Wav2SQL: Direct Generalizable Speech-To-SQL Parsing

Anonymous ACL submission

Abstract

Speech-to-SQL (S2SQL) aims to convert spoken questions into SQL queries given relational databases, which has been traditionally implemented in a cascaded manner while facing the following challenges: 1) model training is faced with the major issue of data scarcity, where limited parallel data is available; and 2) the systems should be robust enough to handle diverse out-of-domain speech samples that differ from the source data. In this work, we propose the first direct speech-to-SQL parsing model Wav2SQL which avoids error compounding across cascaded systems. Specifically, 1) to accelerate speech-driven SQL parsing research in the community, we release a large-scale and multi-accent dataset MASpider; 2) leveraging the recent progress in the large-scale pre-training, we show that it alleviates the data scarcity issue and allow for direct speech-to-SQL parsing; and 3) we include the speech re-programming and gradient reversal classifier techniques to reduce acoustic variance and learned style-agnostic representation, improving generalization to unseen out-of-domain custom data. Experimental results demonstrate that Wav2SQL avoids error compounding and achieves state-of-the-art results by up to 4.1% accuracy improvement over the baseline.

1 Introduction

Speech-to-SQL parsing (S2SQL) aims to generate the SQL query from a spoken question based on relational databases. This technology is highly beneficial as it breaks down barriers among those who lack proficiency in SQL queries and are unable to perform screen inputs while driving or exercising. Furthermore, S2SQL provides flexible and convenient ways of interaction, which opens up a host of practical applications in fields such as vehicle terminals, smart watches, smart speakers, and the medical industry. Conventional S2SQL systems (Kumar et al., 2013; Song et al., 2022) are often composed of a cascade of two components: automatic speech recognition (ASR) (Yu and Deng, 2016; Schneider et al., 2019; Hsu et al., 2021) and text-to-SQL parsing (Bogin et al., 2019b,a; Chen et al., 2020; Guo et al., 2019). Compared to cascaded systems, work on direct S2ST is very limited, with the potential benefits of 1) working on languages without written form (Campbell, 2008), where an estimated half of the 7,000 languages in the world actually do not have written forms; 2) avoiding error compounding across sub-systems (Nakamura et al., 2006; Jia et al., 2019).

The recent development of direct S2SQL parsing still faces several challenges: 1) despite the benefits of direct approaches, model training is faced with the major issue of data scarcity. Human-labeled speech data is expensive to create, there are very few data resources providing parallel speech, and the data amount is quite limited, 2) increasing demand for SQL parsing from personalized speech challenges models especially in unseen scenarios. When the distributions of custom voice (speaker and accent) differ from training data, the system performance deteriorates due to distribution gaps, and 3) the modality gap between the spoken question and text schema hinders the ability of the question schema, making it difficult to align question speech to the intended tables.

To accelerate S2SQL research, we assemble an open-source, multi-speaker, and multi-accent S2SQL corpus MASpider. To the best of our knowledge, MASpider is the first open-source speech-to-SQL parsing dataset. We have attached part of MASpider to the supplementary materials, and we will release the entire dataset after the paper publication. To overcome the aforementioned challenges in this paper, we propose Wav2SQL for direct speech-to-SQL parsing, which is generalizable to unseen acoustic conditions (speaker and accent) in custom data. To be more specific, 1) leveraging self-supervised learning (SSL) (Baevski et al., 2020; Hsu et al., 2021), it alleviates the data scarcity issue and benefits S2SQL model training, 2) we introduce speech re-programming and gradient reverse technique to effectively eliminate the style attributes in representation, which promote the model generalization to unseen speakers and accents in custom data.

Experimental results on the MASpider dataset demonstrate that our Wav2SQL model surpasses the cascaded system in the exact match accuracy and achieves competitive performance with our model trained on the TTS dataset. The main contributions of This work are summarized as follows:

- We introduce the first cross-domain speech-to-SQL parsing benchmark dataset MASpider[1].
- Leveraging self-supervised learning, we propose the first direct speech-to-SQL parsing and show that the large-scale pre-training alleviates the data scarcity issue.
- Through introducing speech reprogramming and gradient reversal technique, we effectively eliminate the style attributes in speech representation and predict the style-agnostic variation, which significantly improves the model generalization to unseen speakers and accents in custom data.
- Experimental results on the MASpider dataset demonstrate that our model outperforms the cascaded systems and achieves state-of-the-art performances.

2 Related Works

2.1 Text-to-SQL Parsing

Semantic parsing of natural language to SQL query recently surged in popularity because of the release of two cross-domain datasets-WikiSQL (Zhong et al., 2017) and Spider (Yu et al., 2018). IR-Net (Guo et al., 2019) encodes the question and schema via bi-LSTM and proposes the string match strategy for schema linking. RATSQL (Wang et al., 2019) presents a unified framework with a relation-aware transformer(RAT) to encode relational databases and NL questions. SADGA (Cai et al., 2021) adopts the graph structure to provide a unified encoding model for both the NL question and databases. In recent years, speech-to-SQL systems usually adopt cascaded automatic speech recognition with text-based SQL parsing. However, the error propagation hurts model performance, not to mention that numerous languages do not have written forms. In this work, we present the first direct speech-to-SQL parsing model without using text, which demonstrates the generalization to different accents and speakers.

2.2 Self-Supervised Learning in Speech

Self-supervised speech representation learning encodes the speech feature into context representations. TERA (Liu et al., 2021) learns speech representation by reconstructing acoustic frames from their altered counterparts. Vq-wav2vec (Baevski et al., 2019) learns discrete representations via a context prediction task using contrastive loss. Similarly, wav2vec 2.0 (Baevski et al., 2020) is an end-to-end version of vq-wav2vec, while HuBERT (Hsu et al., 2021) predicts masked frames pre-quantized using k-means. In this work, we leverage the recent success of self-supervised learning in speech and show that large-scale pre-training alleviates the data scarcity issue and benefits model training.

