MultiSpider: Towards Benchmarking Multilingual Text-to-SQL Semantic Parsing

Longxu Dou¹, Yan Gao², Mingyang Pan¹, Dingzirui Wang¹, Wanxiang Che¹, Dechen Zhan¹, Jian-Guang Lou²
¹ Harbin Institute of Technology
² Microsoft Research Asia
{lxdou, mypan, dzwang, car}@ir.hit.edu.cn, dechen@hit.edu.cn, {yan.gao, jlou}@microsoft.com

Abstract

Text-to-SQL semantic parsing is an important NLP task, which greatly facilitates the interaction between users and the database and becomes the key component in many human-computer interaction systems. Much recent progress in text-to-SQL has been driven by large-scale datasets, but most of them are centered on English. In this work, we present MULTIPIIDER, the largest multilingual text-to-SQL dataset which covers seven languages (English, German, French, Spanish, Japanese, Chinese, and Vietnamese). Upon MULTIPIIDER, we further identify the lexical and structural challenges of text-to-SQL (caused by specific language properties and dialect sayings) and their intensity across different languages. Experimental results under three typical settings (zero-shot, monolingual, and multilingual) reveal a 6.1% absolute drop in accuracy in non-English languages. Qualitative and quantitative analyses are conducted to understand the reason for the performance drop of each language. Besides the dataset, we also propose a simple schema augmentation framework SAVE (Schema-Augmentation-with-Verification), which significantly boosts the overall performance by about 1.8% and closes the 29.5% performance gap across languages.

1 Introduction

Text-to-SQL semantic parsing is the task of mapping natural language sentences into executable SQL database queries, which serves as an important component in many natural language interface systems such as question answering and task-oriented dialogue. Despite the substantial number of systems (Yin and Neubig 2018; Guo et al. 2019; Wang et al. 2020; Scholak, Schucher, and Bahdanau 2021) and benchmarks (Yu et al. 2018, 2019a,b; Guo et al. 2021) for text-to-SQL, most of them are predominantly built in English, excluding this powerful tool’s accessibility to non-English speakers. The reason for this limitation lies in the serious lack of high-quality multilingual text-to-SQL datasets.

Several works attempted to extend to new languages, but currently available multilingual text-to-SQL datasets only support four languages (English, Chinese, and Vietnamese and Portuguese) (Yu et al. 2018; Min and Zhang 2019; Tuan Nguyen, Dao, and Nguyen 2020; José and Cozman 2021), which hinders the study of multilingual text-to-SQL across a broad spectrum of language distances. Besides the language coverage, the existing multilingual datasets also suffer from the following limitations: (1) low-quality: unnatural or inaccurate translations; (2) in-completed translation: the database of Chinese-Spider and Portuguese-Spider are not translated and kept in English. These limitations will inevitably lead to a limited multilingual system. To advance multilingual text-to-SQL, in this paper, we present MULTISPIDER, the largest and high-quality multilingual text-to-SQL dataset, which covers seven main-stream languages (Sec 2.1). Figure 1 lists one example across seven languages including both question and schema. To ensure the dataset quality, we first identify five typical translation mistakes during constructing a multilingual text-to-SQL dataset (Sec 2.2), then we carefully organize the construction pipeline consisting of multi-round translation and validation (Sec 2.3). Most importantly, we take into account of the specific language properties to make the question more natural and realistic.

Besides high-quality, MULTIPIIDER is quite challenging in multilingual text-to-SQL. Concretely, we explore the lex-
ical and structural challenge (Herzig and Berant 2018) of MULTI SPIDER (Sec 2.4); (1) lexical challenge refers to mapping the entity mentions to schema alias (e.g., ‘record companies’ to RECORD, COMPANY); (2) structural challenge refers to mapping the intentions to SQL operators (e.g., ‘sorted descending’ to DESC). Experimental results and analysis demonstrate that (1) the specific language properties like Hiragana and Katakana (Japanese) and morphologically rich language (German and French) make the lexical challenge more difficult by expanding the syntactic difference between schema and tokens; (2) the dialect sayings require commonsense reasoning to address structural challenge (Figure 3).

