SYNTQA: Synergistic Table-based Question Answering via Mixture of Text-to-SQL and E2E TQA

Siyue Zhang[♢♡] Anh Tuan Luu[♢] Chen Zhao[♠♣]

[♢]Nanyang Technological University [♠]NYU Shanghai
[♡]Alibaba-NTU Singapore Joint Research Institute
[♣]Center for Data Science, New York University
siyue001@e.ntu.edu.sg, anhtuan.luu@ntu.edu.sg, cz1285@nyu.edu

Abstract

Text-to-SQL parsing and end-to-end question answering (E2E TQA) are two main approaches for Table-based Question Answering task. Despite success on multiple benchmarks, they have yet to be compared and their synergy remains unexplored. In this paper, we identify different strengths and weaknesses through evaluating state-of-the-art models on benchmark datasets: Text-to-SQL demonstrates superiority in handling questions involving arithmetic operations and long tables; E2E TQA excels in addressing ambiguous questions, non-standard table schema, and complex table contents. To combine both strengths, we propose a Synergistic Table-based Question Answering approach that integrate different models via answer selection, which is agnostic to any model types. Further experiments validate that ensembling models by either feature-based or LLM-based answer selector significantly improves the performance over individual models. Code will be publicly available at https://github.com/siyue-zhang/SynTableQA.

1 Introduction

Table QA (TQA) takes a question and a table, and finds an answer based on the evidence from the table (Pasupat and Liang, 2015). With the help of large scale datasets (Zhong et al., 2017; Yu et al., 2018, 2019; Shi et al., 2020), state-of-the-art (SOTA) TQA systems primarily focus on two approaches: *semantic parsing* (Text-to-SQL) that predicts a SQL query as intermediate semantic representation of the question, and then executes the SQL to find the answer (Wang et al., 2020; Scholak et al., 2021; Li et al., 2023a); *end-to-end system* (E2E TQA) that directly generates the answer from models pre-trained on table corpora, imitating human-like reasoning on questions and tables (Pasupat and Liang, 2015; Iyyer et al., 2017; Gupta et al., 2023). Despite serving for a similar purpose, it's unclear what advantages these approaches have and their potential synergy.

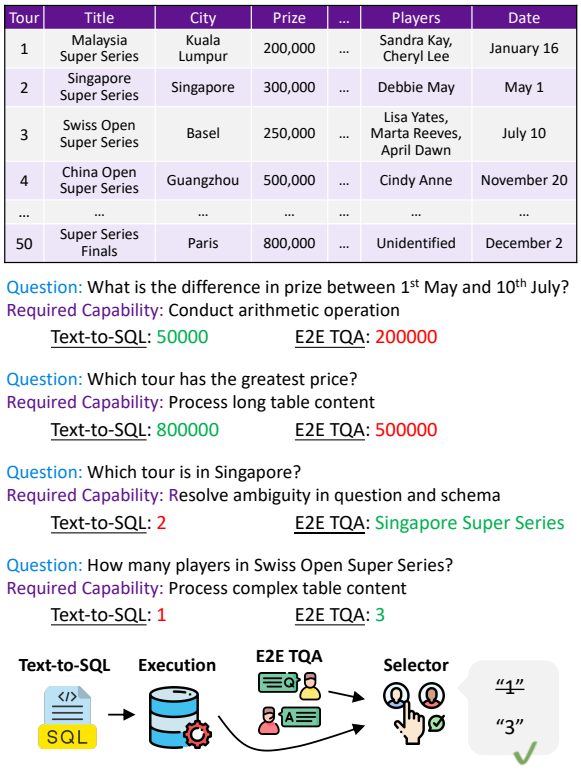

Figure 1: A demonstration of SOTA Table QA models' strengths in solving different types of table-based questions, followed by an overview of SYNTQA. In a synergistic way, SYNTQA aggregates candidate answers from Text-to-SQL and E2E TQA models, and then select the final answer. The answers in green color are the correct answers.

