actual SQL queries in the dataset.

4.4 mBart-50

The mBart model is a state-of-the-art multi-lingual model that can process input text in over 50 languages, making it an ideal choice for cross-lingual text-to-SQL tasks. [1] The model is based on the BART architecture, which is a pre-trained sequence-to-sequence model that has been fine-tuned on a variety of NLP tasks. The mBART model extends BART by incorporating additional pre-training steps that enable it to learn from a diverse range of languages, allowing it to transfer knowledge across languages and improve its performance on downstream tasks.

We utilized a multi-lingual fine-tuned mBART model from Hugging Face as the foundation for our evaluation work in both English, Chinese and Mixed language text-to-SQL tasks. The fine tuned mBART model is based on the mBART-50 architecture, which is a multi-lingual extension of the BART model introduced in "Denosing Pretrained Multilingual Transformers".

4.5.1 Dataset preparation

For the evaluation of the mBART model, we followed a similar data preparation procedure as before. We used the same dataset to evaluate the model's performance for text-to-SQL tasks in English, Chinese, and mixed languages. To train the Chinese version of the mBART model, we used a dataset of 10,000 rows of translated data, which we split into 80% for training and 20% for evaluation. We applied the same pre-processing techniques as before to make the data suitable for training the model.

To create the mixed-language dataset, we inserted English data into the Chinese dataset using a specific architecture. Specifically, we used structure that even index positions for Chinese text and odd index positions for English text (assuming a zero-based indexing). This approach helped to ensure that the appropriate tokenizer was selected during training, which is essential for accurately processing and generating SQL queries in mixed-language environments.

Overall, the data preparation procedure for the mBART model evaluation was similar to that of our previous experiments, with the addition of mixing languages in the dataset to better evaluate the model's performance in mixed-language environments.

4.5.2 Tokenization

Tokenizing Chinese and English text involves different strategies due to the nature of the languages. Chinese text is typically tokenized by character, while English text is tokenized by words. This is because Chinese characters do not have spaces between them, making it challenging to segment the text into individual words. To ensure that the input sequences are properly encoded and can be effectively processed by the mBART-50 model in the text-to-SQL task, we need to use a Chinese-specific tokenizer to segment the Chinese text into individual characters. On the other hand, we can tokenize English text by splitting it into individual words based on whitespace and punctuation. In the mixed language dataset, since we have combined the Chinese and English texts by specific architecture mentioned before, we can easily distinguish them based on their index position.

By properly tokenizing the data for both Chinese and English text, we can ensure that the input sequences are properly encoded and can be effectively processed by the mBART-50 model for the text-to-SQL task, resulting in improved performance and accuracy.

4.5.3 Fine-tuning and training the model

When fine-tuning the mBART-50 model for the text-to-SQL task in Chinese and mixed languages, there are several important hyperparameters that must be considered. The per_device_train_batch_size hyperparameter determines the number of training examples per batch for each device during training. It is important to strike a balance between larger batch sizes for faster training and the increased memory requirements and potential instability that comes with it. The num_train_epochs hyperparameter specifies the number of times the entire training set is passed through the model during training. Increasing the number of epochs can improve the model's performance, but may also increase the risk of overfitting. The evaluation_strategy hyperparameter is important for monitoring the model's performance during training. Setting it to "epoch" allows evaluation to be performed after each epoch. The predict_with_generate hyperparameter is crucial for generating SQL queries based on the input text.

5 EXPERIMENTS

We have performed seven different experiments conducted on the WikiSQL dataset using three different models: Seq2SQL, T5, and mBART-50. The experiments were conducted in English and Chinese, and the models were trained and tested in the same language (En/En or Zh/Zh) or a mix of languages (Zh/En).

5.1 Setup

Figure. 3 shows the architecture of the training, inference and evaluation processes described in this section. We used Execution Accuracy (EA) to evaluate the performance of different models, Execution Accuracy is the percentage of generated SQL queries that, when executed, produce the correct answer according to the ground-truth. It measures how well a model can generate SQL queries that produce the expected results when run on the given database.

