mChase: A Multi-Language and Context-Dependent Text-to-SQL Dataset towards Pragmatic Semantic Parsing

Anonymous EMNLP submission

Abstract

Text-to-SQL aims to translate natural language questions into corresponding SQL queries based on a given database. The development of large general models has made Text-to-SQL research achieve considerable results, especially in English scenes, but the result of non-English scenes is not ideal, which is inconvenient for non-English users in real scenarios. To solve the Text-to-SQL problem more comprehensively and pragmatically and to maximize the ability to understand and utilize large general models, we present mChase, a multi-language and context-dependent Text-to-SQL dataset. mChase consists of 5459 coherent question sequences with annotated SQL queries, including five languages (English, Chinese, French, Japanese, and Russian), on over 280 databases. We also provide Schema Linking and context relationship annotations for each question in mChase. Additionally, we introduce an improved context-dependent evaluation metric, Equivalence Accuracy, specifically designed for mChase. Through experiments on pre-trained models and state-of-the-art methods, we gain insights into multilingual pre-training models. We hope our work can lead the Text-to-SQL research and semantic parsing research based on general models to further.

1 Introduction

Cross-database semantic parsing in context has attracted increasing attention in recent years, especially the Text-to-SQL problem(Yu et al., 2018a; Dong and Lapata, 2018; Choi et al., 2021; Zhao et al., 2022). Text-to-SQL is a task that translates the demands expressed in natural language into corresponding executable SQL queries. The study has evolved to a cross-domain, multi-table, and context-dependent scenario due to the appearance of general language models(Devlin et al., 2018; Radford et al., 2018). While Text-to-SQL is a downstream application of general models, it also addresses the Hallucination problem in database scenarios, making it an area of great research significance.

Figure 1: A question sequence from mChase dataset in five languages.

The dominance of English in Text-to-SQL problems and the construction of popular datasets like Spider(Yu et al., 2018b) and SParC(Yu et al., 2019b) in English raise questions regarding the performance of Text-to-SQL problems in multi-language scenarios. Although some Chinese and non-English datasets have appeared recently(Min et al., 2019; Sun et al., 2020; Wang et al., 2020), most research remains in English. Whether the result of Text-to-SQL will decrease in the multi-language scenario and whether English is enough to support Text-to-SQL come into question.

This paper introduces mChase, the first multi-language, context-dependent, and complex Text-to-SQL dataset, and Equivalence Accuracy, an im-

proved context-dependent evaluation metric specifically designed for mChase. mChase consists of 5459 coherent question sequences annotated with SQL queries, encompassing five languages (English, Chinese, French, Japanese, and Russian) and utilizing over 300 databases. The basic version of mChase is English, manually translated from Chase(Guo et al., 2021), while other language versions are translated manually by native speakers and experts in the respective languages.

Equivalent Accuracy is an improved context-dependent method applied to real-world scenarios that employ rearranging and inserting techniques to handle unseen databases. Then, it executed and compared the gold and predicted SQL queries to determine the consistency in reducing true-negative and false-positive samples.

To validate the effectiveness of Equivalence Accuracy and the human-translated mChase dataset, we conducted experiments on mChase using Equivalence Accuracy. Subsequently, preliminary experiments explored multi-language Text-to-SQL using language models. The comparative analysis revealed that the accuracy of the French, Japanese, and Russian versions of mChase was not ideal, highlighting the significant challenge associated with this task. Furthermore, employing a multi-language corpus enhanced model performance in language-specific scenarios. Furthermore, patterns among different general models were identified, guiding users to select the most suitable model. These findings can serve as a reference for other downstream tasks involving general models. Consequently, developing a multi-language and complex Text-to-SQL dataset becomes an urgent requirement to advance toward more realistic and general Text-to-SQL research.

The main contributions of this work can be summarized as follows:

- We propose the mChase dataset, the first large-scale multi-language and context-dependent Text-to-SQL dataset. Experiments conducted on mChase highlight the importance of having a manually created multi-language dataset.
- We propose Equivalence Accuracy, an improved context-dependent evaluation metric in Text-to-SQL. Through preliminary experiments using the metric, we uncover interesting phenomena that provide valuable insights and directions for multi-language research using general language models.

2 Related Work

This section overviews the commonly used datasets, evaluation metrics, and general methods in Text-to-SQL research.

2.1 Datasets

Spider(Yu et al., 2018b), SparC(Yu et al., 2019b), and CoSQL(Yu et al., 2019a) are three prominent English datasets with diverse applications, all of which are English corpora. Spider is a complex and cross-domain Text-to-SQL dataset, SparC is a context-dependent version of Spider, and CoSQL focuses on dialogue-based Text-to-SQL.

Efforts have been made to generate non-English Text-to-SQL datasets, mainly through machine translation(Min et al., 2019; Nguyen et al., 2020) or human translation(Bakshandaeva et al., 2022; José and Cozman, 2021) from Spider. DuSQL(Wang et al., 2020) is a Chinese single-turn Text-to-SQL dataset, and the SQL hardness inside is relatively low. Chase(Guo et al., 2021) is a Chinese dataset for cross-database context-dependent Text-to-SQL problems, with schema linking and contextual relationship annotations. Table 1 presents more detailed statistics on these datasets.

In order to improve the accuracy of approaches in Text-to-SQL, there are some auxiliary **annotations** in datasets, such as schema linking and context relationship. Schema linking is studied to research the linking between entities in questions and schemes in the database. Contextual relationship studies the relationship between questions in the input question sequences.

2.2 Evaluation Metrics

The evaluation metrics used widely in Text-to-SQL in **context-independent** scenarios are *Exact Match* and *Execution Match* proposed by Yu(Yu et al., 2018b). Exact Match measures the matching accuracy of SQL by splitting them based on keywords. Execution Match evaluates whether the predicted and gold SQL yield the same results when executed on a given database. Zhong et al. (2020) propose *test suite accuracy* to approximate semantic accuracy for Text-to-SQL models by distilling a small test suite of databases to avoid true-negative and false-positive examples using traditional metrics.