To address the lexical challenges, we propose a simple data augmentation framework SAVe from the view of schema, which is more generic compared with the language-specific approaches (Min and Zhang 2019; Tuan Nguyen, Dao, and Nguyen 2020) (e.g. PhoBERT for Vietnamese (Nguyen and Nguyen 2020)). Concretely, SAVe consists of three steps (Sec 3.1); (1) conducting back-translation on contextualized schema using machine-translation; (2) extracting the the schema candidates; (3) measuring the semantic equivalence (Pi et al. 2022) with natural language inference model (NLI) to collect the suitable candidate. The quantitative and qualitative analysis prove that (1) the augmented schema including synonyms and morphological variants; (2) verification improve the accuracy of augmented data from 33.2% to 74.5% under human evaluation (Sec 3.2).

To examine the challenge of MULTI SPIDER and verify the effectiveness of SAVe, we conduct extensive experiments (Sec 4) under representative settings (zero-shot transfer, monolingual and multilingual). Experimental results reveal the absolute drop of accuracy in non-English languages is about 6.1% on average, indicating the difference in language causes the performance gap. SAVe significantly boosts the overall performance by about 1.8%, reducing the performance gap by 29.5% across languages. We further study two research questions: what causes the performance drop in non-English languages? (Sec 5.1) and how schema augmentation SAVe improves the model? (Sec 5.2).

Our contributions can be summarized as follows:

- To our best knowledge, MULTI SPIDER is the largest multilingual text-to-SQL semantic parsing dataset with seven languages.
- We further identify lexical challenge and structure challenge of multilingual text-to-SQL brought by specific language properties.
- We propose a simple-yet-effective data-augmentation method SAVe from the perspective of schema.
- Experimental results reveal that MULTI SPIDER is indeed challenging and SAVe significantly boosts the overall performance by about 1.8%.

## 2 The MULTI SPIDER Dataset

### 2.1 Dataset Collection and Statistic

We build MULTI SPIDER based on Spider (Yu et al. 2018), a large-scale cross-database text-to-SQL dataset in English. Only 9691 questions and 5263 SQL queries over 166 databases (train-set and dev-set) are publicly available. Thus we only translate those data.

Currently, there are two well-known extensions of Spider: (1) CSpider (Min and Zhang 2019) (Chinese, schema kept in English): we improve the existing translation and translate the schema as well. (2) VSpider (Tuan Nguyen, Dao, and Nguyen 2020) (Vietnamese): we re-partition the dataset to be the same as other languages for fair comparison. The mentioned value (e.g., location and name) in question are kept in English, to be consistent with the database content.

### 2.2 Challenge of Dataset Translation.

Based on our preliminary study, we summarize five typical mistakes during translating the text-to-SQL dataset including schema and question (Figure 2).

#### Challenge of Schema Translation

Both insufficient context and domain knowledge make the schema translation challenges, including abbreviation, domain-specific jargon, and polysemy. For example, AID could be interpreted as ‘assistance’ or ‘id of the author’ (Figure 2). We could disambiguate the meaning of schema headers by referring to its content value, neighbor headers, and involved question. Thus we can recognize AID as the abbreviation of ‘id of the author’ by examining its value ‘0001’, neighbor ‘publisher’, and the question ‘Return the aid of the best paper?’.

#### Challenge of Question Translation

We are facing two challenges here: (1) lexical challenge refers to the entity polysemy, such as the ‘capital’ in the case of Figure 2. It’s not easy to deduce the actual meaning of ‘capital’ (‘money’ or ‘metropolis’) simply based on the context of the question, but we could disambiguate its meaning by schema translation of ‘capital’ where domain knowledge is considered; (2) structural challenge points out that the complex logic or syntactic structure causes inaccurate translation. We propose to refer to the corresponding SQL query to validate the logic. For example, as shown in the last line of Figure 2, the machine translation might generate redundancy headers ‘year’. 

---

**Figure 2:** Typical mistakes during the translation, due to the lack of context information and domain knowledge. The correct translation and their explanations from WordNet.
2.3 Translation Pipeline

Hiring Qualified Translators  The translators are college students who majored in the target language\(^1\). There are three students for each language (15 students in total) who are proficient in English (e.g. IELTS >= 7.0) and also meet the criteria: (1) language certificate of the target language, i.e. TEF/TCF for French; or (2) lived abroad for years.