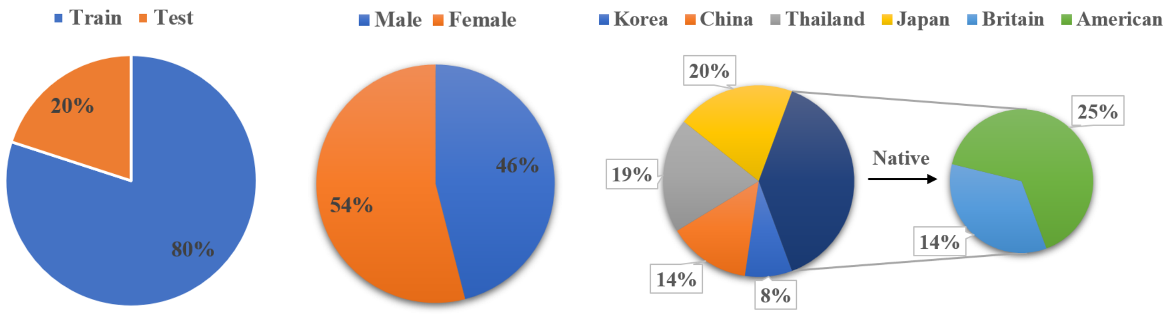
2.3 Domain Generalization

Domain generalization aims to learn domain-invariant knowledge which can be generalized to the target distribution, which attracts attention from researchers (Zhou et al., 2020; Shi et al., 2021; Tian et al., 2022; Huang et al., 2022c). (Li et al., 2018b) propose a conditional invariant adversarial network to learn class-wise adversarial networks and (Zhao et al., 2020) learns domain-invariant features by introducing additional entropy regularization to minimize the KL divergence between the conditional distributions of different source domains. For spoken language understanding, unseen speakers and accents in custom data significantly hurt model performance due to the distribution gaps. In this work, we introduce speech reprogramming and gradient reverse to disentangle semantically irrelevant information, leading to the significant promotion of model generalization to custom scenarios.

3 Dataset Construction

We build MASpider upon the Spider (Yu et al., 2018), which has 8659/1034 train/evaluation splits and an unreleased test set. MASpider consists of 9693 spoken utterances recorded by eleven speakers from six different countries. MASpider consists of 15 hours of speech samples recorded in a profes-

[1] Audio samples are available at https://Wav2SQL.github.io/



(a) Train/Test Split (b) Gender Statistics (c) Visualization of different country distributions

Figure 1: The statistics for MASpider.

sional recording studio, including 8.1 hours from 6 females and 6.9 hours from 5 males apart from the person-of-interest (POI). Figures 1 summarize the distribution of dataset split, gender, and country. More details on MASpider are available in the appendix D. The major features of MASpider include:

- Open source. A lack of data could hinder the construction of speech-to-SQL systems, so we release our corpus to accelerate research in the community.
- Diversity. Since the distribution of custom voice could be different from training data, we construct a dataset with different gender, accent, and language background to improve model generalization.
- High quality. High-quality audios without excessive noise or error annotation are essential for S2SQL training. A strict verification ensures high-quality utterances in MASpider.

3.1 Data Collection and Verification

Collection Procedure For all 9693 utterance-SQL pairs in MASpider, we ensure that each speaker is assigned no more than 1500 sentences to avoid excessive data distribution bias. Next, we collect the audio sample of the given text utterances in a professional recording studio. Finally, the spoken utterances are saved in wav format, sampled at 16kHz, and quantized by 16 bits.

Data Labeling For further study, we tag additional statistics such as the native language and age of the speakers, and the year of their English study. Following this, the dataset is split into 12-hour spoken questions for training, additional 3-hour utterances with unseen accents, speakers, and databases for testing, which enable the evaluation of model generalization to custom data. Figure 1(a) illustrates the distribution of the training and test sets on MASpider.

Data Verification Firstly, we check that the accent in the recording matches the speaker's country. Then, We listen to every recording to check for mispronounced errors and re-record the recording with more than two mispronunciations. Finally, we run the preliminary qualified recordings through an ASR system to control the recorded audio quality. In our case, we used the fine-tuned wav2vec 2.0 ASR model to filter out recordings with their character error rates higher than 25%. For audio with these error rates above the threshold, it is discarded and recollected again until passed.

3.2 Dataset Statistics

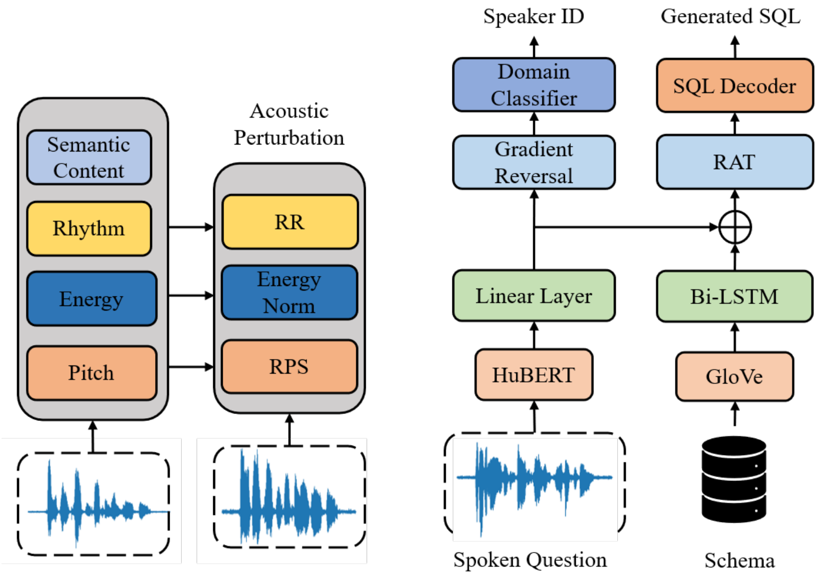
After the data collection and processing procedure, we check for audio quality and conduct the statistical evaluation.