Yu et al. (2019b) proposed *Question Match* and *Interaction Match* in **context-dependent** scenarios, corresponding to the accuracy of a single question-SQL pair and a whole question-SQL sequences. In

Datasets	Language	#DB	#Pair.	Parent Dataset	HT/MT	Contextual Dependency	Annotations
Spider	English	200	8659	Original	-	✗	✗
SparC	English	200	12726	Spider	-	✓	✗
CoSQL	English	200	15598	Spider+Sparc	-	✓	✗
CSpider	Chinese	200	8659	Spider	MT	✗	✗
PAUQ	Russian	200	8659	Spider	HT	✗	✗
ViText2SQL	Vietnamese	200	8659	Spider	HT	✗	✗
PortugueseSpider	Portuguese	200	8659	Spider	MT	✗	✗
DuSQL	Chinese	120	**23,797**	Original	-	✗	✗
Chase	Chinese	**280**	17940	Sparc+Original	HT	✓	✓
mChase	English, Chinese, French, Russian, Japanese	**280**	17940	Chase	HT	✓	✓

Table 1: Statistics of mCHASE and existing popular or non-English dataset. MT stands for machine translation and HT stands for Human translation.

context-dependent research, context-independent metrics are used to evaluate Question Match, while Interaction Match considers the accuracy across all question-SQL pairs in the interaction.

2.3 Text-to-SQL Research

The commonly used Text-to-SQL model is Transformer, consisting of a joint encoder based on a large-scale pre-trained language model(Kelkar et al., 2020; Zhao et al., 2022; Deng et al., 2020) plus a grammar-based Tree-decoder(Guo et al., 2019; Wang et al., 2019). Several improvements have been made on the encoder side, including incorporating graph learning for content understanding of the database, modeling the relationship between the database and the problem, and considering historical problems Duorat(Scholak et al., 2020), LGESQL(Cao et al., 2021). Other approaches introduce autoregressive decoding models or Lora models for efficient fine-tuning (Scholak et al., 2021; Qi et al., 2022; Li et al., 2023b), and leverage the Chain-of-thought approaches for few-shot learning or instruction tuning(Pourreza and Rafiei, 2023). These methods have demonstrated impressive performance on the Spider and SparC leaderboards.

3 Construction of Datasets

To account for the variations introduced by language diversity in the Text-to-SQL field, we construct mChase, a five-language complex dataset for interactions. The construction of the mChase dataset involves three stages: (1) basic dataset construction; (2) multi-language dataset translation, and (3) data review.

3.1 Basic

After researching the existing Text-to-SQL datasets, we finally chose Chase, a large-scale context-dependent, cross-domain, and complex dataset with relatively difficult SQL, as the basic model. Thanks to Guo et al. (2021) for providing us with the original dataset for further use.

English is chosen as the benchmark language for mChase due to its status as the official language of the United Nations(Trudgill and Hannah, 2017) and its significant development in the field of NLP.

Chase consists of 5459 question sequences and 280 databases. To construct the basic mChase dataset, we translate Chase manually. Our main project team comprises two professors and four graduate students proficient in Chinese and English, focusing on SQL-related majors. The team members translated the original Chase dataset into English and made minor modifications to account for language expression differences. Inspired by the constructions of existing datasets(Min et al., 2019; Nguyen et al., 2020), we translated all the questions, SQL queries, and databases in Chase to English and other target languages.

After constructing the English version of mChase, translators are asked to annotate schema linking and contextual relationships. Following Chase(Guo et al., 2021), five basic contextual relations for question sequences are defined: Context Independent, Coreference, Ellipsis Continuation, Ellipsis Substitution, and Far side. Each question in the sequence can be categorized into at least one class. The definitions of the relationships and examples are given in Appendix A.1 and A.2.

3.2 Translation

The objective was to create versions of mChase in multiple languages, including English, Chinese, French, Japanese, and Russian. English served as the base language, and the Chase dataset was directly used for Chinese. This section outlines constructing the French, Japanese, and Russian versions of mChase from the English mChase dataset.

To ensure accurate translations, we enlisted the assistance of professionals and native speakers with expertise in the respective languages. For French, Japanese, and Russian, we engaged two computer science professors, one professor specializing in Japanese studies and one specializing in Russian studies. In addition, 22 native speakers or individuals with a native background were involved (12 for French, 5 for Japanese, and 7 for Russian).

The translators were assigned databases from different domains based on their interests, with each translator responsible for particular databases. Before commencing the translation, the translators underwent a preliminary familiarization phase, acquiring the relevant background knowledge associated with the databases. This ensured their comprehensive understanding of each database and the corresponding question sequences.

Then, translators begin to translate questions. There are three criteria: (1) Combine the contexts in the question sequences and the corresponding database; (2) Be semantic instead of literal translation; and (3) Be diverse in expressions. The translators also translated the databases, aiming to produce general, appropriate, and authentic translations encompassing all schema columns. Furthermore, the translators annotated schema linking and contextual relationship information.

3.3 Data review

The translation underwent three rounds of data review to ensure data quality. Firstly, translators received training from our main team members before beginning the translation. Each translator creates his or her first 20 question sequences under key members' supervision. Permissions for translation were granted only after confirmation of their proficiency. Secondly, the entire dataset was scrambled and distributed to translators to minimize overlap. Each translator was responsible for checking the correctness of the existing translations and executing the translated SQL to check the correctness of the answers. The crosscheck will guarantee the correctness of question sequences and annotated SQL queries. Furthermore, the professors conducted a final review of the translations to ensure their correctness.

	#DB.	#Seq.	#Pair
Train	200	3949	12892
Dev	40	755	2494
Test	40	755	2532

Table 2: mChase split statistics

3.4 Data Statistics

The mChase dataset consists of 5459 coherent question sequences (17k+ questions annotated SQL queries), including five languages (English, Chinese, French, Japanese, and Russian), based on over 300 databases, provided Schema Linking and contextual relationship annotations for each question. One example is shown in Figure 1.

Data Split Following the cross-database definition, we split mChase dataset so that each database appears once in the whole train, development, and test sets. We refer to the data split of Chase for each language and the training, development, and test set across different languages. Table 2 shows the data split statistics.

3.5 Data Analysis

The Chase dataset (Guo et al., 2021) exhibits a gradual progression of questions and SQL in the same sequence. The questions and SQL statements in the sequences often build upon the content and SQL statements of previous questions, resulting in increased overall difficulty, as shown in Table 19. This characteristic aligns with the concept of Chain-of-thought, making the mChase dataset a valuable resource for chain-of-thought research.

4 Evaluation Metrics

Beyond *Exact Match Accuracy* and *Execution Accuracy*, (Zhong et al., 2020) designs a Text-to-SQL semantic evaluation with distilled test suites that avoids possible true negatives using *Exact Match Accuracy* and possible false positives of *Execution Accuracy*(Examples are shown in Table 3). However, during the usage of the mChase dataset, we encountered two problems with these metrics: 1. The test-suite struggles to handle multi-table joints and multi-table simultaneous transformations in mChase, easy to result in None or execution error, which causes ambiguity in determining the correct-

Type	SQL
Gold SQL	`SELECT count(*) FROM players`
TN Example	`SELECT count(name) FROM players`
Gold SQL	`SELECT Name FROM country WHERE IndepYear > 1950`
FP Example	`SELECT Name FROM country WHERE IndepYear >= 1950`

Table 3: Some True-negative Cases and False-positive Cases in mChase, TN stands for False-negative and FP stands for False-positive

ness of the SQL. 2 Inconsistencies exist in Text-to-SQL datasets regarding the inclusion of commonly used column names (such as "id" and entity names) and the correctness of SQL queries involving these columns is questionable. To solve these problems, we propose *Equivalence Accuracy*.