Translation and Validation  To be effective, we first use Google NMT to translate the Spider, then let each translator post-edit the translation individually. According to the preliminary study about translation mistakes in Sec 2.2, the translation pipeline is organized as three steps (Figure 4): (1) schema translation to let the translators leverage the content values of the corresponding schema, the neighbor headers, and the involved questions, to obtain sufficient context information of schema; (2) question translation by referring to the translated schema and translating the corresponding SQL simultaneously to valid the complex logic of the sentence; (3) cross validation to merge the annotated data through voting the best translations among three annotators.

---

\(^1\)The payment of translators is listed in Ethical Statement.

---

Table 1: The statistics of post-editing data for each language. ZH starts from CSpider while the others are translated from Google NMT.

<table>
<thead>
<tr>
<th>Question</th>
<th>DE</th>
<th>ES</th>
<th>FR</th>
<th>JA</th>
<th>ZH</th>
</tr>
</thead>
<tbody>
<tr>
<td>4,607</td>
<td>3,567</td>
<td>4,723</td>
<td>4,092</td>
<td>989</td>
<td></td>
</tr>
<tr>
<td>1,248</td>
<td>682</td>
<td>1,382</td>
<td>1,601</td>
<td>1,469</td>
<td></td>
</tr>
<tr>
<td>362</td>
<td>225</td>
<td>327</td>
<td>470</td>
<td>670</td>
<td></td>
</tr>
</tbody>
</table>
### 3 Schema Augmentation: SAVE

Lexical challenge becomes more severe in multilingual settings due to different language properties (Figure 3). To address this problem, we propose SAVE (Schema-Augmentation-with-Verification) to generate more schema variations, to improve the grounding ability of the parser.

Specifically, we first adopt machine-translation to generate the synonym candidates of schemas by multi-rounds back-translation. Then we use natural-language-inference model to select the semantic equivalency candidates, via measuring the entailment scores between schema and candidate (to ensure the data quality). Eventually, the augmented schemas would be used to expand the training data.

#### 3.1 Augmentation Pipeline

**Back Translation** generates the synonym candidates of schema (e.g. CHIEF and BRAIN are the candidates of HEAD).

At first, to leverage the context of the schema for a better translation, we design a special template to insert the information of the database and the affiliated table like \{COLUMN\} of \{TABLE\} from \{DATABASE_NAME\}. Then we translate this template from the target language into \(K\) intermediate languages. To further improve the candidate diversity, \(N\) rounds of translation are conducted between intermediate language and target language alternatively. Finally, we obtain \(K \times N\) synonym candidates (duplicate exists) in the target language.

**Table Verification**

We propose to measure the semantic equivalence between the original schema and the candidate synonym to collect the suitable candidates inspired by Pi et al. (2022). The main challenge in schema verification is to compute the similarity of contextualized schema \((head of department vs. brain of department)\). Hill, Reichart, and Korhonen (2015) shows that natural language inference (NLI) model achieves promising performance (70% accuracy) than baselines (Word2Vec (Mikolov et al. 2013) and Glove (Pennington, Socher, and Manning 2014)) in computing semantic equivalency. Thus, NLI model is a good choice to collect the synonym of schema via enumerating the candidates. Concretely, we design a template to construct hypotheses and premise using schema and candidate as input. The template is \(TABLE\) \{COLUMN\} \{TYPE\} which contextualizes the schema with table context. Finally, we compute the entailment scores from both directions \((premise to hypothesis and hypothesis to premise)\), as the judgment of semantic similarity. If they are both above the threshold \(0.68\) for Chinese and \(0.65\) for others), we select this pair as augmented schema data.

#### 3.2 Quality of Augmented Schema

To examine the effectiveness of schema verification, we conduct the human-evaluation of the augmented schema. Concretely, we sample 300 schemas from each language respectively. The accuracy (i.e., the percentage of semantic equivalent items) is about 74.5% with verification and drops drastically to 33.2% without verification.

#### 3.3 Synthesising New Training Data

Text-to-SQL data example consists of three parts: question, schema and SQL. To expand the training corpus, for each data example, we randomly replace the schema items (e.g., COLUMN or TABLE) with the corresponding augmented schemas (e.g., replace HEAD with CHIEF in the above case) to compound the new training data examples. Consequently, we expand the training data by two to three times.

### 4 Experiments

#### 4.1 Experimental Setup

**Baseline Models** We choose two types of representative models: (1) task-specific model RAT-SQL (Wang et al. 2020), equipped with pretrained multilingual encoder mBERT (Devlin et al. 2019) and XLM-Roberta-Large (Conneau et al. 2020); (2) pretrained multilingual encoder-decoder mBART (Liu et al. 2020) which is inspired by the recent work of Scholak, Schucher, and Bahdanau (2021) that reveals the excellent performance of pretrained encoder-decoder model.