Gender The visualization of gender statistics is displayed as Figure 1(b). As we can see, the ratio of male to female speakers is relatively average.

Country The recorders mainly include 4 English native speakers and 7 non-native speakers from Japan, China, Thailand, and Korea. We count the proportion of utterances recorded by these speakers and visualize it as shown in Figure 1(c)

Group By Difficulty Following the common practice (Yu et al., 2018) to better demonstrate the model performance on different SQL queries, we group the difficulty of each spoken question into 4 levels according to the number of SQL components, selections, and conditions. Specifically, SQL queries that contain more keywords (e.g., GROUP BY, ORDER BY, INTERSECT, etc.) will be considered harder. In the end, The test set of MASpider consists of 25.5% easy, 37.9% medium, 20.9% hard, and 15.7% extra hard SQL queries.

We build MASpider upon the Spider whose



(a) Speech Re-programming (b) Wav2SQL

Figure 2: The information flow with dotted lines is included during training. Subfigure(a) denotes the implementation process of Speech Re-programming RR: random resampling; RPS: a chain function for random pitch shifting of the raw waveform. Subfigure(b) is the overall architecture of our Wav2SQL. RAT: relation-aware transformer.

train/evaluation division is 8569/1034 and the test is not released. For this reason, MASpider consists of 9693 spoken utterances recorded by 11 speakers from six different countries. MASpider consists of 15 hours of speech samples recorded in a professional recording studio, including 8.1 hours from 6 females and 6.9 hours from 5 males apart from the person-of-interest (POI). Figures summarize the distribution of dataset split, gender, and country. More details about MASpider can be found in the appendix.

4 Proposed Method

4.1 Overview

The overall architecture has been presented in Figure 2b. To alleviate the data scarcity issue (Huang et al., 2022a,b), we leverage the large-scale self-supervised models including Hubert (Hsu et al., 2021) for the spoken question and language model for the textual schema to derive discriminative representation, enabling direct speech to SQL parsing. For generalizable speech to SQL parsing, we propose several techniques to promote model robustness for unseen (speaker and accent) custom data: 1) we re-program acoustic attributes and perturb the style information in speech, selectively extracting only the linguistic-related information for domain-agnostic modeling; 2) we include gradient reversal classifier to eliminate speaker information with an auxiliary gradient reversal classifier.

In the end, the tree-structure decoder produces results with an abstract syntax tree (AST) in depth-first traversal order. The training procedures are included in Section 4.5, and more information has been attached in Appendix C.

4.2 Enhanced Speech Encoder

4.2.1 Self-Supervised Pre-training

To alleviate the data scarcity issue (Huang et al., 2023a,b) and learn linguistic content from raw waveform (Huang et al., 2022e,d; Lam et al., 2021), we leverage recent progress in large-scale self-supervised learning with Hubert (Hsu et al., 2021), with a multi-layer convolution waveform encoder to generate the feature sequence followed by a Transformer (Vaswani et al., 2017) context encoder to build the contextualized representations.

We adopt the Hubert-Base model as speech representation, which is pre-trained on 960 hours LibriSpeech (Panayotov et al., 2015). Notably, speech representations (Choi et al., 2021) are found to merge not only rich acoustic information but also acoustic attributes related to accents and speakers. To reduce domain-specific variations for better generalization, we investigate a novel technique in the following parts, which effectively eliminates accent and speaker information in speech representations while preserving linguistic content.

4.2.2 Analysis: Speech Quality Across Layers

Before introducing our techniques for removing accent and speaker information, we first discuss the impact of the selection of different layers of Hubert on the model performance. Similar to natural language understanding, exploring the transformer layers of the BERT model shows that the underlying blocks encode syntactic information, while high-level semantic information appears in higher blocks. To make a more intuitive sense of this, we separately extract frozen representations of Hubert's 12 layers as audio features. We then input these audio features into the S2SQL model and evaluate their performance by exact match accuracy. Figure 3 demonstrates that the first 7 layers as well as the last two layers have poor performance compared to layers 8 to 10 whose accuracy is higher than 39.0 %. Layer 9 achieves the best accuracy of 41.5 %.

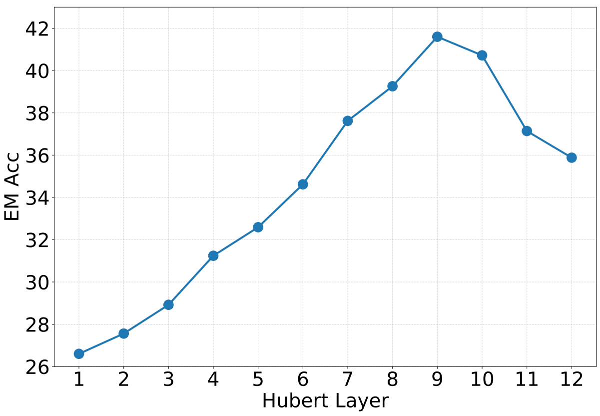


Figure 3: Speech-to-SQL generation using representations from different Hubert layers pre-trained on LibriSpeech. EM Acc: Exact match accuracy.

4.2.3 Re-program on Acoustic Condition

An intuitive way (Li et al., 2018a; Bui et al., 2021) to achieve better generalization is to decompose a model into the domain-agnostic and domain-specific parts via disentangled representation learning and eliminate the domain-specific variations.