To answer these questions, we first (re)evaluate SOTA Text-to-SQL models, i.e., T5 (Raffel et al., 2020), GPT (OpenAI, 2023), and DIN-SQL (Pourreza and Rafiei, 2024), as well as E2E TQA models, i.e., TAPEX (Liu et al., 2022), OmniTab (Jiang et al., 2022), and GPT, on benchmark datasets WTQ (Pasupat and Liang, 2015) and WIKISQL (Zhong et al., 2017). The experiments show that while both Text-to-SQL and E2E TQA approaches

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 2352–2364
November 12-16, 2024

are adept at simple questions, they have complementary strengths for complex questions and tables, as shown in Figure 1 (top): Text-to-SQL is more proficient in numerical reasoning and processing long tables while E2E TQA is better at ambiguous questions, complex schema and contents.

Motivated by their distinct strengths, we propose **Syn**ergistic **T**able-based **Q**uestion **A**nswering (**SYNTQA**, bottom of Figure 1), which aims to integrate the strengths of both models via answer selection. At each time, given the input of table, question, Text-to-SQL answer, and E2E TQA answer, the answer selector identifies the more probable correct one from Text-to-SQL and E2E TQA answers. Experiments show that both feature-based selector and LLM-based selector provide significant improvement over single models.

2 Table Question Answering Task

Table Question Answering has received significant attention as it helps non-experts interact with complex tabular data. Formally, given an input question $\mathcal{Q} = \{q_1, q_2, \ldots, q_n\}$ and a table $\mathcal{T}$ with $\mathcal{R}$ rows and $\mathcal{C}$ columns, and each cell $\mathcal{T}_{i,j}$ contains a real value, Table QA aims to produce an answer $\mathcal{A} = \{a_1, a_2, \ldots, a_k\}$, where q_n and a_k are tokens. Then we introduce two main approaches for Table QA: Text-to-SQL and E2E TQA.

Text-to-SQL. Table QA problem is originally framed as semantic parsing, also known as Text-to-SQL parsers, where a parser takes both question and table header as input, and predicts a SQL query that is directly executable to get the answer. Early neural sequence-to-sequence parsers (Guo et al., 2019; Wang et al., 2020; Rubin and Berant, 2021) encode question/schema with attention mechanism and uses SQL grammar to guide the decoding process. Recent approaches take advantages of pre-trained models, and they either fine-tune (Wang et al., 2018; Scholak et al., 2021) or prompt (Gao et al., 2024; Pourreza and Rafiei, 2024) large models for Text-to-SQL parsing.

E2E TQA. Several issues limit applying Text-to-SQL parsers into real scenarios: training SOTA parsers require large amounts of expensive SQL annotations; existing parsers largely ignore the value of table contents. With the help of model pre-trained on large scale table corpus, recent works focus on end-to-end Table QA that ignores generating SQL queries as an intermediate step and directly predicts the final answer through either fine-tune (Liu et al., 2022; Zhao et al., 2022; Jiang et al., 2022) or prompt large models (Chen, 2023).

3 Evaluating Text-to-SQL and E2E TQA

In this section, we evaluate existing Text-to-SQL and E2E TQA models on two benchmark datasets: WTQ and WIKISQL.

3.1 Experimental Setup

Dataset. WTQ comprises 22,033 instances with a diverse array of intricate questions and tables. SQUALL (Shi et al., 2020) annotates 11,276 WTQ instances with pre-processed tables and SQL queries.[1] Compared with classic datasets, e.g., WIKISQL and Spider (Yu et al., 2018), designed for SQL prediction on well-maintained databases, WTQ contains complex tables and questions which are difficult to answer with SQL queries. As a large portion of Spider tables does not have table content, we use WIKISQL to validate the generalizability of our findings, which contains 80,654 instances.

Model and Metric. We evaluate SOTA models that have publicly available source code or APIs: Text-to-SQL includes T5 (Raffel et al., 2020), GPT (OpenAI, 2023), and DIN-SQL (Pourreza and Rafiei, 2024); E2E TQA includes TAPEX (Liu et al., 2022), OmniTab (Jiang et al., 2022), and GPT (OpenAI, 2023). As Text-to-SQL models often generate invalid SQL queries (Lin et al., 2020), we devise a post-processing module to screen table content, rectify query misspellings, identify the closest string values, and resolve mismatches. Similar to (Scholak et al., 2021), the post-processing module ensures the correct SQL grammar. For fine-tuned models, we choose the `large` version. For fair comparison, we report answer string exact match (EM) accuracy.