As in the test suite, rearrangement and insertion strategies are applied to the database to do executions match several times. Rearrangement involves randomly shuffling each column and distinguishing between correct and incorrect SQL. The database is rearranged ten times when evaluating every predicted SQL to ensure accuracy. Insertion involves adding adjacent values related to the value in gold SQL. For text values, the inserted data will contain the text, while other columns are randomly selected from tables. The inserted data will be a number or a sample of Gaussian distribution for numeric values. After continuous rearrangement and insertion, if the outputs of Gold SQL and predicted SQL remains consistent, the two SQLs are equivalent.

Additionally, *Equivalence Accuracy* solved the problem of multi-table by strongly restricting the content and position of the JOIN and WHERE keywords in gold SQL and predicted SQL based on the databases. Specifically, when inserting data into databases, the tables referred to in SQL should be considered simultaneously, and the inserted data should have the same value in the primary and foreign keys of tables. In that case, the execution result is no longer empty and can be used to evaluate the prediction. For question 2, consider practical applications where users generally accept additional dispensable information if the machine outputs all the required information. Therefore, *Equivalence Accuracy* first judges whether the selected columns of gold and predicted SQL are consistent. It is considered correct if the predicted SQL query selects one more column without involving additional tables compared to the gold SQL query. More details are available in Appendix A.4.

Due to the inability to execute on mChase, the experimental results of the test-suite evaluation are not included in the experimental section.

5 Experiments

mChase covers five languages, of which only English and Chinese have been studied. There needs to be a study on context-dependency Text-to-SQL in French, Japanese, and Russian. As the primary study of Text-to-SQL in French, Japanese and Russian, we developed some reasonable approaches and conducted studies based on it.

5.1 Experiments setup

Approaches We utilized four strong pre-training models, two state-of-the-art methods on the spider leaderboard, and one chatbot API on mChase. Methods includes BERT(Devlin et al., 2018), BART(Lewis et al., 2019), T5(Raffel et al., 2020), FLANT5(Chung et al., 2022), Duorat(Scholak et al., 2020), RESDSQL(Li et al., 2023a) and ChatGPT. We added a grammar-based decoder to bert-base-multilingual-uncased model and employed the original structure of mbart-large-50, mt5-base, and flan-t5-base. We modified the input format to adapt to RESDSQL. For ChatGPT, we utilized gpt-3.5-turbo model API for zero-shot learning. More details are offered in Appendix A.6.

The introduction to the google/flan-t5-base model in Huggingface indicates that the flan-t5 model is suitable for tasks in 60 languages. However, in actual use, the model cannot work in Chinese, Japanese, and Russian.

Datasets We adjusted mChase in five different languages to adapt to the models. We also made slight changes to the question marks in the French dataset and the commas in the Russian dataset for adaptation. More details are offered in Appendix A.6.

Evaluation Metrics The evaluation metrics used in Section 5.2 are Exact Match, Execution Match, and Equivalent Accuracy. After verifying the effectiveness of Equivalent Accuracy, all evaluation metrics of a single question-SQL pair in this paper are Equivalent Accuracy. The article also uses Question Match and Interaction Match proposed by Yu et al. (2019b) in some tables, and those not marked only use Question Match.

Evaluation Metrics		English	Chinese	French	Russia	Japanese
QM	Exact Match	40.66	37.85	21.02	22.09	24.57
	Execution Match	42.00	38.79	23.96	22.94	26.12
	Equivalent Accuracy	41.76	38.01	22.83	22.17	25.22
IM	Exact Match	18.50	14.89	6.26	5.70	8.80
	Execution Match	19.23	16.23	9.80	9.00	10.20
	Equivalent Accuracy	19.0	15.63	7.85	7.15	9.00

Table 4: Question Match and Interaction Match of different language versions of mChase dataset.

Model	Chinese		Japanese		Japanese*		French		Russia	
	H	M	H	M	H	M	H	M	H	M
BERT	38.01	35.82	25.22	10.85	25.22	22.83	23.96	22.17	22.94	18.64
BART	14.83	12.32	10.13	7.78	10.13	9.31	10.06	8.79	11.08	9.40
T5	15.51	14.00	12.77	5.4	12.77	8.78	11.75	10.15	9.28	7.33
FLAN T5	-	-	-	-	-	-	14.53	13.72	-	-

Table 5: Question Match using Equivalent Accuracy between human translations and machine translations in different versions of the mChase. 'Japanese*' means the Chinese-originated machine translation.

	EN	ZH	JP	FR	RU
Duorat	45.39	40.3	25.02	24.45	24.85
+SL	52.00	49.3	29.27	36.49	32.32
+CR	44.31	44.10	27.51	26.90	27.27

Table 6: Question Match using Equivalent Accuracy of mchase using schema linking and contextual relationship or not. SL stands for schema linking and CR stands for contextual relationship.

5.2 The effectiveness of mChase

Equivalent Accuracy We first conduct experiments on Equivalent Accuracy to judge its effectiveness. As our method is based on a test suite with existing theoretical support(Zhong et al., 2020), we conducted experimental demonstrations to validate its effectiveness. We use Exact Match, Execution Match, and Equivalent Accuracy metrics simultaneously and use BERT model to conduct experiments on mChase. The experimental results are shown in Table 4. More results based on T5 model are provided in Appendix A.7 for the length limitation.

Table 4 shows that for each language experiment, the digit of Equivalence Accuracy is between Exact Match and Execution Match, aligning with the expected theoretical value. Moreover, the relaxation of the answer in Section 4 does not lead to many discrepancies on mChase. Manual sampling and evaluation of the results also revealed no errors. The experiment highlights the effectiveness of Equivalence Accuracy, and thus, we exclusively utilize it in subsequent experiments.

Human translation To examine the necessity of human translation, we conducted experiments comparing human and machine translation. We utilized Google Translation API to machine-translate English mChase to other languages. Considering real-life user scenarios, in this part of the experiment, we used human-translated or machine-translated mChase as the training set and the corresponding language's human-translated mChase as the test set. Table 5 compares human and machine translation in various language scenarios. We made two versions of translations for the Japanese part; one is translated from English, and the other is translated from Chinese.

