Pay More Attention to History: A Context Modeling Strategy for Conversational Text-to-SQL

Yuntao Li¹, Hanchu Zhang², Yutian Li², Sirui Wang², Wei Wu², Yan Zhang¹

¹Peking University ²Meituan

{ltyt, zhyzhy001}@pku.edu.cn, {zhanghanchu, liyutian, wangsirui, wuwei30}@meituan.com

Abstract

Conversational text-to-SQL aims at converting multi-turn natural language queries into their corresponding SQL representations. One of the most intractable problems of conversational text-to-SQL is modeling the semantics of multi-turn queries and gathering proper information required for the current query. This paper shows that explicit modeling the semantic changes by adding each turn and the summarization of the whole context can bring better performance on converting conversational queries into SQLs. In particular, we propose two conversational modeling tasks in both turn grain and conversation grain. These two tasks simply work as auxiliary training tasks to help with multi-turn conversational semantic parsing. We conducted empirical studies and achieve new state-of-the-art results on large-scale open-domain conversational text-to-SQL dataset. The results demonstrate that the proposed mechanism significantly improves the performance of multi-turn semantic parsing. ¹

Index Terms: conversational text-to-SQL, human-computer interaction, computational paralinguistics

1. Introduction

Semantic parsing is a task that maps natural language queries into corresponding machine-executable logical forms. Being one of the most popular branches of semantic parsing, text-to-SQL, which relieves real users from the burden of learning about techniques behind the queries, has drawn quantities of attention in the field of natural language processing. Existing work mainly focused on converting individual utterances into SQL queries. However, in real scenarios, users tend to interact with systems through conversations to acquire information, in which the conversation context should be considered. To solve this demand of users, the attention of researches on single-turn text-to-SQL shifted to conversational text-to-SQL.

Conversational text-to-SQL is an extension of the standard text-to-SQL task, which frees the restriction of natural language queries from single-turn settings into multi-turn settings. Recent studies [1] [2] [3] show that conversational text-to-SQL shows much higher difficulty compared with single-turn text-to-SQL, especially on modeling multi-turn natural language queries. Figure 1 shows an example of conversational semantic parsing. Three utterances appear in this conversation. The second query is asked according to the first query, and the SQL of the second turn is an modification of the first one by adding an additional restriction. The third query changes the selected columns based on the second query, which in result modifies the selected columns of the SQL.

1 To help with reproduce our results, our code will be made public available till the acceptance of this paper.

It can be observed from the example that to better understanding a contextual query and generating a corresponding SQL, modeling both the semantics changes by adding each separate turn is essential, as well as mapping those changes into the SQL operations. On the one hand, modeling the semantic changes by adding each single turn is conducive to better understanding the semantic flow during a conversation and thus helps to better summarizing them into a single SQL. On the other hand, in order to generate correct predicted SQLs, it is essential to correlate those semantic changes with database schema.

Motivated by these observations, in this paper, we propose RAT-SQL-TC, which uses two auxiliary tasks to help with better modeling multi-turn conversational context and generating correct SQL representations based on RAT-SQL [4]. The first task is Turn Switch Prediction (TSP), which predicts how SQL changes while adding a new turn during a conversation goes. And the second task is Contextual Schema Prediction (CSP), which helps with mapping the contextual changes to database schema operations. CSP requires the utterance encoder model to predict the changes of usage of each column w.r.t the current turn of a conversation. CSP also enhance the encoder model to make a better understanding of database schemas. These two tasks work as auxiliary tasks of multi-task learning that are trained together with the SQL generation task. Our proposed two tasks work from a natural-language-understanding perspective and a database-schema-aware perspective respectively, to enhance the understanding of conversation context and further promote text-to-SQL generation.

We evaluate our proposed method on a popular large-scale cross-domain conversational text-to-SQL benchmarks, i.e., SPARC [5]. By adding our mechanisms, the accuracy on both query match and interaction match is significantly improved against baseline methods. We also achieve new state-
of-the-art results on the leaderboard at the time of writing this paper.