In contrast, the representation derived from self-supervised models contains not only rich linguistic content but also information related to pitch and speaker, which are style-specific attributes that may decrease model generalization. As such, we conduct re-programming on speech attributes and perturb the rhythm, pitch, and energy information, which disentangle acoustic variation and selectively extract only the linguistic-related information, exhibiting better generalization to unseen custom data. As shown in Figure 2a, we apply bottlenecks on acoustic conditions and create re-programmed speech samples. Additional details have been attached in Appendix B.

Rhythm Rhythm characterizes how fast the speaker utters each syllable. To perturb rhythm information, we adopt random resampling RR to divide the input into segments of random lengths, and we randomly stretch or squeeze each piece along the time dimension.

Pitch Pitch is an indispensable component of intonation. First, We normalize the pitch contour to a common mean and standard deviation, removing the timbre variations in speech. Secondly, a chain function is adopted to randomly shift the pitch contour.

Energy Energy represents the magnitude of the raw waveforms and visually reflects the volume of the speech. We re-program energy attributes and create samples with different energy distributions.

4.3 Gradient Reversal Classifier

To eliminate the speaker identity in speech representation, we introduce a gradient reversal layer (GRL) in speaker classifier (Ganin et al., 2016), which regards speaker variations as a classification problem and directly maximizes the loss of the domain classifier by reversing its gradients. In backpropagation, GRL takes the gradient from the subsequent layer and changes its sign by multiplying with -1 before passing it to the preceding layer:

$$R(x) = x, \frac{dR}{dx} = -\mathbf{I}, \quad (1)$$

where I denotes an identity matrix.

It ensures that the feature distributions between different speakers are similar (i.e. as distinguishable as possible), resulting in speaker-agnostic features. Therefore, we can further obtain audio features that preserve semantics regardless of accent and speaker, demonstrating better generalization to custom data in SQL decoding.

4.4 SQL Decoder

The SQL decoder follows the grammar-based architecture of Yin and Neubig (2017), which generates the SQL as an abstract syntax tree(AST) in depth-first traversal order. The generation process of SQL AST is factorized into sequential actions, which are divided into two cases: (1) APPLYRULE which expands the last generated node according to the grammar rules or completes a leaf node, (2) SELECTCOLUMN and SELECTTABLE represent that selects a column or table item from the schema respectively.

Firstly, the probability of generating a SQL y is defined as:

$$p(y|x) = \prod_t p(a_t|x, a_{<t})$$

where x is the encoded memory of questions, columns, and tables, a_t is the action token at time step t, and $a_{<t}$ is the sequential actions before time step t. Then in the tree-structured LSTM decoder, the hidden states at each time step t are updated as $m_t, h_t = \mathrm{LSTM}([a_{t-1}; p_t; c_t; n_t], m_{t-1}, h_{t-1})$, where m_t is the cell state of time step t, h_t is the hidden state, a_{t-1} is the previous action embedding, p_t is the parent information of the current node, c_t is the context vector, and n_t is the embedding of the current node type. Finally, how action probabilities $p(a_t|x, a_{<t}$ are computed are explained as follows: For APPLYRULE action,

$$p(a_t = \mathrm{AR}[r]|x, a_{<t}) = \mathrm{softmax}_R(g(h_t)) \quad (2)$$

where AR is the APPLYRULE action and $g(\cdot)$ is the feed-forward network that is composed of two linear layers and a $tanh$ activation function.
For SELECTTABLE action,

$$\gamma_j = softmax_j(\frac{(h_t W_Q)(x_j W_K + \mathbf{R}_{ij})^T)}{\sqrt{d}}), \quad (3)$$

$$p(a_t = \mathrm{ST}[i]|x, a_{<t}) = \sum_j \gamma_j \quad (4)$$

where ST denotes SELECTTABLE action. The calculation of SELECTCOLUMN action is similar.

4.5 Training and Inference

We formulate speech-to-SQL parsing as a sequence-to-tree generation problem. The input is the question (audio) and schema (text), which belong to two different modalities, while the output is the SQL query. We adopt the pre-trained self-supervised speech representation model Hubert (Hsu et al., 2021) and language model GloVe (Pennington et al., 2014) as the backbone of our model.

Training. The final loss terms in training are composed of the two parts:1) domain classification loss $\mathcal{L}_{CE}$: cross-entropy loss between the predicted speaker ID and the ground-truth; 2) SQL generation loss $\mathcal{L}_{MLE}$: maximum likelihood estimation(MLE) based on the given SQL query to maximize the predicted probability $p(y|x, a_{<t})$ based on a given SQL query. Note that, the domain classification loss $\mathcal{L}_{CE}$ is trained to remove speaker information but preserve semantic information, which is helpful for the final objective $\mathcal{L}_{MLE}$ to generate more accurate SQL query and the loss weight of $\mathcal{L}_{CE}$ is set to be 0.01.

Inference. After training, for each pair of the spoken question and database schema, we generate the target SQL query according to grammar rules with heuristics decode. We replace the special tokens in the target sequences with the SQL keywords.

5 Experiments

5.1 Experimental Setup

Evaluation Metrics Following the common practice (Yu et al., 2018), we evaluate the performance by exact match accuracy and component matching accuracy provided by (Yu et al., 2018), where exact match accuracy measures whether the predicted query is equivalent to the gold query as a whole while component matching measures the average exact match between the prediction and ground truth on different SQL components.

Training and Inference We train our model on a single 82G NVIDIA A100 GPU with a batch size of 20 for 100 epochs using the AdamW optimizer. The learning rate is $5e-4$ and the weight decay coefficient is $1e-4$. We preprocess column names and table names for tokenization and lemmatization using Stanza toolkit (Qi et al., 2020). In inference, we adopt beam search decoding with beam size 5.

Model Configurations In the encoder, the hidden size is 256 and the number of RAT layers is 8. In the SQL decoder, we set the rule embedding size as 128 and the node type embedding size as 128. Following (Huang et al., 2022e; Lee et al., 2021), the ASR model in our work is Wav2vec 2.0 Large(LV-60) + Self Training / 960 hours / Libri-Light + LibriSpeech [2]. A comprehensive table of hyperparameters is available in Appendix C in the supplementary materials.