Table 5 demonstrates that the performance gap between human and machine translation is noticeable, particularly in Japanese mChase. The accuracy of the machine-translated Japanese version of mChase is less than half that of the human-translated Japanese mChase (25.22 vs. 10.85). However, the accuracy improves significantly when using a Chinese-oriented machine-translated Japanese version (25.22 vs. 22.83). The results underscore the detrimental impact of automatically constructed datasets in other languages on Text-to-SQL approaches. This verifies the importance of a manually created multi-language dataset and reflects the significance of the mChase dataset.

Annotations We also conducted basic experiments on schema linking and context relationships in mChase. Table 6 presents the results of question match of the Duorat model(Scholak et al., 2020) in five languages mChase, from which we can find that the annotated Schema Linking and contextual relationships in mChase improve the approaches, proving the effectiveness of the role of Schema

Model	Metric	English	Chinese	French	Japan	Russia
BERT+Grammer Decoder	QM	41.76	38.01	22.83	25.22	22.17
	IM	19.0	15.63	7.85	9.00	7.15
BART	QM	15.32	14.83	10.06	10.13	11.08
	IM	4.11	4.11	2.91	3.18	3.04
T5	QM	16.40	15.51	13.74	12.77	9.82
	IM	5.30	4.64	4.37	4.11	2.25
RESDSQL	QM	54.13	49.72	39.73	38.53	38.01
	IM	18.01	15.76	12.58	13.24	12.84
GPT-3-turbo	QM	29.43	27.35	18.72	22.61	18.12
	IM	7.94	7.81	4.24	5.03	5.17
FLAN T5	QM	19.28	-	14.53	-	-
	IM	5.16	-	1.99	-	-

Table 7: Question and Interaction Match using Equivalent Accuracy of different language versions of mChase.

Linking. The structure and implementation details of the model are given in the Appendix A.6.

5.3 Prior studies in mChase

Model		BERT	BART	T5
Language	ZH	38.79	14.83	15.51
	+EN	39.86	14.64	15.00
	EN+	41.30	26.43	16.40
	+FR	38.81	14.60	15.67
	FR+	40.34	14.92	16.03
	+JP	39.57	15.04	16.12
	JP+	**41.78**	**28.87**	**20.23**
	+RU	38.89	14.43	16.48
	RU+	41.54	15.03	15.72
	+ALL	36.13	10.06	10.06

Table 8: Question Match using Equivalent Accuracy adding different languages in Chinese mChase

The Text-to-SQL task is a downstream task of the large model. We aim to explore downstream tasks of the multilingual general model using multilingual Text-to-SQL research as an example. We seek to answer three research questions(RQs): **RQ1** What is the performance of mChase in different languages on the text-to-SQL model? **RQ2** Can mChase in different languages improve the accuracy of text-to-SQL? **RQ3** How do the different pretrained models differ on mChase?

RQ1: Performance in language diversity

Table 7 presents the experimental results of the horizontal comparison of five languages, including the basic general model, Text-to-SQL state-of-the-art model RESDSQL and dialogue robot ChatGPT. We used different model parameters in all models but used the same parameters in different language experiments of the same model since our focus was on comparing the performance of the same model across different languages.

From Table 7, the English version performs the best in all evaluation metrics scenarios, followed by the Chinese version. The overall performances in French, Japanese, and Russian datasets are similar. The lower performance on the French and Russian datasets may be attributed to schema modifications we made to accommodate the approaches. We also conducted experiments on the gpt-3.5-turbo model, which revealed that the mChase dataset still poses a certain challenge for the gpt-3.5-turbo model.

RQ2: Cross-linguistic promotions

Wu et al. (2019); Pires et al. (2019) have proposed that the mBERT model exhibits transferability across languages, primarily focusing on the transfer ability between completely different languages in training and test sets. José and Cozman (2021) also did experiments on Portuguese and English Spider. To explore whether different language families can learn from each other, we conducted a series of experiments on mChase.

When fine-tuning a language-specific Text-to-SQL model, we added a certain proportion of mChase of other languages to the training set (the ratio chosen in this paper is 1:1, and we replaced the output SQLs of the added language with those of the original language), and we performed experiments on Chinese and French, respectively. Specifically, for the Chinese mChase experiment, the training set concatenates another language's and Chinese mChase's training set, and the development set is a Chinese data set. When inserting the data from another language in one language, we added three forms before or after the other language while keeping the training set unshuffled and shuffling. We used three/four multilingual versions of pre-trained models for experiments. The results

of Chinese mChase are shown in Table 8. We did not show the results of the shuffled version in the table because the results fluctuate too much after shuffling (range at most from 0.01% to 29.0%), and we released the French version and detailed parameters in Appendix A.7 due to the length limitation.

Table 8 demonstrates a noticeable improvement when adding one other language to the training set, particularly when the Japanese version is added before the Chinese version without shuffling. However, when five languages were added together to the training set, the effect did not increase but decreased. Overall, there are three conclusions: (1) Accuracy can be higher if added language is similar in linguistic origins, (2) It is better to put multiple languages at the front end of the original data, and (3) the introduction of other languages will improve the accuracy in different degrees.

Table 8 shows that adding data of the same task in other languages can effectively improve the performance when fine-tuning a language-specific model, demonstrating that the general model possesses cross-linguistic and transferability capabilities to enhance downstream task performance.

This experiment suggests that future researchers consider adding corpora from other languages for the same task when fine-tuning a language-specific model, especially when the corpus is not so rich in the language-specific scene, to improve the model's accuracy. However, the specific ratio in this task needs further exploration.

RQ3: Outputs of different models

	Easy	Medium	Hard	Extra Hard
GPT-3.5-turbo	44	**7.04**	**6.08**	**1.12**
BART	38.8	3.0	0.5	0.3
T5	29.3	1.4	0.1	0
FLANT5	**44.7**	2.3	0.1	0

Table 9: Question Match based on Equivalent Accuracy classified by SQL Hardness.

When observing the accuracy and content of SQL output by different models, we found patterns in the length and difficulty of SQL by different models. Table 9 is the accuracy distribution classified by the four models' SQL hardness. SQL hardness is defined as a four-level complexity for SQLs: easy, medium, hard, and extra hard, according to the number of components, selections, and conditions in a SQL query(Yu et al., 2018b). Only ChatGPT can generate correct SQLs in the Hard and Extra Hard categories, while other models are more likely to create easy SQLs correctly.