Our proposed mechanisms show advantages in the following aspects. (1) TSP and CSP works from a natural-language-understanding perspective and a database-schema-aware perspective on better modeling conversational context. (2) Our proposed method works as auxiliary tasks of multi-task learning, which avoids troublesome synthetic conversational data collection and extensive computational costs compared with pre-training methods. (3) We boosts baseline method significantly and achieves new state-of-the-art results on a large-scale cross-domain benchmarks.

2. Related Work

2.1. Semantic Parsing and Text-to-SQL
Semantic parsing has been studied for a long period. Previous semantic parsers are generally based on either expert-designed rules [5] or statistical techniques [6][7][8]. In the recent years, neural semantic parsers come to the fore. Neural semantic parsers generally treat semantic parsing as a sequence-to-sequence task, and solve it with encoder-decoder framework [11][12][13][14][15].

Text-to-SQL takes a large share of all semantic parsing tasks. Previous text-to-SQL task mainly focus on relative-simple in-domain text-to-SQL task, and state-of-the-art models show promising performance in this scenario [16][17][18]. Recently, a cross-domain multi-table text-to-SQL dataset called Spider is proposed [19]. Compared with in-domain text-to-SQL, cross-domain multi-table text-to-SQL requires models for higher ability of generalization on both natural language and database schema understanding. On better solving this task, besides pure sequence-to-sequence methods, a new skeleton-then-detail paradigm is proposed and widely applied. This paradigm generates a SQL skeleton first and then fills the skeleton with database schema tokens. Models belong to this paradigm includes SQLNet [20], TypeSQL [21], SQLova [22], Coarse2Fine [23], XSQL [24], HydraNet [25], etc. Besides, some other strategies are proposed for enhancing text-to-SQL parsers, including intermediate representation enhancement [26][27][28], reasoning through GNN model [29][30][31], and data augmentation [32][33].

2.2. Conversational Text-to-SQL

Compared with single-turn text-to-SQL, conversational text-to-SQL requires semantic parsers to understand context of conversations to make correct SQL predictions. More recently, two large-scale cross-domain benchmarks for conversational text-to-SQL (i.e., SParC and CoSQL [1, 35]) are constructed, two large-scale cross-domain benchmarks for conversational conversations to make correct SQL predictions. This method tend to fail when users ask for a new question less related with conversa-

3. Problem Formalization

Conversational text-to-SQL is a task that maps multi-turn natural language queries $u = [u_1, u_2, \ldots , u_T]$ into corresponding SQL logical forms $y = [y_1, y_2, \ldots , y_T]$ w.r.t a pre-defined database schema $s$, where $T$ is the number of turns of a conversa-

4. Methodology

In this paper, we propose RAT-SQL-TC for conversational text-
to-SQL, which adds two auxiliary tasks into the widely applied 

4.1. Overview of RAT-SQL-TC

RAT-SQL is one of the state-of-the-art neural semantic parsers in recent years [4]. RAT-SQL is a unified framework which encodes both relational structure in the database schema and the given question for SQL generation. We take the RAT-SQL as the basis to build our model. Concretely, we use a relation-

$a_bcer$
where \( y = [y_1, \ldots, y|Y|] \) is the ground-truth label of the AST during decoding.

Besides decoding SQL AST, we add two auxiliary tasks to help the model better modeling contextual information and relation to database schema during a conversation. The first one is a Turn Switch Prediction (TSP) task, which requires the encoder model to tell how semantics changes by adding each turn of utterance. The second one is a Contextual Schema Prediction (CSP) task that enforces the model to map those semantics changes to database schema. Loses of these two auxiliary tasks is computed according to the encoding vectors and is optimized simultaneously with the SQL decoding loss.

### 4.2. Turn Switch Prediction

Turn Switch Prediction (TSP) task aims at enhancing the encoder model on understanding the conversation flow between each pair of adjacent queries. This task requires the encoder model to predict whether a type of modification is made on the SQL by adding a new turn of utterance. A total number of \( N_T = 17 \) types of operations are defined, e.g., changing aggregate operation of selection (SELECT sales \( \rightarrow \) SELECT count(sales)) and adding new condition in condition clause (None \( \rightarrow \) WHERE sales > 100). For each type of operation, we make a binary classification on whether such a change is made.