**Evaluation Metric** We report results using the same metrics as (Wang et al. 2020): exact match accuracy on all examples, as well as divided by difficulty levels determined by the official evaluation script (Yu et al. 2018).

**Training with Augmented Data** During the training phase, we first adopt the augmented data to warm up the model three epochs to alleviate the noise in augmented data, then fine-tune the model with original high-quality training data.

---

1. We choose schema augmentation rather than question augmentation since it’s more efficient. The augment coverage of a single table modification includes all affiliated SQLs.

2. The back-translation would run 3 turns among 11 languages (seven languages of MULTISPIDER plus Russian, Portuguese, Dutch, Swedish), i.e. \(K = 11, N = 3\). The extra four languages are decided by their translation performance and the scale of their training corpus as reported in M2M100 paper.

3. Code available at https://github.com/microsoft/ContextualSP
4.2 Experimental Results

Follow the popular multilingual datasets MTOP (Li et al. 2021) and MultiATIS++ (Xu, Haider, and Mansour 2020), we conduct extensive experiments under three settings: zero-shot, monolingual, and multilingual. The results demonstrate that (1) the absolute drop of accuracy in non-English languages is about 6.1% on average; (2) SAVE significantly improves the performance about 1.8% overall.

<table>
<thead>
<tr>
<th>Model</th>
<th>EN</th>
<th>DE</th>
<th>ES</th>
<th>FR</th>
<th>JA</th>
<th>ZH</th>
<th>VI</th>
<th>AVG(6 langs)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Monolingual Training (only use target language training data)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mBART</td>
<td>57.3</td>
<td>39.7</td>
<td>41.3</td>
<td>37.5</td>
<td>45.7</td>
<td>55.0</td>
<td>42.2</td>
<td>43.6</td>
</tr>
<tr>
<td>mBART + SAVE</td>
<td>58.3</td>
<td>42.6</td>
<td>42.6</td>
<td>51.2</td>
<td>46.9</td>
<td>45.8</td>
<td>43.1</td>
<td><strong>45.5 (+1.9%)</strong></td>
</tr>
<tr>
<td>RAT-SQL + XLM-R</td>
<td>68.6</td>
<td>62.5</td>
<td>61.7</td>
<td>67.7</td>
<td>63.7</td>
<td>53.1</td>
<td>65.9</td>
<td>67.8</td>
</tr>
<tr>
<td>RAT-SQL + XLM-R + SAVE</td>
<td>66.8</td>
<td>59.9</td>
<td>52.7</td>
<td>65.5</td>
<td>54.3</td>
<td>66.2</td>
<td>66.1</td>
<td><strong>63.2 (+1.4%)</strong></td>
</tr>
</tbody>
</table>

* Multilingual Training (use training data from multiple languages)

<table>
<thead>
<tr>
<th>Model</th>
<th>EN</th>
<th>DE</th>
<th>ES</th>
<th>FR</th>
<th>JA</th>
<th>ZH</th>
<th>VI</th>
<th>AVG(6 langs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>mBART</td>
<td>58.3</td>
<td>42.7</td>
<td>45.9</td>
<td>42.9</td>
<td>52.2</td>
<td>57.8</td>
<td>43.2</td>
<td>47.5</td>
</tr>
<tr>
<td>mBART + SAVE</td>
<td>59.7</td>
<td>46.9</td>
<td>47.1</td>
<td>43.0</td>
<td>54.3</td>
<td>61.9</td>
<td>45.6</td>
<td><strong>49.8 (+2.3%)</strong></td>
</tr>
<tr>
<td>RAT-SQL + XLM-R</td>
<td>68.8</td>
<td>64.7</td>
<td>67.1</td>
<td>65.3</td>
<td>60.2</td>
<td>66.1</td>
<td>67.1</td>
<td>65.2</td>
</tr>
<tr>
<td>RAT-SQL + XLM-R + SAVE</td>
<td>70.8</td>
<td>66.7</td>
<td>69.3</td>
<td>67.5</td>
<td>61.6</td>
<td>67.3</td>
<td>67.8</td>
<td><strong>66.7 (+1.5%)</strong></td>
</tr>
</tbody>
</table>

Table 2: Exact-match Accuracy on MultiSPIDER for 7 languages. Notice that the AVG is calculated across 6 non-English languages to be comparable to English results. The performance boosts brought by SAVE are bolded.