Baseline models We compare generated SQL queries of our Wav2SQL with other systems, including: (1) Cascaded: the cascaded model composed of automatic speech recognition(ASR) and text-to-SQL parsing model, which adopts the wav2vec 2.0 (Baevski et al., 2020) and RAT-SQL (Wang et al., 2019). (2) S2SQL-TTS: the S2SQL model trained on the dataset synthesized as our upper bound, where S2SQL means the model

[2] https://github.com/facebookresearch/fairseq/tree/main/examples/wav2vec

Method	SELECT	WHERE	GROUP	ORDER	AND/OR	KEYWORD	Exact Match
Model Performance							
S2SQL-TTS	71.2	55.0	67.1	66.1	95.4	77.2	44.8
WhisperSQL	65.2	46.3	58.0	62.7	94.8	77.2	24.6
Cascaded	66.3	49.3	62.6	62.6	94.7	72.5	38.2
Wav2SQL	**69.7**	**51.9**	**63.6**	**69.3**	**95.0**	**77.4**	**42.3**
Generalization to Custom Data							
S2SQL-TTS	70.9	54.9	56.5	69.8	94.7	75.3	40.8
WhisperSQL	66.2	41.4	52.0	65.7	94.5	**73.2**	20.9
Cascaded	64.0	42.3	55.2	**66.6**	**95.9**	72.2	32.3
Wav2SQL	**68.8**	**47.1**	**57.7**	59.0	94.2	67.7	**35.3**
Ablation Study							
w/o Speech reprogramming	65.3	45.2	56.9	58.3	94.4	67.7	33.9
w/o Gradient reversal classifier	66.1	43.8	48.8	58.8	94.3	67.5	33.2
w Hubert Only	65.3	44.8	56.1	53.9	93.7	66.9	31.8

Table 1: partial matching accuracy and exact match accuracy on the MASpider test set comparison with baseline systems. We adopt Hubert as the speech feature extractor and GloVe as the language model.

after removing the speech reprogramming, and adversarial learning in our Wav2SQL. The TTS model we adopt here is FastSpeech 2 (Ren et al., 2020). (3) WhisperSQL: This baseline employs Whisper-Base (Radford et al., 2022), the state-of-the-art model of ASR, as the speech encoder instead of Hubert.

5.2 Model Performance

For in-domain evaluation, we prepare spoken questions with seen accents and speakers according to different SQL components, including SELECT, WHERE, GROUP, ORDER, AND/OR, and KEYWORD following (Yu et al., 2018). The results are compiled and presented in Table 1, and we have the following observations: Wav2SQL surpasses the cascaded system across all SQL component matching and exact match accuracy on all SQL queries. Specifically, the SELECT and ORDER component has increased significantly by 3.4% and 6.7% respectively, and exact match accuracy has increased by 4.1%, demonstrating the effectiveness of our direct speech-to-SQL parsing model. It indicates that our direct S2SQL model avoids error compounding across subsystems. Besides, Wav2SQL greatly surpasses WhisperSQL which proves the superiority of Hubert. Compare to the upper bound less variance dataset constructed by a single-speaker single-accent TTS system, we still achieve competitive performance, indicating the efficiency of our proposed techniques for reducing acoustic attributes and promoting generalization.

To further verify the effectiveness of our methods, we compare our model with the cascaded system and WhisperSQL model. We group the parsing difficulty into easy, medium, hard, and extra according to the number of component selections and conditions of the target SQL queries. As illustrated in Table 2, we have the following observations:

1) As the parsing difficulty increases, a distinct degradation could be witnessed in generation accuracy; and 2) our direct speech-to-SQL parsing model outperforms the cascade baseline since it avoids error compounding across subsystems, demonstrating a large margin improvement by 10.2% in the extra hard part; and 3) Our model is far superior to WhisperSQL at all levels, indicating that the self-supervised Hubert model is more suitable for our task than the ASR Encoder.

Dataset	Easy	Medium	Hard	Extra
WhisperSQL	44.8	22.4	17.8	7.2
Cascaded	**64.5**	33.9	29.9	19.3
Wav2SQL	63.9	**38.1**	**34.5**	**29.5**

Table 2: A comparison to the cascaded model and WhisperSQL model in-domain setting according to the level of difficulty.

5.3 Generalization To Custom Data

For out-of-domain testing, we prepare spoken questions with databases, accents, and speakers that are unseen in custom data. The results are summarized in Table 1, and we have the following observations: 1) As shown in the table, we see that our proposed Wav2SQL outperforms WhisperSQL by a large margin of 14.4% on exact match accuracy. In addition, the component matching of our model on all SQL components outperforms WhisperSQL, especially in WHERE and GROUP components by both 5.7%; 2) Under the challenge of invisible

Medium	Show name, country, age for all singers ordered by age from the oldest to the youngest.
Cascaded	SELECT singer.Country, singer.Age FROM singer ORDER BY singer.Age Desc
Wav2SQL	SELECT singer.Name, singer.Country, singer.Age FROM singer ORDER BY singer.Age Desc
Gold SQL	SELECT Name, Country, Age FROM singer ORDER BY Age Desc
Hard	List all song names by singers above the average age.
Cascaded	SELECT singer.Song_Name FROM singer WHERE singer.Age < 'terminal' ORDER BY singer.Song_Name Asc
Wav2SQL	SELECT singer.Song_Name FROM singer WHERE singer.Age > (SELECT Avg(singer.Age) FROM singer
Gold SQL	SELECT Song_Name FROM singer WHERE Age > (SELECT avg(Age) FROM singer)
Extra	Find the average age of students who do not have any pet.
Cascaded	SELECT Student.Fname FROM Student WHERE Student.StuID NOT IN (SELECT Has_Pet.StuID FROM Has_Pet)
Wav2SQL	SELECT Avg(Student.Age) FROM Student WHERE Student.StuID NOT IN (SELECT Has_Pet.StuID FROM Has_Pet)
Gold SQL	SELECT avg(Age) from Student where stuid not in (select stuid from has_pet)