Notate \( t_i \) as the encoding vector of the special token “(s)” of the i-th turn. We use both \( t_i \) and \( t_{i-1} \) to predict whether a type of modification is made. And the TSP loss is a summation of that of all modification types between every adjacent utterance pairs.

\[
s_i = [t_{i-1}; t_i; t_i - t_{i-1}; t_{i-1} \ast t_i],
\]

\[
p_i = \text{Sigmoid} \left( W_{TSP}^i(s_i) \right),
\]

\[
L_{TSP} = \sum_{n=1}^{N_T} \sum_{i=1}^{T} \left( \bar{q}_i^j \log p_i^j + (1 - \bar{q}_i^j) \log(1 - p_i^j) \right) .
\]

\( s_i \) is a mixture of features for \( t_i \) and \( t_{i-1} \). \( W_{TSP}^j \) is the parameter matrix on prediction whether the j-th type of operation is made. \( \bar{q}_i^j \in (0, 1) \) is the ground-truth label on making the j-th operation with the i-th turn and \( p_i^j \) is the predicted probability on making it. We set \( t_0 \) to be zero vector while computing. \( N_T \) binary classification is calculated instead of a single multi-class classification since several types of modification could be made in one breath by adding a new turn.

### 4.3. Contextual Schema Prediction

Contextual Schema Prediction (CSP) task is designed to help the encoder model to map each modification operation to each database operation applied on columns from tables. And thus we use the representations of schema tokens to make predictions.

We also use the encoding vector of the special token “(s)” as the representation of a column from database schema, and use the column representation to predict which kind of change is made on it. A number of \( N_C = 11 \) types of modifications are defined, including adding to select, deleting from where, changing of distinct etc. As the same reason as in the TSP task, a single column can have multiple modification in different sub-clauses of a SQL, so we also use \( N_C \) binary classifications as the objective of this task. Notate \([c_1, \ldots, c_M]\) to be the encoding vector of the M columns from database schema, CSP loss is computed as

\[
q_i^j = \text{Sigmoid} \left( W_{CSP}^j(c_i) \right),
\]

\[
L_{CSP} = \sum_{n=1}^{N_C} \sum_{m=1}^{M} \left( \bar{q}_i^j \log q_i^j + (1 - \bar{q}_i^j) \log(1 - q_i^j) \right),
\]

where \( W_{CSP}^j \) is trainable parameter matrix for the j-th kind of schema usage changing, and \( \bar{q}_i^j \) is the ground-truth label indicating whether a j-th kind of change is applied on the i-th column.

Notice that different from TSP which computes semantic changes between every adjacent turn pairs, CSP only takes the effect of the last turn into consideration and neglecting the previous ones. In this way, CSP can enforce the encoder model to better focus on the semantics of the last turn and in turn boosting the text-to-SQL parser to generate correct SQLs.

### 4.4. Training Objective

The text-to-SQL parser is trained in a multi-task training way that the proposed three losses are optimized at the same time.

\[
\mathcal{L} = \mathcal{L}_{dec} + \alpha \mathcal{L}_{TSP} + \beta \mathcal{L}_{CSP},
\]

where \( \alpha > 0 \) and \( \beta > 0 \) are two hyper-parameters to control the weight of TSP loss and CSP loss. In practice, we set \( \alpha = 0.5 \) and \( \beta = 8 \) to harvest our best results. Compared with pre-train-then-fine-tune paradigm (e.g., [19]), multi-task training significantly more efficient in terms of computational cost.
5. Experiments

5.1. Datasets
Experiments are conducted on SparC, a large-scale cross-domain dataset for conversational text-to-SQL. SparC is a context-dependent dataset among which parsing the following SQLs requires a correct understanding of the previous turns. There are 2,159 and 422 conversations in the training set and development set respectively, with the average number of turns being 2.97 and 2.85. An online judgement is available for submission, of which the test set is not publicly released.

5.2. Implementation Details
We follow the same hyper-parameter settings as in [40]. Both QM (query exact match) and IM (interaction exact match) are chosen as metrics following the same standard with our baseline methods. We implement RAT-SQL-TC with the GAP model [40]. GAP is a domain-adapted version of BERT, which is tuned with the single-turn SQL generation task in a sequence-to-sequence manner, and only the BERT encoder is kept.