**Zero-shot Transfer**  Zero-shot transfer is a realistic scenario where only the English training dataset is available. We study three fine-grained zero-shot settings:

- **Directly Predict:** The parser is trained on English. During the inference, we directly predict with the question and schema in the target-language.
- **Translate-then-Predict:** The parser is trained on English. During the inference, we first translate the input question and schema from the target-language into English using Google NMT and then predict it.
- **Translate-then-Train:** We first translate the original English dataset into the target language, then train the parser on this machine-translated training dataset.

From Table 3, we observed that (1) the performance of zero-shot transfer largely depends on the choice of pre-trained encoder, where a better model enables better zero-shot transfer, i.e. XLM-R-Large beats mBERT by a large margin; (2) compared with translation-then-test, directly predict receives better performance about 1.6% since machine-translation might create mistakes, especially for schema translation; (3) with strong pretrained language model and machine-translation model, we could receive the promising results, which reveals that machine-translated data could be an economical proxy of human-translated data as Sherborne, Xu, and Lapata (2020).

<table>
<thead>
<tr>
<th>Model</th>
<th>DE</th>
<th>ES</th>
<th>FR</th>
<th>JA</th>
<th>ZH</th>
<th>VI</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Directly Predict</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mBERT</td>
<td>50.9</td>
<td>52.2</td>
<td>40.7</td>
<td>43.1</td>
<td>49.6</td>
<td>45.3</td>
</tr>
<tr>
<td>XLM-R</td>
<td>57.6</td>
<td>60.8</td>
<td>59.1</td>
<td>48.3</td>
<td>55.5</td>
<td>56.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Translate-then-Predict</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>mBERT</td>
<td>49.6</td>
<td>51.2</td>
<td>47.6</td>
<td>39.1</td>
<td>46.7</td>
<td>43.3</td>
</tr>
<tr>
<td>XLM-R</td>
<td>58.8</td>
<td>57.2</td>
<td>58.7</td>
<td>46.3</td>
<td>55.3</td>
<td>53.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Translate-then-Train</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>mBERT</td>
<td>49.5</td>
<td>51.2</td>
<td>51.3</td>
<td>38.2</td>
<td>45.8</td>
<td>49.3</td>
</tr>
<tr>
<td>XLM-R</td>
<td>60.2</td>
<td>61.9</td>
<td>61.7</td>
<td>51.3</td>
<td>57.6</td>
<td>63.9</td>
</tr>
</tbody>
</table>

Table 3: Exact-match Accuracy under zero-shot settings.

**Monolingual Training** In this setting, the parser is trained on the human-translated training dataset in the target-language. From the results of the upper half of Table 2, we observed that (1) The performance of Japanese is significantly behind other languages. It’s mainly caused by Hiragana and Katakana, which will be further analyzed in Sec 5.1; (2) BART exhibits strong performance in English and Chinese compared with the task-specific model, indicating the potential growth of pretrained seq2seq model in text-to-SQL; (3) SAVE significantly improve the non-English languages (1.4%-1.9%) but raised less performance in English (0.2%-1.0%). We found that the most data pairs (schema and mention) in English are exactly/partly match (Gan et al. 2021), which is much easier than other languages so that it would benefit less from SAVE.

**Multilingual Training** In this setting, the parser is trained on the concatenation of training data from all languages. From the results of the bottom half of Table 2, we observed that (1) the multilingual training receives the best results overall. mBART and RAT-SQL receive a performance boost of about 3.9% from multilingual training in all languages; (2) English still benefits from multilingual training which is also proved by other multilingual datasets (Xu, Haider, and Mansour 2020; Li et al. 2021); (3) Notably, SAVE would improve the model further by 1.5%, indicating the effectiveness of data augmentation.
<table>
<thead>
<tr>
<th>Question (ZH): 哪些城市有多于一个未满30岁的员工？</th>
<th>Lexical Mistake</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Which cities do more than one employee under age 30 come from?)</td>
<td>Gold: SELECT City FROM employee WHERE Age &lt; 30 GROUP BY City HAVING Count(*) &gt; 1</td>
<td>Pred: SELECT City FROM employee WHERE Age &lt; 30 GROUP BY City HAVING Count(*) &gt; 1</td>
</tr>
<tr>
<td>Lexical Mistake refers that the schema has not been grounded in SQL, which is usually caused by the syntactic difference between schema and tokens also known as schema-linking problem (Wang et al. 2020; Lei et al. 2020).</td>
<td>Mention: 未满30岁</td>
<td>SQL Operator: Age &lt; 30</td>
</tr>
</tbody>
</table>