Table 3: Three examples compared with the cascaded system. We mark the wrong part of the cascaded model in red while the corresponding correct part in Wav2SQL is in blue. The input question is represented by SQL query difficulty. Cell values in the SQL queries are replaced with placeholder "terminal".

accents, Wav2SQL can still achieve better performance with a 3.0% exact match accuracy increase compared with the cascaded system, which validates the superiority of our model by exploiting speech re-program and adversarial training to get deterministic representations invariant to accents and speakers; 3) Although we are pleasantly surprised to find that Wav2SQL maintains comparative results with S2SQL-TTS in SELECT, GROUP BY and AND/OR component accuracy, there is still a certain gap with it in exact match accuracy due to the limited acoustic information in the TTS dataset.

5.4 Ablation Studies

As shown in Table 1, we conduct ablation studies to demonstrate the effectiveness of several designs in our model, including speech re-programming and gradient reversal classifier technique. The results have been presented in Table 1, and we have the following discovering: 1) the removal of the speech re-programming method shows an improvement in exact match accuracy by 1.4% and a significant increase of 3.5% in SELECT component matching, indicating its efficiency in reducing acoustic variance and learning deterministic representations; 2) Removing the gradient reversal classifier has witnessed a distinct degradation in model performance by 2.1% accuracy especially in GROUP component matching (8.9%), showing its superiority in learning speaker-agnostic speech representation; 3) Keeping only the Hubert module(i.e. removing both the speech re-programming and gradient reversal classifier) results in a significant performance decrease compared to adding each of them. This once again proves that both of the methods we propose are able to effectively preserve only semantic information in audio to improve model performance.

5.5 Case Study

We compare the SQL query generated by Wav2SQL with the cascaded system in Table 3. The results demonstrate that Wav2SQL outperforms the baseline system. For example, in the first and third cases, the cascaded fails to fill the values into the correct slots, thus, it stupidly forgets the 'Name' of table 'Singer' and is unable to select the correct column name 'Age'. In addition, Wav2SQL successfully completes the averaging operation on "Age" in the third case. Unfortunately, in the second example, the cascaded system incorrectly constructs the WHERE clause so that it fails to pick singers who are older than the average age.

6 Conclusion

We released MASpider, the first large-scale, multi-speaker, and multi-accent S2SQL parsing dataset, which we hope would accelerate S2SQL research in the community. In this work, we presented the first direct speech-to-SQL model Wav2SQL which avoided error compounding across cascaded systems. To tackle the data scarcity issue, we leveraged recent progress in large-scale pre-training and utilized self-supervised models to derive discriminate representation. To promote model generalization and robustness to custom out-of-distribution data, we further introduced speech re-programming and gradient-reversal classifier techniques which reduced acoustic variance and learned style-agnostic representations. Experimental results demonstrated that our approach achieved new state-of-the-art results by up to 4.1% accuracy improvement over baseline. In the future, we will investigate techniques to further enhance the model generalization in direct Speech-to-SQL parsing.

7 Limitation and Potential Risks

As mentioned in the model performance, there is still a certain gap between Wav2SQL and S2SQL-TTS. One of our future directions is to further remove accent and speaker information to improve generation performance. In addition, our experiments find that the schema linking we adopt is still rough compared to text schema linking, which seriously affects the performance of our model. In future work, we will study how to obtain accurate and fine-grained schema linking.

Wav2SQL lowers the requirements for speech-to-SQL generation, which may cause unemployment for people with related occupations database developers, and SQL programmers. Furthermore, there is the potential for leading to the misuse of databases than they expect.

References

Alexei Baevski, Steffen Schneider, and Michael Auli. 2019. vq-wav2vec: Self-supervised learning of discrete speech representations. *arXiv preprint arXiv:1910.05453*.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460.

Ben Bogin, Matt Gardner, and Jonathan Berant. 2019a. Global reasoning over database structures for text-to-sql parsing. *arXiv preprint arXiv:1908.11214*.

Ben Bogin, Matt Gardner, and Jonathan Berant. 2019b. Representing schema structure with graph neural networks for text-to-sql parsing. *arXiv preprint arXiv:1905.06241*.

Manh-Ha Bui, Toan Tran, Anh Tran, and Dinh Phung. 2021. Exploiting domain-specific features to enhance domain generalization. *Advances in Neural Information Processing Systems*, 34:21189–21201.

Ruichu Cai, Jinjie Yuan, Boyan Xu, and Zhifeng Hao. 2021. Sadga: Structure-aware dual graph aggregation network for text-to-sql. *Advances in Neural Information Processing Systems*, 34:7664–7676.

Lyle Campbell. 2008. Ethnologue: Languages of the world.

Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. 2020. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3438–3445.

Hyeong-Seok Choi, Juheon Lee, Wansoo Kim, Jie Lee, Hoon Heo, and Kyogu Lee. 2021. Neural analysis and synthesis: Reconstructing speech from self-supervised representations. *Advances in Neural Information Processing Systems*, 34:16251–16265.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030.

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. *arXiv preprint arXiv:1905.08205*.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460.

Rongjie Huang, Chenye Cui, Feiyang Chen, Yi Ren, Jinglin Liu, Zhou Zhao, Baoxing Huai, and Zhefeng Wang. 2022a. Singgan: Generative adversarial network for high-fidelity singing voice generation. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 2525–2535.