3.2 Results

According to Table 1, prompting methods (i.e., GPT and DIN-SQL) underperform fine-tuned models in table understanding on WTQ and WIKISQL, aligning with findings in (Li et al., 2024; Liu et al., 2024). Thus, we primarily focus on fine-tuned Text-to-SQL and E2E TQA models. Best Text-to-SQL and E2E TQA models achieve comparable accuracy, but notably, 27.6% of WTQ and 11.7%

[1]We train Text-to-SQL models on SQUALL and test on WTQ, as 20% of WTQ questions lack SQL annotations and cannot be answered by Text-to-SQL.

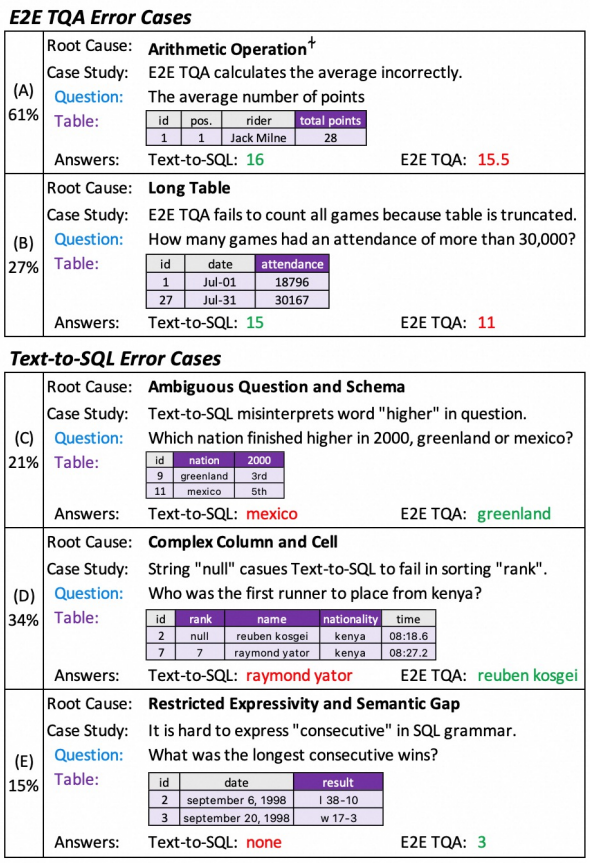

Figure 2: Error case analysis. †Arithmetic operation errors include questions with both long and short tables. Tables are regarded as long if their linearized sequences have more tokens than the Table QA model input length. The percentage numbers on the left indicate the quantity of error cases, and remaining percentage points correspond to other errors, such incorrect labels.

of WIKISQL questions were correctly answered exclusively by either Text-to-SQL or E2E TQA. It implies that models excel in tackling different types of table-based questions. To further investigate the strengths and weaknesses, we analyze 200 erroneous cases summarized in Figure 2 (see detailed breakdown in Appendix B).

Text-to-SQL is skilled at arithmetic operations. It is evident in Figure 2 (A) that 61% of E2E TQA error cases involve arithmetic operations including counting, summation, averaging, and subtraction. Despite existing E2E TQA approaches (Herzig et al., 2020; Eisenschlos et al., 2020) have incorporated a separate aggregation operator into model design, the range of supported operations is limited with suboptimal performance. In contrast, Text-to-SQL provides more accurate and consistent results for arithmetic operations through symbolic reasoning (Cheng et al., 2023; Liu et al., 2024).

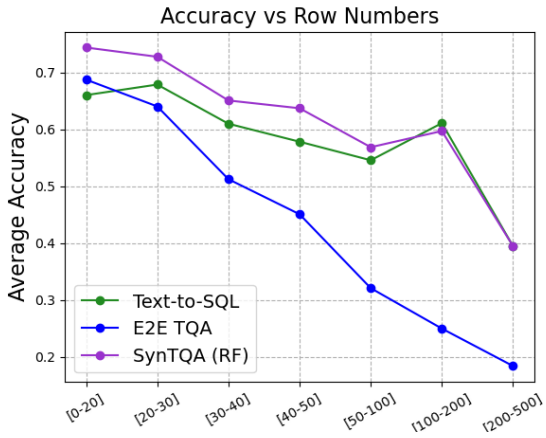

Figure 3: The impact of table size (i.e., number of rows) on the accuracy of E2E TQA, Text-to-SQL, and SYNTQA (RF) on the the test set of WTQ. The x-axis represents the row number ranges, and the y-axis shows the average accuracy for each method.