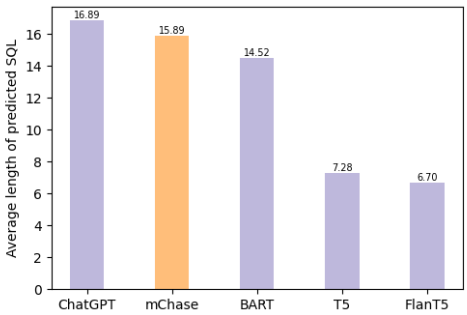

Figure 2: SQL Average Length

Figure 2 shows the average length of SQL generated by the four models, split by spaces. It can be found that the length of SQL generated by T5 and FlanT5 is limited, which means that the models cannot continue after generating a fixed length. The average length of the SQL generated by ChatGPT is longer than that of mChase, probably because some keywords that Gold SQL has not been introduced when generating the SQL.

Table 9 and Figure 2 shows that when choosing a general model, one can pre-train it using a small amount of data, observe the output of different general models, and choose the appropriate model based on the requirements. If longer-length scenes (such as dialogues) need to be generated, a BART model can be chosen, and vice versa.

6 Conclusion

This work presents mChase, a five-language, context-dependent, and complex Text-to-SQL dataset, and Equivalent Accuracy, an improved Text-to-SQL evaluation metric. Experimental results prove the effectiveness of Equivalent Accuracy and mChase dataset, highlighting the importance of manually constructed Text-to-SQL dataset, which emphasizes the challenging problem of multi-language Text-to-SQL problem. As a specific downstream task of general models, we conducted experiments using Equivalent Accuracy in mChase. The experiments show that the Text-to-SQL task performed differently on datasets of different languages, but we can choose the appropriate general models and use multilingual datasets to improve the accuracy of a single language. The research of multi-language Text-to-SQL is still so far from the real. We hope to build a global Text-to-SQL with richer languages and richer annotations to support the development of Text-to-SQL.

Limitations

Overall, there are three limitations to this work. First, there are only five language versions of the mChase dataset, and the diversity of languages needs to be more rich. The illustration of the importance of multi-language Text-to-SQL datasets can be stronger if there are more language versions. Unfortunately, we spent much time constructing datasets in French, Japanese, and Russian versions of mChase to ensure the quality of existing language versions. We hope to have the opportunity to enrich language versions of mChase in the future. Second, the number of Text-to-SQL approaches used in the experiment needs to be increased, especially lacking state-of-the-art approaches with natural language understanding related modules to verify the influence of multi-language Text-to-SQL scenarios better. Regrettably, there needs to be an approach suitable enough with an ideal Interaction Match at the time of writing. This point is expected to follow up research.
Third, the experiments we have done on the multilingual general model are only preliminary experiments, and most of them are only experimental phenomena, lacking theoretical support and more experimental support.

Ethics Statement

This work presents mChase, a free and open platform for multi-language and context-dependent Text-to-SQL problems, including a large-scale five-language context-dependent Text-to-SQL dataset and Equivalent Accuracy, an improved evaluation metric in Text-to-SQL. mChase originated from Chase, a free and open dataset for the Chinese Context-dependent Text-to-SQL problem. We recruit 2 Chinese college professors who research the corresponding language and 22 native speakers or native background graduate student (12 for French, 5 for Japanese, and 7 for Russia). Each of them is either native to the language of the corresponding country or has lived there for at least one year. Our payment is calculated based on the workload. Each sequence typically includes 3-5 question-SQL pairs, and we pay each annotator $5 per dialogue set. Since Chase is an open-access dataset, and the test part is shared with us by the Chase team, there is no privacy issue. The translation details are presented in Section 3.

References

Daria Bakshandaeva, Oleg Somov, Ekaterina Dmitrieva, Vera Davydova, and Elena Tutubalina. 2022. Pauq: Text-to-sql in russian. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2355–2376.

Ruisheng Cao, Lu Chen, Zhi Chen, Yanbin Zhao, Su Zhu, and Kai Yu. 2021. Lgesql: line graph enhanced text-to-sql model with mixed local and non-local relations. *arXiv preprint arXiv:2106.01093*.

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.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Xiang Deng, Ahmed Hassan Awadallah, Christopher Meek, Oleksandr Polozov, Huan Sun, and Matthew Richardson. 2020. Structure-grounded pretraining for text-to-sql. *arXiv preprint arXiv:2010.12773*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Li Dong and Mirella Lapata. 2018. Coarse-to-fine decoding for neural semantic parsing. *CoRR*, abs/1805.04793.

Jiaqi Guo, Ziliang Si, Yu Wang, Qian Liu, Ming Fan, Jian-Guang Lou, Zijiang Yang, and Ting Liu. 2021. Chase: A large-scale and pragmatic chinese dataset for cross-database context-dependent text-to-sql. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2316–2331.

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. Association for Computational Linguistics.

Marcelo Archanjo José and Fabio Gagliardi Cozman. 2021. mrat-sql+ gap: a portuguese text-to-sql transformer. In *Intelligent Systems: 10th Brazilian Conference, BRACIS 2021, Virtual Event, November 29–December 3, 2021, Proceedings, Part II 10*, pages 511–525. Springer.

Amol Kelkar, Rohan Relan, Vaishali Bhardwaj, Saurabh Vaichal, Chandra Khatri, and Peter Relan. 2020.

Bertrand-dr: Improving text-to-sql using a discriminative re-ranker. *arXiv preprint arXiv:2002.00557*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Haoyang Li, Jing Zhang, Cuiping Li, and Hong Chen. 2023a. Decoupling the skeleton parsing and schema linking for text-to-sql. *arXiv preprint arXiv:2302.05965*.

Jinyang Li, Binyuan Hui, Reynold Cheng, Bowen Qin, Chenhao Ma, Nan Huo, Fei Huang, Wenyu Du, Luo Si, and Yongbin Li. 2023b. Graphix-t5: Mixing pre-trained transformers with graph-aware layers for text-to-sql parsing. *arXiv preprint arXiv:2301.07507*.

Qingkai Min, Yuefeng Shi, and Yue Zhang. 2019. A pilot study for chinese sql semantic parsing. *arXiv preprint arXiv:1909.13293*.

Anh Tuan Nguyen, Mai Hoang Dao, and Dat Quoc Nguyen. 2020. A pilot study of text-to-sql semantic parsing for vietnamese. *arXiv preprint arXiv:2010.01891*.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert ? *arXiv preprint arXiv:1906.01502*.

Mohammadreza Pourreza and Davood Rafiei. 2023. Din-sql: Decomposed in-context learning of text-to-sql with self-correction. *arXiv preprint arXiv:2304.11015*.

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*.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Torsten Scholak, Raymond Li, Dzmitry Bahdanau, Harm de Vries, and Chris Pal. 2020. Duorat: towards simpler text-to-sql models. *arXiv preprint arXiv:2010.11119*.

Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. 2021. Picard: Parsing incrementally for constrained auto-regressive decoding from language models. *arXiv preprint arXiv:2109.05093*.