5.3. Results
The performance of our proposed RAT-SQL-TC (GAP) and several baseline methods is shown in Table 1.

| Method                  | SparC Dev | SparC Test |
|-------------------------|-----------|------------|
| EditSQL + BERT [36]     | 47.6      | 39.5       |
| IGSQL + BERT [36]       | 50.7      | 51.2       |
| R^2 SQL + BERT [36]     | 54.1      | 55.8       |
| RAT-SQL (BERT)          | 56.8      | 33.4       |
| RAT-SQL + Score [39]    | 62.5      | 62.4       |
| RAT-SQL (GAP)           | 59.6      | 40.5       |
| RAT-SQL-TC (GAP)        | 64.1      | 44.1       |

| Model   | Turn 1 | Turn 2 | Turn 3 | Turn 4 |
|---------|--------|--------|--------|--------|
| RAT-SQL-TC w/o. TC | 75.4 (±3.1) | 72.5   | 56.9   | 40.9 (±1.3) |
| RAT-SQL-TC w/ CST  | 64.8 (±7.1) | 54.3 (±4.0) | 40.9 (±1.3) | 39.8 |
| RAT-SQL-TC w/ CST  | 64.8 (±7.1) | 54.3 (±4.0) | 40.9 (±1.3) | 39.8 |

Table 1: QM and IM accuracy of our proposed RAT-SQL-TC (GAP) and several baseline methods. RAT-SQL-TC (GAP) outperforms all baseline methods and achieves new state-of-the-art results.

5.4. Ablation Studies and Analysis
In order to better understand how our proposed RAT-SQL-TC works, we conducted ablation studies and analysis with RAT-SQL-TC (GAP) on the SparC development set.

Both TSP and CSP aim at better modeling information flow during a conversation, and thus we evaluate how they each influences the overall performance. We remove each of them and test the model’s performance, whose results are shown in Table 2. Significant performance decline is observed without either TSP or CSP on both QM and IM. To be specified, there shows a 4.5% absolute drop on IM and a 3.9% absolute drop on QM without TSP, which demonstrates the effectiveness on explicitly modeling context changes to track the information flow during a conversation. Interestingly, the IM accuracy is even lower than that of pure RAT-SQL, which indicates that an over attention on column usage changes without modeling semantic changes on natural language aspect may even harm the performance. By removing CSP which maps semantic changes into database schema tokens, both QM and IM decrease for 3.1%, proving that a proper mechanism on modeling semantics with database schema is essential for making correct predictions. TSP and CSP work from natural-language-understanding and database-schema-aware aspects respectively on enhancing semantic parsers to generate correct SQLs. Although each of them alone cannot bring significant improvement on accuracy metrics, the combination of these two objectives work well on achieving even better performance.

| Method          | QM    | IM    |
|-----------------|-------|-------|
| RAT-SQL-TC      | 64.1  | 44.1  |
| RAT-SQL-TC w/ TSP | 61.0 (±3.1) | 41.0 (±3.1) |
| RAT-SQL-TC w/ CST | 60.2 (±3.9) | 39.6 (±4.5) |
| RAT-SQL         | 59.6 (±4.5) | 40.5 (±3.5) |

Table 2: Model performance by ablating TSP and CSP.

Since the two tasks of TC are designed to better modeling contextual information during a conversation, we evaluate how much improvement can TC bring on individual turns in terms of question match accuracy. Table 3 shows the QM accuracy at each separate turn. Both RAT-SQL and RAT-SQL-TC show the same trend on predicting poorer SQLs with a larger turn number, indicating it harder on understanding the whole context with more turns to generate correct predictions. However, compared with pure RAT-SQL, adding TC as auxiliary tasks can significantly improve QM accuracy on queries with two or three turns. TC performs as an context modeling strategy on both natural-language and database perspectives, and thus improves semantic parser on modeling queries with long contextual information.

6. Conclusion
Modeling semantic flows during a conversation for semantic parsing is a tough task for multi-turn semantic parsing. On handling this obstacle, in this paper we proposed RAT-SQL-TC which adds two auxiliary tasks (i.e., TSP and CSP) during semantic parser training. TSP and CSP works from natural-language-understanding perspective and database-schema-aware perspective respectively on modeling multi-turn conversation and converting semantics into SQLs. We demonstrate the highly effectiveness of TC on a large-scale open-domain benchmark and achieve new state-of-the-art results.
7. References

[1] T. Yu, R. Zhang, M. Yasunaga, Y. C. Tan, X. V. Lin, S. Li, H. Er, I. Li, B. Pang, T. Chen et al., “Sparc: Cross-domain semantic parsing in context,” arXiv preprint arXiv:1906.02235, 2019.