5 Discussion and Analysis

5.1 What Causes the Performance Drop in Non-English Languages?

In this section, we conduct both qualitative analysis and quantitative analysis about the accuracy drop in non-English languages (Sec 4.2). Concretely, we conduct case studies (Figure 6) for incorrect SQL prediction in non-English compared to the correct SQL prediction in English. All these SQL are predicted by RAT-SQL+XLM-R+SAVE under multilingual settings, which is the SOTA model in experiments (Table 2). Furthermore, we divide these bad cases into two categories: lexical mistakes and structural mistakes.

<table>
<thead>
<tr>
<th>Question (ZH): 哪些城市有多于一个未满30岁的员工？</th>
<th>Structural Mistake</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Which cities do more than one employee under age 30 come from?)</td>
<td>Gold: SELECT Count(*) FROM cars_data WHERE cylinders &gt; 4</td>
<td>Pred: SELECT Count(*) FROM cars_data WHERE weight &gt; 4</td>
</tr>
<tr>
<td>Structural Mistake refers to the incorrect prediction of SQL operators. The models are acquired to leverage the commonsense reasoning ability to match the SQL spans with intent mentions. However compared with the English, MULTI-SPIDER contains more dialect sayings in question annotation. In the last case of Figure 6, it’s difficult to deduce the actual meanings of the expression ‘Age &lt; 30’ in Chinese.</td>
<td>Mention: 未满30岁</td>
<td></td>
</tr>
</tbody>
</table>

Figure 7: Case studies of non-English languages under two categories: lexical mistakes and structural mistakes.

<table>
<thead>
<tr>
<th>Question (ZH): 哪些城市有多于一个未满30岁的员工？</th>
<th>Lexical Mistake</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Which cities do more than one employee under age 30 come from?)</td>
<td>Gold: SELECT Count(*) FROM cars_data WHERE cylinders &gt; 4</td>
<td>Pred: SELECT Count(*) FROM cars_data WHERE weight &gt; 4</td>
</tr>
<tr>
<td>Lexical Mistake refers that the schema has not been grounded in SQL, which is usually caused by the syntactic difference between schema and tokens also known as schema-linking problem (Wang et al. 2020; Lei et al. 2020).</td>
<td>Mention: 未满30岁</td>
<td>SQL Operator: Age &lt; 30</td>
</tr>
</tbody>
</table>

Figure 7: Fuzzy-match based schema-linking score.
### 5.2 How Schema Augmentation SAVE Improves the Model?

Sec 4.2 demonstrates that SAVE significantly improves the performance by about 1.8% overall in all languages across three settings. The performance gain of augmented data might come from two aspects: (1) addressing the lexical challenge by synthesizing more schema-token pairs; (2) improving the robustness of the text-to-SQL model through varies the schema input as studied by (Pi et al. 2022).

We conduct both qualitative and quantitative analysis on augmented schema to understand the reason of performance gain. For qualitative analysis, after conducting cases studies on seven languages, we roughly classify the augmented schema items into two categories (Figure 8): *synonyms* which is semantically identical with the original schema but with different lemmas (i.e. don’t have string overlap); and *morphological variants* that changes the forms of schema syntactically. For quantitative analysis, we sample 500 schemas from each language respectively, and we found that (1) for DE and ES, the most augmented schema (over 70%) are morphological variants (2) for JA and ZH, it usually generate the synonyms.

### 6 Related Work

#### 6.1 Multilingual Text-to-SQL Datasets

The recent development of text-to-SQL is greatly driven by the large-scale annotation datasets. These corpora cover a wide range of settings: single-table (Zhong, Xiong, and Socher 2017), multi-table (Yu et al. 2018), multi-turn (Yu et al. 2019a,b). There are also a few non-English text-to-SQL datasets (Min and Zhang 2019; Tuan Nguyen, Dao, and Nguyen 2020; Guo et al. 2021; Jose and Cozman 2021).