Ningyuan Sun, Xuefeng Yang, and Yunfeng Liu. 2020. Tableqa: a large-scale chinese text-to-sql dataset for table-aware sql generation. *arXiv preprint arXiv:2006.06434*.

Peter Trudgill and Jean Hannah. 2017. *International English: A guide to varieties of English around the world*. Taylor & Francis.

Bailin Wang, Richard Shin, Xiaodong Liu, Oleksandr Polozov, and Matthew Richardson. 2019. Rat-sql: Relation-aware schema encoding and linking for text-to-sql parsers. *arXiv preprint arXiv:1911.04942*.

Lijie Wang, Ao Zhang, Kun Wu, Ke Sun, Zhenghua Li, Hua Wu, Min Zhang, and Haifeng Wang. 2020. Dusql: A large-scale and pragmatic chinese text-to-sql dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6923–6935.

Shijie Wu, Alexis Conneau, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Emerging cross-lingual structure in pretrained language models. *arXiv preprint arXiv:1911.01464*.

Tao Yu, Zifan Li, Zilin Zhang, Rui Zhang, and Dragomir Radev. 2018a. Typesql: Knowledge-based type-aware neural text-to-sql generation. *arXiv preprint arXiv:1804.09769*.

Tao Yu, Rui Zhang, He Yang Er, Suyi Li, Eric Xue, Bo Pang, Xi Victoria Lin, Yi Chern Tan, Tianze Shi, Zihan Li, et al. 2019a. Cosql: A conversational text-to-sql challenge towards cross-domain natural language interfaces to databases. *arXiv preprint arXiv:1909.05378*.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, et al. 2018b. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. *arXiv preprint arXiv:1809.08887*.

Tao Yu, Rui Zhang, Michihiro Yasunaga, Yi Chern Tan, Xi Victoria Lin, Suyi Li, Heyang Er, Irene Li, Bo Pang, Tao Chen, et al. 2019b. Sparc: cross-domain semantic parsing in context. *arXiv preprint arXiv:1906.02285*.

Chen Zhao, Yu Su, Adam Pauls, and Emmanouil Antonios Platanios. 2022. Bridging the generalization gap in text-to-sql parsing with schema expansion. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5568–5578.

Ruiqi Zhong, Tao Yu, and Dan Klein. 2020. Semantic evaluation for text-to-sql with distilled test suite. In *The 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

A Appendix

A.1 The definition of contextual relationship used in mChase dataset

We define five kinds of context phenomenon: Context Independent, coreference, ellipsis substation, and ellipsis continuation. Context Independent means that the current question has enough information to be independent. Coreference means some tokens directly refer to entities mentioned in the precedent question. Ellipsis Substitution means that there are relationships between the current question and the precedent question. However, there is no specific token in the present question to refer to the entity in the last question. Furthermore, the relationship is not an inclusive one. Ellipsis Continuation means that there are relationships between the current question and the precedent question. However, there is no specific token in the current question to refer to the entity in the last question. Moreover, the relationship is an inclusive one. Far aside means that the current question is not independent enough to generate SQL queries and relies on the question earlier than the precedent question. Examples of question sequences with SQL queries annotating contextual relationships in different languages are given in Figure 10.

A.2 The user interface of our annotation tool

An example of the user interface to annotate the Schema Linking in Japanese is shown in Figure 5. Figure 5 is also an example of schema linking. The linking annotation can be used in model to learn the relation dependency between the questions and the schema of the database

Figure 6 is an example of the user interface to translate the schema of the database, which is translated from English.

A.3 Examples and prediction examples in mChase

Table 10 offers some examples in different languages in mChase.

More examples in case study for French, Japanese and Chinese is shown as Table 12, Table 13 and Table 11.

Table 14 shows the prediction in the Russian version of mChase. *y1* is the output of machine translation and *y2* is the output of human translation. It consists of two questions in a question sequence. Both predicted SQL queries in the first round are correct. The predicted SQL of machine translation in the second round failed to generate the join table, probably because of the ellipsis phenomenon between the questions. More examples in other language versions are presented in Appendix A.6.

name	age
David	22
David	23
Sally	26

name	age
Sally	22
David	23
David	26

Figure 3: database before rearrangement (left) and after rearrangement (right)

A.4 Equivalent Accuracy

Inspired by (Zhong et al., 2020), who designs a new semantic evaluation for Text-to-SQL with distilled test suites, we propose a new evaluation metrics for Text-to-SQL task, called ***Equivalence Accuracy***. *Equivalence Accuracy* is simpler than (Zhong et al., 2020) but achieve competitive evaluation effectiveness. Experiments also show that the newly proposed metrics perform better than metrics based on *Logical Form Accuracy* and *Execution Accuracy* in the C2C dataset.

A.5 Implementation Detail

The invisibility of the databases may influence the performance of *Execution Accuracy*. For example, on the left table of Figure 3, if the ground truth SQL query is `"SELECT name FROM person WHERE age = 23"` while the model gives a wrong prediction `"SELECT name FROM person WHERE age = 22"`, *Execution Accuracy* will consider it a positive sample because the execution results are both "*David*". We use rearrangement and insertion strategies on evaluation databases to generalize to unseen databases.

Rearrangement As shown in Figure 3, we shuffle each column randomly and independently to resolve the proposed problem. In the case of the right table of Figure 3, the wrongly predicted SQL query returns "*Sally*" instead of "*David*", so we can tell them apart. Indeed, in our experiment, databases are rearranged ten times when evaluating every predicted SQL query to ensure correctness. Rearranging the corresponding tables referred to in SQL is still a useful strategy when encountering *JOIN* query.

Insertion The other strategy in *Equivalence Accuracy* is insertion. It is designed to address two problems in databases: (1) The WHERE condition