[2] V. Zhong, M. Lewis, S. I. Wang, and L. Zettlemoyer, “Grounded adaptation for zero-shot executable semantic parsing,” arXiv preprint arXiv:2009.07396, 2020.

[3] R. Zhang, T. Yu, H. Y. Er, S. Shin, E. Xue, X. V. Lin, T. Shi, C. Xiong, R. Socher, and D. Radev, “Editing-based sql query generation for cross-domain context-dependent questions,” arXiv preprint arXiv:1909.00786, 2019.

[4] B. Wang, R. Shin, X. Liu, O. Polozov, and M. Richardson, “Ratsql: Relation-aware schema encoding and linking for text-to-sql parsers,” arXiv preprint arXiv:1911.04942, 2019.

[5] F. B. Thompson, P. C. Lockemann, B. Dostert, and R. Deverill, “Rel: A rapidly extensible language system,” in Proceedings of the 1969 24th national conference, 1969, pp. 399–417.

[6] W. A. Woods, “Progress in natural language understanding: an application to lunar geology,” in Proceedings of the June 4-8, 1973, national computer conference and exposition, 1973, pp. 441–450.

[7] M. Templeton and J. F. Burger, “Problems in natural-language interface to dbms with examples from eufid,” in First Conference on Applied Natural Language Processing, 1983, pp. 3–16.

[8] J. M. Zelle and R. J. Mooney, “Learning to parse database queries using inductive logic programming,” in Proceedings of the national conference on artificial intelligence, 1996, pp. 1050–1055.

[9] C. Thompson, “Acquiring word-meaning mappings for natural language interfaces,” Journal of Artificial Intelligence Research, vol. 18, pp. 1–44, 2003.

[10] T. Kwiakowski, L. Zettlemoyer, S. Goldwater, and M. Steedman, “Inducing probabilistic ccg grammars from logical form with higher-order unification,” in Proceedings of the 2010 conference on empirical methods in natural language processing, 2010, pp. 1223–1233.

[11] L. Dong and M. Lapata, “Language to logical form with neural attention,” arXiv preprint arXiv:1601.01280, 2016.

[12] R. Jia and P. Liang, “Data recombination for neural semantic parsing,” arXiv preprint arXiv:1606.03622, 2016.

[13] J. Cheng, S. Reddy, V. Sarawat, and M. Lapata, “Learning structured natural language representations for semantic parsing,” arXiv preprint arXiv:1704.08387, 2017.

[14] J. Krishnamurthy, P. Dasigi, and M. Gardner, “Neural semantic parsing with type constraints for semi-structured tables,” in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 1516–1526.

[15] L. Dong, C. Quirk, and M. Lapata, “Confidence modeling for neural semantic parsing,” arXiv preprint arXiv:1805.04064, 2018.

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

[17] Y. Sun, D. Tang, N. Duan, J. Ji, G. Cao, X. Feng, B. Qin, T. Liu, and M. Zhou, “Semantic parsing with syntax-and-table-aware sql generation,” arXiv preprint arXiv:1804.00358, 2018.

[18] T. Guo and H. Gao, “Content enhanced bert-based text-to-sql generation,” arXiv preprint arXiv:1910.07179, 2019.

[19] T. Yu, R. Zhang, K. Yang, M. Yasunaga, D. Wang, Z. Li, J. Ma, I. Li, Q. Yao, S. Roman et al., “Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task,” arXiv preprint arXiv:1809.08887, 2018.

[20] X. Xu, C. Liu, and D. Song, “Sqlnet: Generating structured queries from natural language without reinforcement learning,” arXiv preprint arXiv:1711.04436, 2017.

[21] T. Yu, Z. Li, Z. Zhang, R. Zhang, and D. Radev, “Typsql: Knowledge-based type-aware neural text-to-sql generation,” arXiv preprint arXiv:1804.00769, 2018.