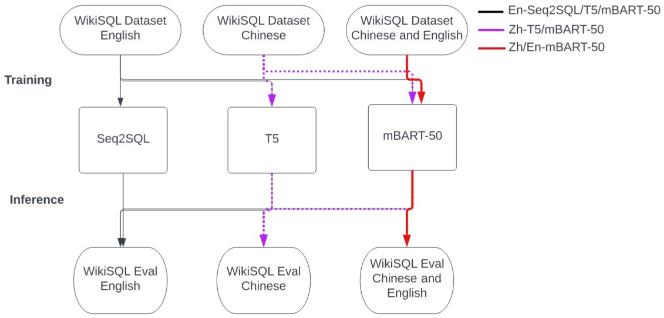
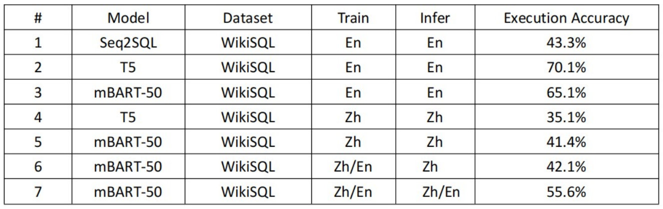


Figure 3: Architecture of the training, inference, evaluation.

5.2 Experiment and Analysis

Results can be found in Figure. 4. This table shows the results of Execution Accuracy all models. We have trained 3 three models with the WikiSQL dataset and get the following results.



#	Model	Dataset	Train	Infer	Execution Accuracy
1	Seq2SQL	WikiSQL	En	En	43.3%
2	T5	WikiSQL	En	En	70.1%
3	mBART-50	WikiSQL	En	En	65.1%
4	T5	WikiSQL	Zh	Zh	35.1%
5	mBART-50	WikiSQL	Zh	Zh	41.4%
6	mBART-50	WikiSQL	Zh/En	Zh	42.1%
7	mBART-50	WikiSQL	Zh/En	Zh/En	55.6%

Figure 4: Results

The experiments were divided into three sets, the first set of experiments are #1 to #3, the second set of experiments are #4 and #5, and the third set of experiments is #6 and #7.

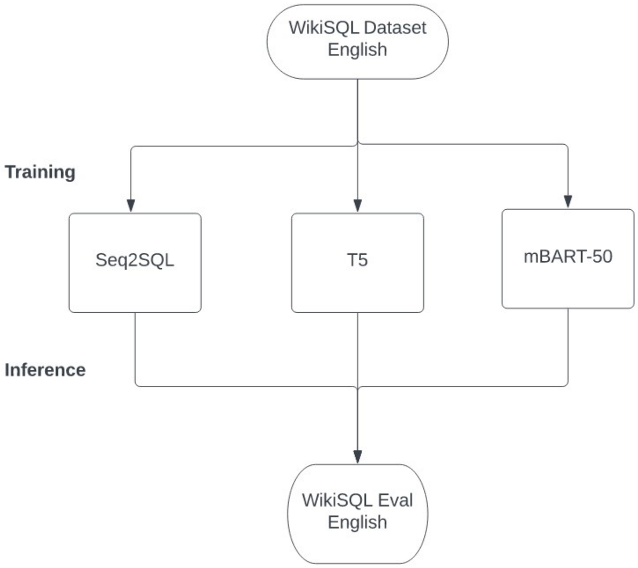


Figure 5: The procedure of the first set of experiments.

The procedure of the first experiment is shown in figure 5 and the results of the first set of experiments are as follows. The first model is Seq2SQL, which achieved an accuracy of 43.3% on the test set, and the second model is T5, which achieved a much higher accuracy of 70.1% on the test set. T5 is a more recent language model that has shown to be highly effective at a wide range of natural language processing tasks. The third model is mBART-50, which achieved an accuracy of 65.1% on the test set. mBART-50 is a multilingual version of the popular language model BART, which has been proven efficient in multilingual tasks.