#	Question & SQL Query	Contextual Dependency
Question Sequence $\mathcal{Q}^1$		
q_1^1	Existe-t-il le livre "Autant en emporte le vent" de Margaret? (Is there the book "Gone with the Wind" by Margaret?)?	Independent
y_1^1	*select livre from livres_en_langues_étrangères where nom_anglais = "Gone with the Wind"* *(select book title from foreign language book where English name = "Gone with the Wind")*	
q_2^1	Qui est le traducteur? (Who is the translator?)	Dependent (Coreference)
y_2^1	*select T3. traducteur from publication_de_livre AS T1 JOIN livres_en_langues_étrangères AS T2 ON T1. Id_de_livre = T2.id JOIN traducteur AS T3 ON T1. Id_de_traducteur = T3. Id_de_traducteur where T2. nom_anglais = "Gone with the Wind" (select T3.name from book_publication_information AS T1 JOIN foreign_language_book AS T2 ON T1. book_id = T2.id JOIN translator AS T3 ON T1. translator_id = T3. translator_id where T2. English_name = "Gone with the Wind")*	
q_3^1	Quelle maison d'édition publie ce livre? (Which publisher published this book?)	Dependent (Ellipsis)
y_3^1	*select T3.nom from publication_de_livre AS T1 JOIN livres_en_langues_étrangères AS T2 ON T1. Id_de_livre = T2.id JOIN maisons_d'édition AS T3 ON T1. Id_de_maison_d'édition = T3. Id_de_maison_d'édition where T2. nom_anglais = "Gone with the Wind" (select T3.name from book_publishing_information AS T1 JOIN foreign_language_book AS T2 ON T1. book_id = T2.id JOIN publishing_house AS T3 ON T1. publishing_house_id = T3. publishing_house_id where T2. English_name = "Gone with the Wind")*	
Question Sequence $\mathcal{Q}^2$		
q_1^2	どのブランドの市シェアが一番高いかべたいです。(I want to investigate which brand has the largest market share?)	Independent
y_1^2	*select 洗濯ブランド名 from 洗濯ブランド order by 市シェア desc limit 1* *(select name from washing machine brand order by market share desc limit 1)*	
q_2^2	タオバオでその能にするレビュは何点ですか? (What is its rating on Taobao?)	Dependent (Coreference)
y_2^2	*select T2. 能の得点 from 洗濯ブランド AS T1 JOIN 洗濯ブランドのプラットフォムの得点 AS T2 ON T1. のID = T2. のID where T2. プラットフォム = "タオバオ" order by T1. 市シェア desc limit 1 (select T2. function score from washing machine brand AS T1 JOIN washing machine brand platform score AS T2 ON T1. brand id = T2. brand id where T2. platform = "Taobao" order by T1. market share desc limit 1)*	
Question Sequence $\mathcal{Q}^3$		
q_1^3	哪个节日人们会吃腊八粥? (Which festival do people eat Laba porridge?)	Independent
y_1^3	*select T2.名称 from 节日饮食文化 AS T1 JOIN 传统节日 AS T2 ON T1.节日id = T2.节日id where 饮食 = "腊八粥" (select T2.name from festival food culture AS T1 JOIN traditional festival AS T2 ON T1. festival id = T2. festival id where food = "Laba porridge")*	
q_2^3	是哪个城市有这种习俗? (Which city has this custom?)	Dependent (Coreference)
y_2^3	*select T1.城市 from 节日饮食文化 AS T1 JOIN 传统节日 AS T2 ON T1.节日id = T2.节日id where 饮食 = "腊八粥" (select T1. city from festival food culture AS T1 JOIN traditional festival AS T2 ON T1. festival id = T2. festival id where food = "Laba porridge")*	
q_3^3	对了，那哪个节日有吃腊八蒜的习惯? (By the way, which festival has the habit of eating Laba garlic?)	Independent
y_3^3	*select T2.名称 from 节日饮食文化 AS T1 JOIN 传统节日 AS T2 ON T1.节日id = T2.节日id where 饮食 = "腊八蒜" (select T2. Name from Festival Food Culture AS T1 JOIN Traditional Festival AS T2 ON T1. Festival id = T2. Festival id where food = "Laba Garlic")*	
Question Sequence $\mathcal{Q}^4$		
q_1^4	В какой стране находится Париж? (In which country is Paris located?)	Independent
y_1^4	select страны from город where имя = "Париж" (select country from city where name = "Paris")	
q_2^4	Пожалуйста, найдите принимающее население 2015 года. (Please find out the reception population in 2015.)	Dependent (Ellipsis)
y_2^4	select принимать население from количество туристов AS T1 JOIN город AS T2 ON T1. городid = T2. городid where имя = "Париж"and год = "2015" (select reception population from tourist count AS T1 JOIN city AS T2 ON T1.cityid = T2.cityid where name = "Paris"and year = "2015")	
q_3^4	А вышеупомянутый ежегодный доход? (What about that year's income?)	Dependent (Coreference)
y_3^4	select Поступления from количество туристов AS T1 JOIN город AS T2 ON T1.городid = T2. городid where имя = "Париж"and год = "2015" (select income from number of tourists AS T1 JOIN city AS T2 ON T1.cityid = T2.cityid where name = "Paris"and year = "2015")	

Table 10: Question sequence examples in mChase dataset.

in the SQL query is not satisfied; (2) The JOIN condition in the SQL query is not satisfied. The two problems lead to empty execution results, so the original *Execution Accuracy* does not work.

To address the first problem, we insert the databases according to the WHERE condition in the ground truth SQL query. If the value in the WHERE condition is a text, then the inserted data will contain the text, while other columns are randomly selected from tables. If the value in WHERE conditions is a number, the corresponding column in the inserted data will be either the number or a sample from Gaussian distribution, which takes the number as its mean. After insertion, the WHERE condition in the ground truth is satisfied, and *Execution Accuracy* works.

On the other hand, we find that the JOIN condition is only sometimes satisfied and brings empty execution results. We also use an insertion strategy to resolve it. When inserting data into databases, the tables referred to in SQL should be considered simultaneously. Specifically, the inserted data should have the same value in the primary and foreign keys of tables. In that case, the execution result is no longer empty and can be used to evaluate the prediction.

A.6 Details in experiments

Environments There are three types of environments used in the experiment, two of which are the

q_1	Veuillez me donner toutes les informations sur le vote. (Please give me all the voting information.)
y_1	*SELECT * FROM votes*
$\hat{y}_1$	*SELECT * FROM contestants*

Table 11: Predictions $\hat{y}_j$ of a French question sequence in mChase. SQL queries are translated to English.

q_1	列出所有主人的信息。(List information of all owners.)
y_1	*SELECT * FROM owners*
$\hat{y}_1$	*SELECT * FROM owners*
q_2	专业人士的呢？(What about professionals?)
y_2	*SELECT * FROM professionals*
$\hat{y}_2$	*SELECT * FROM owners JOIN dogs on dogs. owner id = owners. owner id JOIN professionals JOIN treatments on treatments. professional id = professionals. professional id*

Table 12: Predictions $\hat{y}_j$ of a Chinese question sequence in mChase. SQL queries are translated to English.