[22] W. Hwang, J. Yin, S. Park, and M. Seo, “A comprehensive exploration on wikitext with table-aware word contextualization,” arXiv preprint arXiv:1902.01069, 2019.

[23] L. Dong and M. Lapata, “Coarse-to-fine decoding for neural semantic parsing,” arXiv preprint arXiv:1805.04793, 2018.

[24] P. He, Y. Mao, K. Chakrabarti, and W. Chen, “X-sql: reinforce context into schema representation,” Microsoft Research: Artificial Intelligence, 2018.

[25] Q. Lyu, K. Chakrabarti, S. Hathi, S. Kundu, J. Zhang, and Z. Chen, “Hybrid ranking network for text-to-sql,” arXiv preprint arXiv:2008.04759, 2020.

[26] T. Yu, M. Yasunaga, K. Yang, R. Zhang, D. Wang, Z. Li, and D. Radev, “Syntaxsqlnet. Syntax tree networks for complex and cross-domain text-to-sql task,” arXiv preprint arXiv:1810.05237, 2018.

[27] J. Guo, Z. Zhan, Y. Gao, Y. Xiao, J.-G. Lou, T. Liu, and D. Zhang, “Towards complex text-to-sql in cross-domain database with intermediate representation,” arXiv preprint arXiv:1905.08205, 2019.

[28] J. Herzig, P. Shaw, M.-W. Chang, K. Guu, P. Pasupat, and Y. Zhang, “Unlocking compositional generalization in pre-trained models using intermediate representations,” arXiv preprint arXiv:2104.07478, 2021.

[29] B. Bogin, M. Gardner, and J. Berant, “Representing schema structure with graph neural networks for text-to-sql parsing,” arXiv preprint arXiv:1905.06241, 2019.

[30] ——, “Global reasoning over database structures for text-to-sql parsing,” arXiv preprint arXiv:1908.11214, 2019.

[31] R. Cao, L. Chen, Z. Chen, S. Zhu, and K. Yu, “Lqsegsl: Line graph enhanced text-to-sql model with mixed local and non-local relations,” arXiv preprint arXiv:2106.01093, 2021.

[32] Z. Chen, L. Chen, Y. Zhao, R. Cao, Z. Xu, S. Zhu, and K. Yu, “Shadowwnn: Graph projection neural network for text-to-sql parsing,” arXiv preprint arXiv:2104.04689, 2021.

[33] J. Andreas, “Good-enough compositional data augmentation,” arXiv preprint arXiv:1904.06545, 2019.

[34] B. Wang, W. Yin, X. V. Lin, and C. Xiong, “Learning to synthesize data for semantic parsing,” arXiv preprint arXiv:2104.05827, 2021.

[35] T. Yu, R. Zhang, H. Y. Er, S. Li, E. Xue, B. Pang, X. V. Lin, Y. C. Tan, T. Shi, Z. Li et al., “Cosql: A conversational text-to-sql challenge towards cross-domain natural language interfaces to databases,” arXiv preprint arXiv:1909.05758, 2019.

[36] Y. Cai and X. Wan, “Igsql: Database schema interaction graph based neural model for context-dependent text-to-sql generation,” arXiv preprint arXiv:2014.05744, 2020.

[37] R.-Z. Wang, Z.-H. Ling, J.-B. Zhou, and Y. Hu, “Tracking interaction states for multi-turn text-to-sql semantic parsing,” arXiv preprint arXiv:2012.04995, 2020.

[38] B. Hui, R. Geng, Q. Ren, B. Li, Y. Li, J. Sun, F. Huang, L. Si, P. Zhu, and X. Zhu, “Dynamic hybrid relation network for cross-domain context-dependent semantic parsing,” arXiv preprint arXiv:2101.01686, 2021.

[39] T. Yu, R. Zhang, A. Polozov, C. Meek, and A. H. Awadallah, “Score: Pre-training for context representation in conversational semantic parsing,” in International Conference on Learning Representations, 2020.

[40] P. Shi, P. Ng, Z. Wang, H. Zhu, A. H. Li, J. Wang, C. N. d. Santos, and B. Xiang, “Learning contextual representations for semantic parsing with generation-augmented pre-training,” AAAI, 2021.