Overall, for the original WikiSQL dataset, T5 is the most effective model among the three, achieving the highest accuracy on the test set. mBART-50 is specifically designed to handle multilingual tasks, while T5 was originally trained on English but can be fine-tuned for other languages as well. T5 has been trained on a diverse range of tasks and datasets, while mBART-50 has been trained on a smaller set of multilingual datasets. This may impact their performance on specific tasks, depending on the similarity between the training data and the task at hand. The performance of both T5 and mBART-50 on text-to-SQL tasks will depend on the quality of the fine-tuning dataset and the fine-tuning process itself.

Since previous experiments are specifically focused on English text-to-SQL, T5 might be a better choice, as it has been extensively pre-trained and fine-tuned on a wide range of English-language tasks, including text-to-SQL.

Our goal is to find a text to SQL model suitable for Chinese, and the next task is to repeat the above experiments using Chinese datasets and explore ways to optimize the performance. One potential approach is to train with original and translated training datasets together(English version + Chinese version), even if a single target language is desired, when training a multilingual model.

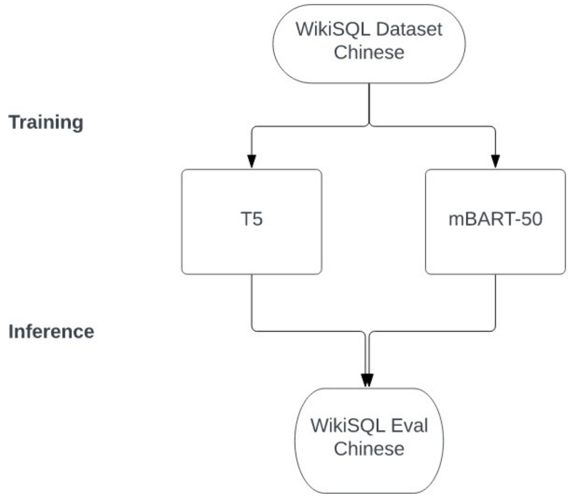


Figure 6: The procedure of the second set of experiments.

Then we conducted the second set of experiments,which focused on evaluating the performance of T5 and mBART-50 on Chinese text-to-SQL tasks. The procedure of the second set of experiment is shown in figure 6, the results showed that T5 achieved an execution accuracy of 35.1% when trained and tested on Chinese datasets, while mBART-50 achieved an accuracy of 41.4%.

Interestingly, mBART-50 outperformed T5 on the Chinese dataset, which is consistent with its design as a multilingual model. However, it's important to note that the difference in accuracy between the two models was relatively small, and both models performed significantly worse on the Chinese dataset than on the original English dataset.

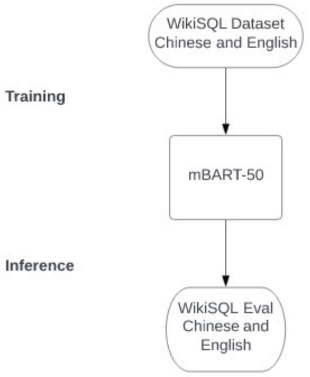

WikiSQL Dataset Chinese and English
Training
mBART-50
Inference
WikiSQL Eval Chinese and English

Figure 7: The procedure of the third set of experiments.

Finally, we conducted the third set of experiments using a mixed-language dataset, which contained both English and Chinese text-to-SQL examples. The procedure of the last set of experiment is shown in figure 7. We trained T5 and mBART-50 on this mixed-language dataset and evaluated their performance on Chinese text-to-SQL tasks. The results showed that mBART-50 achieved the highest accuracy of 55.6%, while T5 achieved an accuracy of 35.1%.

Overall, our results suggest that mBART-50 is a more suitable model for Chinese text-to-SQL tasks than T5. However, fine-tuning on a mixed-language dataset that includes both English and Chinese examples can only improve the performance of multilingual models on Chinese text-to-SQL tasks slightly. Further research is needed to optimize the performance of text-to-SQL models on multilingual tasks, such as exploring different fine-tuning approaches and training on larger multilingual datasets.