Rongjie Huang, Jiawei Huang, Dongchao Yang, Yi Ren, Luping Liu, Mingze Li, Zhenhui Ye, Jinglin Liu, Xiang Yin, and Zhou Zhao. 2023a. Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models. *arXiv preprint arXiv:2301.12661*.

Rongjie Huang, Max WY Lam, Jun Wang, Dan Su, Dong Yu, Yi Ren, and Zhou Zhao. 2022b. Fastdiff: A fast conditional diffusion model for high-quality speech synthesis. *arXiv preprint arXiv:2204.09934*.

Rongjie Huang, Mingze Li, Dongchao Yang, Jiatong Shi, Xuankai Chang, Zhenhui Ye, Yuning Wu, Zhiqing Hong, Jiawei Huang, Jinglin Liu, et al. 2023b. Audiogpt: Understanding and generating speech, music, sound, and talking head. *arXiv preprint arXiv:2304.12995*.

Rongjie Huang, Yi Ren, Jinglin Liu, Chenye Cui, and Zhou Zhao. 2022c. Generspeech: Towards style transfer for generalizable out-of-domain text-to-speech synthesis. *arXiv preprint arXiv:2205.07211*.

Rongjie Huang, Zhou Zhao, Huadai Liu, Jinglin Liu, Chenye Cui, and Yi Ren. 2022d. Prodiff: Progressive fast diffusion model for high-quality text-to-speech. *arXiv preprint arXiv:2207.06389*.

Rongjie Huang, Zhou Zhao, Jinglin Liu, Huadai Liu, Yi Ren, Lichao Zhang, and Jinzheng He. 2022e. Transpeech: Speech-to-speech translation with bilateral perturbation. *arXiv preprint arXiv:2205.12523*.

Ye Jia, Ron J Weiss, Fadi Biadsy, Wolfgang Macherey, Melvin Johnson, Zhifeng Chen, and Yonghui Wu. 2019. Direct speech-to-speech translation with a sequence-to-sequence model. *arXiv preprint arXiv:1904.06037*.

Sachin Kumar, Ashish Kumar, Pinaki Mitra, and Girish Sundaram. 2013. System and methods for converting speech to sql. *arXiv preprint arXiv:1308.3106*.

Max WY Lam, Jun Wang, Rongjie Huang, Dan Su, and Dong Yu. 2021. Bilateral denoising diffusion models. *arXiv preprint arXiv:2108.11514*.

Ann Lee, Peng-Jen Chen, Changhan Wang, Jiatao Gu, Xutai Ma, Adam Polyak, Yossi Adi, Qing He, Yun Tang, Juan Pino, et al. 2021. Direct speech-to-speech translation with discrete units. *arXiv preprint arXiv:2107.05604*.

Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. 2018a. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5400–5409.

Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. 2018b. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 624–639.

Andy T Liu, Shang-Wen Li, and Hung-yi Lee. 2021. Tera: Self-supervised learning of transformer encoder representation for speech. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:2351–2366.

Satoshi Nakamura, Konstantin Markov, Hiromi Nakaiwa, Gen-ichiro Kikui, Hisashi Kawai, Takatoshi Jitsuhiro, J-S Zhang, Hirofumi Yamamoto, Eiichiro Sumita, and Seiichi Yamamoto. 2006. The atr multilingual speech-to-speech translation system. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(2):365–376.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Adam Polyak and Lior Wolf. 2019. Attention-based wavenet autoencoder for universal voice conversion. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6800–6804. IEEE.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages.

Kaizhi Qian, Yang Zhang, Shiyu Chang, Mark Hasegawa-Johnson, and David Cox. 2020. Unsupervised speech decomposition via triple information bottleneck. In *International Conference on Machine Learning*, pages 7836–7846. PMLR.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. *arXiv preprint arXiv:2212.04356*.

Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2020. Fastspeech 2: Fast and high-quality end-to-end text to speech. *arXiv preprint arXiv:2006.04558*.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. 2019. wav2vec: Unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862*.

Yuge Shi, Jeffrey Seely, Philip HS Torr, N Siddharth, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. 2021. Gradient matching for domain generalization. *arXiv preprint arXiv:2104.09937*.

Yuanfeng Song, Raymond Chi-Wing Wong, Xuefang Zhao, and Di Jiang. 2022. Speech-to-sql: Towards speech-driven sql query generation from natural language question. *arXiv preprint arXiv:2201.01209*.

Chris Xing Tian, Haoliang Li, Xiaofei Xie, Yang Liu, and Shiqi Wang. 2022. Neuron coverage-guided domain generalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

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*.

Pengcheng Yin and Graham Neubig. 2017. A syntactic neural model for general-purpose code generation. *arXiv preprint arXiv:1704.01696*.

Dong Yu and Li Deng. 2016. *Automatic speech recognition*, volume 1. Springer.

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

Shanshan Zhao, Mingming Gong, Tongliang Liu, Huan Fu, and Dacheng Tao. 2020. Domain generalization via entropy regularization. *Advances in Neural Information Processing Systems*, 33:16096–16107.

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

Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. 2020. Deep domain-adversarial image generation for domain generalisation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13025–13032.

A Domain Classifier

Domain classifier effectively captures the audio's long-term speaker identity and predicts the speaker ID for the spoken question. After training on augmented data, the domain classifier could attain robust representations that capture an ample speaker identity space. Combined with gradient reversal, we can get deterministic representation agnostic to speaker discrepancy, significantly reducing intro-speaker variance and making it possible for tree-structured depth-first decoding.

B Acoustic Perturbation

To obtain speech samples with acoustic information enhancement, we adopt the following functions (Qian et al., 2020; Choi et al., 2021) to perturb the acoustic features, that is 1) random resampling RR, and 2) formant shifting fs, and 3)pitch randomization pr, 4) random frequency shaping using a parametric equalizer peq. Next, we feed augmented audios into the model along with original audios.