Ubuntu18.04.2LTS operating system and the other is the CentOS operating system. Both use the PyTorch deep learning development framework and Python as the development language. The GPUs are NVIDIA GeForce with 24G video memory, 2TITAN RTX with 24G video memory, and GRID T4-8Q with 8G video memory. Adam is the optimizer in the training process, and the initial learning rate is 0.0005. BeamSearch's Grammar Decoder is added to the back end of the BERT model, the Batchsize is 12, and the Epoch is set to 50; the BART model's Batchsize is 10, and the Epoch is set to 150; the Batchsize of the T5 and FlanT5 models is 12, and the Epoch is selected to 150; the Duorat model uses bert-base -multilanguage-uncased, Batchsize is 8, Epoch is set to 50. RESDSQL uses the mt5-base model, Batchsize is 12, and Epoch is set to 200. ChatGPT uses API for zero-shot learning. The batch size of all models is selected based on the upper limit of the current card's video memory. The shortest running time is BART, T5, FLANT5, then RESDSQL and BERT, and finally DUORAT.

At the same time, we do not use any word tokenization to segment the words in different languages; For English, Russian, and French, the tokenization is based on the word level; For Chinese and Japanese, the tokenization is based on the character level. The parameters of all experiments are the same in different languages. The datasets used are the same except for languages and language-related content. French single quotes and Russian commas are converted to underscores during execution to avoid the possible impact of French and Russian when parsing.

Figure 4 shows the model using the contextual relationship.

A.7 Supplementation of results in Experiments

The Equivalent Accuracy's experiments in T5 and the statistic of adding other languages to the French mChase are offered here.

A.8 Chain-of-thought Examples

Table 19 is an English example in mChase from easy to hard, which is consistent with the idea of chain-of-thought.

q_1	すべての地域情を教えてください。 (Please tell me all the regional information.)
y_1	*SELECT * FROM area code state*
$\hat{y}_1$	*SELECT area code state. area code, contestants. contestant name FROM votes JOIN area code state on votes. state = area code state. state*

Table 13: Predictions $\hat{y}_j$ of a Japanese question sequence in mChase. SQL queries are translated to English.

q_1	Подробно опишите всех кандидатов. (Please describe all candidates in detail.)
y_1	*SELECT * FROM contestant*
$\hat{y}_1$	*SELECT * FROM contestants*
q_2	Подробная информация обо всех бюллетенях. (Detailed information about all votes.)
y_2	*SELECT * FROM votes*
$\hat{y}_2$	*SELECT * FROM contestants JOIN votes on votes.vote id = contestants. contestant number*

Table 14: Predictions $\hat{y}_j$ of a Russian question sequence in mChase. SQL queries are translated to English.

prediction	**ground truth**
`SELECT T1.Model FROM CAR_NAMES AS T1 JOIN CARS_DATA AS T2 ON T1.MakeId = T2.Id WHERE T2.horsepower = (SELECT MIN(horsepower) FROM CARS_DATA)`	`SELECT T1.Model FROM CAR_NAMES AS T1 JOIN CARS_DATA AS T2 ON T1.MakeId = T2.Id ORDER BY T2.horsepower ASC LIMIT 1`
`SELECT count(*) FROM players`	`SELECT count(name) FROM players`

Table 15: Some False-negative Cases in Sparc Dataset

prediction	**ground truth**
`SELECT Name FROM country WHERE IndepYear > 1950` (Same execution result because any IndepYear isn't equal to 1950 in the database)	`SELECT Name FROM country WHERE IndepYear >= 1950`
`SELECT T1.name FROM conductor AS T1 JOIN orchestra AS T2 ON T1.Conductor_ID = T2.Conductor_ID GROUP BY T2.Conductor_ID HAVING COUNT(*) > 1` (The JOIN condition is not satisfied)	`SELECT T1.age FROM conductor AS T1 JOIN orchestra AS T2 ON T1.Conductor_ID = T2.Conductor_ID GROUP BY T2.Conductor_ID HAVING COUNT(*) > 1`

Table 16: Some False-positive Cases in Sparc Dataset Using *Execution Accuracy*

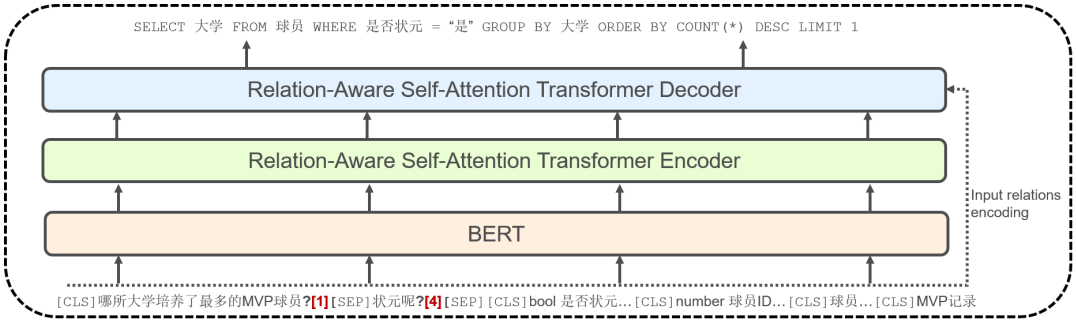

Figure 4: The model using the contextual relationship

Figure 5: The user interface of our annotation tool.

Figure 6: The user interface of a database in mChase

Evaluation Metrics		English	Chinese	French	Japanese	Russia
QM	Exact Match	16.16	15.10	11.33	11.97	9.46
	Execution Match	16.40	15.51	14.77	14.53	9.82
	Equivalent Accuracy	16.40	15.51	13.74	12.77	9.82
IM	Exact Match	5.30	2.65	3.97	3.44	2.12
	Execution Match	5.30	4.64	4.37	4.90	2.25
	Equivalent Accuracy	5.30	4.64	4.37	4.11	2.25

Table 17: EA experiments based on T5 model

Model		BERT	BART	T5	FLANT5
Language	FR	22.83	10.06	13.74	14.53
	+EN	23.58	12.63	13.87	14.64
	EN+	26.78	15.02	14.96	15.24
	+ZH	23.01	10.63	12.20	-
	ZH+	24.98	14.15	13.71	-
	+JP	22.89	9.78	12.35	-
	JP+	22.73	12.51	13.07	-
	+RU	22.94	9.82	13.03	-
	RU+	24.26	13.43	14.03	-
	+ALL	21.50	9.42	12.01	-

Table 18: Question Match using Equivalent Accuracy adding different languages in French mChase

q_1	The number of online novels written by Anna Todd
y_1	select count (t1.title) from network_novel as t1 join author as t2 on t1.author_id = t2.author_id where t2.full_name = "Anna Todd"
q_2	What is the name of her published book?
y_2	select t1.title from publishing_books as t1 join author as t2 on t1.author_id = t2.author_id where t2.full_name = "Anna Todd"
q_3	List the names of the books with higher scores than her published book.
y_3	select title from publishing_books where score >(select t1.score from publishing_books as t1 join author as t2 on t1.author_id = t2.author_id where t2.full_name = "Anna Todd")

Table 19: Examples in mCHASE