6 CONCLUSIONS

Throughout our study, we have delved into the potential benefits of incorporating a multilingual language model such as mBart-50 in enhancing the accuracy of Chinese text to SQL tasks. Moreover, we have created a comparison baseline for monolingual language models, namely Seq2Seq and T5, in both English and Chinese Wikisql datasets. This baseline will enable us to evaluate the effectiveness of using a multilingual language model like mBart-50 in comparison to these monolingual models in pure Chinese WikiSQL dataset and mixed dataset. In particular, we can analyze how mBart-50 outperforms these models in handling multilingual and cross-lingual tasks by providing better generalization across different languages. Additionally, by examining the performance of Seq2Seq and T5 in both English and Chinese Wikisql datasets, we can gain insights into how these models handle the nuances and complexities of the two languages and compare their performance with that of mBart-50. Overall, our exploration sheds light on the importance of multilingual language models in natural language processing tasks and the need for continuous improvement in this field.

REFERENCES

[1] H. A. Chipman, E. I. George, R. E. McCulloch, and T. S. Shively. mbart: Multidimensional monotone bart, 2021.
[2] N. Deng, Y. Chen, and Y. Zhang. Recent advances in text-to-SQL: A survey of what we have and what we expect. In *COLING*, Gyeongju, Republic of Korea, Oct. 2022. International Committee on Computational Linguistics.
[3] L. Dong and M. Lapata. Language to logical form with neural attention, 2016.
[4] L. Dou, Y. Gao, M. Pan, D. Wang, W. Che, D. Zhan, and J.-G. Lou. Multispider: Towards benchmarking multilingual text-to-sql semantic parsing, 2022.
[5] A. Guo, X. Zhao, and W. Ma. Er-sql: Learning enhanced representation for text-to-sql using table contents. *Neurocomput.*, 465(C):359–370, nov 2021.
[6] M. A. José and F. G. Cozman. mrat-sql+gap: A portuguese text-to-sql transformer. *CoRR*, abs/2110.03546, 2021.
[7] G. Katsogiannis-Meimarakis and G. Koutrika. A deep dive into deep learning approaches for text-to-sql systems. In *Proceedings of the 2021 International Conference on Management of Data*, SIGMOD '21, page 2846–2851, New York, NY, USA, 2021. Association for Computing Machinery.
[8] J. Li, B. Hui, R. Cheng, B. Qin, C. Ma, N. Huo, F. Huang, W. Du, L. Si, and Y. Li. Graphix-t5: Mixing pre-trained transformers with graph-aware layers for text-to-sql parsing, 2023.
[9] Q. Min, Y. Shi, and Y. Zhang. 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, Nov. 2019. Association for Computational Linguistics.
[10] M. Rabinovich, M. Stern, and D. Klein. Abstract syntax networks for code generation and semantic parsing. In *Proceedings of the 55th Annual Meeting of*

the Association for Computational Linguistics (Volume 1: Long Papers), pages 1139–1149, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[11] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2020.

[12] P. Shi, R. Zhang, H. Bai, and J. Lin. Xricl: Cross-lingual retrieval-augmented in-context learning for cross-lingual text-to-sql semantic parsing, 2022.

[13] C. Wang, K. Tatwawadi, M. Brockschmidt, P.-S. Huang, Y. Mao, O. Polozov, and R. Singh. Robust text-to-sql generation with execution-guided decoding, 2018.

[14] L. Wang, A. Zhang, K. Wu, K. Sun, Z. Li, H. Wu, M. Zhang, and H. Wang. Dusql: A large-scale and pragmatic chinese text-to-sql dataset. In *Conference on Empirical Methods in Natural Language Processing*, 2020.

[15] K. Xu, Y. Wang, Y. Wang, Z. Wen, and Y. Dong. Sead: End-to-end text-to-sql generation with schema-aware denoising, 2023.

[16] P. Yin and G. Neubig. A syntactic neural model for general-purpose code generation, 2017.

[17] T. Yu, M. Yasunaga, K. Yang, R. Zhang, D. Wang, Z. Li, and D. Radev. Syntaxsqlnet: Syntax tree networks for complex and cross-domaintext-to-sql task, 2018.

[18] L. Zeng, S. H. K. Parthasarathi, and D. Hakkani-Tur. N-best hypotheses reranking for text-to-sql systems, 2022.

[19] V. Zhong, C. Xiong, and R. Socher. Seq2sql: Generating structured queries from natural language using reinforcement learning. *CoRR*, abs/1709.00103, 2017.