- For RR, a random resampling is adopted to modify the rhythm. The raw waveform is divided into segments, whose length is randomly uniformly drawn from 19 frames to 32 frames (Polyak and Wolf, 2019). Each segment is resampled using linear interpolation with a resampling factor randomly drawn from 0.5 to 1.5.
- For fs, a formant shifting ratio is sampled uniformly from $\mathrm{Uniform}(1, 1.4)$. After sampling the ratio, we again randomly decided whether to take the reciprocal of the sampled ratio or not.
- For pr, a pitch shift ratio, and a pitch range ratio are sampled uniformly from $\mathrm{Uniform}(1, 2)$ and $\mathrm{Uniform}(1, 1.5)$, respectively. Again, we randomly decide whether to take the reciprocal of the sampled ratios or not. For more details on formant shifting and pitch randomization, please refer to Parselmouth `https://github.com/YannickJadoul/Parselmouth`.
- Lastly, peq denotes a serial composition of low-shelving, peaking, and high-shelving filters. We use one low-shelving HLS, one high-shelving HHS, and eight peaking filters HPeak.

C Model Architectures

We list the model hyperparameters of Wav2SQL in Table 4 and illustrate the architecture for the relational-aware transformer(RAT), SQL decoder, and domain classifier in Figure 4. The schema linking used by RAT in the train set is borrowed from RATSQL(Wang et al., 2019) while the schema linking of the test set comes from string matching between the ASR text and the schema. The ASR text is obtained through Whisper. The reason why wav2vec2.0 is not selected here is that the numbers generated by it are in English and cannot be matched correctly.

Hyperparameter		Wav2SQL
Speech Encoder	Hubert Hidden	768
	Linear Size	256
Text Encoder	GloVe Embedding	300
	LSTM Hidden	256
	LSTM Layers	1
Domain Classifier	Scale Factor	0.1
	Clipping Bounds	10
	Output Dimension	11
Transformer	Transformer Block	8
	Hidden Size	256
	Attention Heads	8
	FFN Size	1024
	Dropout	0.2
SQL Decoder	Action Embedding	128
	Node Embedding	128
	LSTM Size	512
	Dropout	0.2

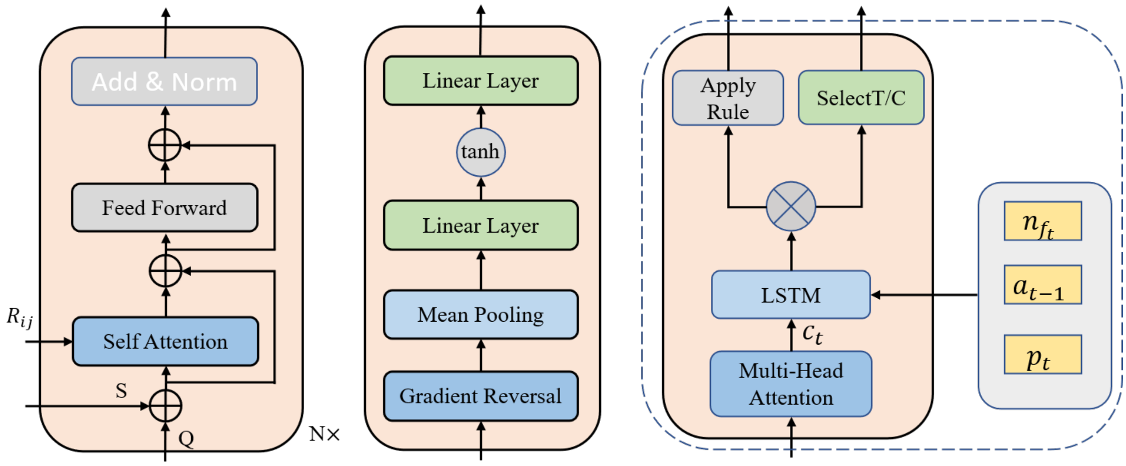
Table 4: Hyperparameters of Wav2SQL.

Hubert The Hubert feature extractor consists of seven blocks and the temporal convolutions in each block have 512 channels with strides (5,2,2,2,2,2,2) and kernel widths (10,3,3,3,3,2,2), and 12 transformer blocks, model dimension 768, inner dimension (FFN) 3,072 and 8 attention heads.

Relation-Aware Transformer The relation-aware encoder consists of 8 transformer layers. Each layer contains a relation-aware self-attention module, the final output passes through a feed-forward layer.

D Dataset Annotation

We outsource the hiring of annotators and handle the data verification process internally. For the verification of the ASR model, we also utilize Wav2vec 2.0 Large(LV-60) + Self Training / 960 hours / Libri-Light + LibriSpeech. Each sentence is recorded by a single speaker, with a minimum, maximum, and average number of recorded utterances per speaker at 747, 1232, and 881 respectively.



(a) Relation-Aware Transformer (b) Domain Classifier (c) SQL Decoder

Figure 4: Architecture for relation-aware transformer, SQL decoder, and domain classifier. The self-attention hear is relation-aware. Q: spoken question; S: schema; R_ij: relations from any question item or schema item; SelectT/C: SELECTTABLE/SELECTCOLUMN.

In-domain and Out-of-domain Test Set The division of in-domain is based on the Spider, resulting in 8659/1034 train/test sets, where speakers and accents are seen during training. Conversely, in the custom out-of-domain split, neither the speaker nor the accent is visible during training, and the train/test split is 8001/1692. It is worth noting that the databases used in these two divisions are invisible.

E Ethical Considerations

Our MASpider benchmark presented in this work is a free and open source for the community to study speech-to-SQL parsing. We collect and annotate recordings from the mainstream text-to-SQL dataset, Spider (Yu et al., 2018), which is also a free and open dataset for research use. For audio recording, we hire annotators from different countries to record audio in a quiet environment. We pay the annotators an average of 80 dollars per hour.