However, all these multilingual text-to-SQL datasets only support three languages. The language coverage is limited compared with other multilingual datasets. For example, the multilingual task-oriented dialogue dataset MTOP (Li et al. 2021) and MultiATIS++ (Xu, Haider, and Mansour 2020) support six languages and nine languages respectively. Therefore, to advance the research on multilingual text-to-SQL, we propose MULTISPIDER covering seven mainstream languages and quite challenging.

#### 6.2 Multilingual Text-to-SQL Systems

Driven by the large-scale English text-to-SQL dataset, many powerful task-specific model have been proposed for text-to-SQL, including effective input encoding (Wang et al. 2020), intermediate representation of SQL (Guo et al. 2019) and grammar-based decoding for valid SQL (Yin and Neubig 2018). Among a wide range of fancy models, RAT-SQL (Wang et al. 2020) is the most popular one which attracts a lot of attention from the research community and industry. Specifically, it adopts the relation-aware transformer to learn the joint representation of database and question, and achieves the promising results.

For non-English text-to-SQL, previous work (Min and Zhang 2019; Tuan Nguyen, Dao, and Nguyen 2020) typically adopts language-specific tokenizer or pretrained language model like PhoBERT for Vietnamese (Nguyen and Nguyen 2020), to extend the English parser for multilingual scenario. Therefore, we adopt the RAT-SQL with multilingual encoder like multilingual-BERT (Devlin et al. 2019) and XLM-R (Conneau et al. 2020) as our main baseline models.

Besides the task-specific approaches, there is also another research trend that using the pretrained encoder-decoder models to track with the text-to-SQL. It attempts to formula the text-to-SQL parsing tasks as seq2seq translation task. Recently, researchers have developed lots of powerful parsers (Scholak, Schucher, and Bahdanau 2021; Shin et al. 2021) built on the top of pretrained language models like BART (Liu et al. 2020) and T5 (Raffel et al. 2020). Thus, we attempt to choose mBART (Liu et al. 2020), a multilingual pretrained encoder-decoder model, as another baseline model.

#### 7 Conclusion and Future Work

Most existing work on text-to-SQL are centered on English, excluding the powerful interaction technique’s accessibility to non-English speakers. In this paper, we present the largest dataset MULTISPIDER covering seven mainstream languages to promote the research on multilingual text-to-SQL. We ensure the dataset quality by hiring sufficient qualified translators and multi-rounds checking. The results MULTISPIDER is natural, accurate and also challenging in terms of text-to-SQL. We further explore the lexical challenge and structural challenge in multilingual text-to-SQL and find that language-specific properties would make these two challenges more difficult. Therefore, we propose a simple and generic schema-augmentation method SAVE to expand the size of training data. Extensive experiments verify the effectiveness of SAVE, which boosts the model performance by about 1.8%. We propose a series of popular baseline methods and conduct extensive experiments on MULTISPIDER to encourage future research for multilingual text-to-SQL systems.

Future work would include (1) developing a multilingual text-to-SQL system and apply it in the real globalization scenario; (2) leveraging better pretrained model and advancing architecture design to address the lexical challenge and structure challenge in multilingual settings; (3) expanding SAVE to other table-related task (Wenhui Chen 2020) and further improve the schema verification accuracy.
Ethical Statement

This work presents MULTISPIDER, a free and open dataset for the research community to study the multilingual text-to-SQL problem. Data in MULTISPIDER are collected from Spider (Yu et al. 2018), a free and open cross-database English text-to-SQL dataset. We also collect data from the CSpi- der (Min and Zhang 2019) and VSpider (Tuan Nguyen, Dao, and Nguyen 2020), which are also free and open text-to-SQL dataset. To annotate the MULTISPIDER, we recruit 15 Chinese college students (8 females and 7 males). Each student is paid 2 yuan ($0.3 USD) for translating the schema or questions. This compensation is determined according to prior work on similar dataset construction (Guo et al. 2021). Since all question sequences are collected against open-access databases, there is no privacy issue. The details of our data collection and characteristics are introduced in Section 2.

Acknowledgements

We thank all anonymous reviewers for their constructive comments. Wanxiang Che was supported via the grant 2020AAG1016501 and NSFC grants 62236004 and 61976072. Dechen Zhan is the corresponding author.

References


Wenhu Chen, e., Hongmin Wang. 2020. TabFact: A Large-scale Dataset for Table-based Fact Verification. In ICLR.


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