Text-to-SQL is adept at long tables. When faced with long tables, comparing with Text-to-SQL, E2E TQA accuracy dramatically declines with increased table size (see details in Figure 3 and Appendix C). This is because existing E2E TQA approaches are limited in processing and understanding long context, therefore are only able to take truncated table as input. In contrast, Text-to-SQL approaches primarily focus on table headers, and are more robust to incomplete or long table content. For example, in a long table like Figure 2 (B), Text-to-SQL is able to aggregate all critical information over the rows.

E2E TQA is robust to ambiguous questions and non-standard table schema. Rather than centering on table schema, E2E TQA prioritizes table content. Analysing table content is particularly useful for resolving the ambiguity. As shown in Figure 2 (C), the term "higher" may refer to a bigger value in quantity or a smaller value in rank. And E2E TQA is more effective to infer that "higher" corresponds to a smaller ranking value by incorporating the table content (e.g., "3rd" and "5th"). Instead, Text-to-SQL relies on the relevant column header "`2000`" and mistakenly searches for the bigger value. Furthermore, as depicted in the third question of Figure 1, the non-standard header `tour` misleads Text-to-SQL to retrieve the identity number. In contrast, E2E TQA accurately predicts the official title of the tour.

E2E TQA is flexible to process complex table content. Complex content arises with the mixing of data types within same column, and Text-to-SQL cannot find such nuanced difference, without looking at table contents. According to Figure 2 (D), Text-to-SQL cannot exclude the row with "null" rank and makes the wrong prediction then.

Some questions cannot map to a SQL query. As pointed out by (Shi et al., 2020), there are cases where SQL queries are insufficiently expressive. According to Figure 2 (E), Text-to-SQL cannot answer questions related to phrases "approximate", while E2E TQA is about to find the answers.

Text-to-SQL requires post-process the executed answers. We also find that in some cases, additional steps are needed to translate SQL query results to natural language answers, where a notable semantic gap exists. For example, mapping "1" to "longer" for "longer or shorter" question. E2E TQA approaches do not have these limitations as they directly predict the final answers.

Model	WTQ		WIKISQL	
	Dev	Test	Dev	Test
Text-to-SQL Models				
DIN-SQL		44.6		81.7
GPT + TC + P		50.0		82.2
T5 + TC + P	66.7	**64.7**	88.3	**89.6**
E2E TQA Models				
GPT		56.8		62.6
TAPEX	57.5	57.0	89.2	**89.5**
OmniTab	63.7	**62.6**	89.7	89.0
Ensemble Models				
SYNTQA (RF)		**71.6**		**93.2**
SYNTQA (GPT)		70.4		93.0
SYNTQA (Oracle)		77.5		95.1

Table 1: Accuracy on WTQ and WIKISQL datasets comparing SYNTQA with baselines. The best test result is highlighted in **bold**. Oracle result indicates the maximum potential of mixing Text-to-SQL and E2E TQA models (TC: Table Content, P: Post-processing).

4 SYNTQA: Selecting Correct Answer

Above findings show that different models solve different questions, so we use a selector to choose the answer. Specifically, at each time, the selector receives the input of table $\mathcal{T}$, question $\mathcal{Q}$, Text-to-SQL prediction and confidence $\widehat{\mathcal{A}}_{SQL}$, along with E2E TQA prediction and confidence $\widehat{\mathcal{A}}_{E2E}$. Afterwards, the selector determines the correct answer $\widehat{\mathcal{A}}_{SEL}$, where $\widehat{\mathcal{A}}_{SEL} \in \{\widehat{\mathcal{A}}_{SQL}, \widehat{\mathcal{A}}_{E2E}\}$. In general, answer selection can be done through feature-based classification or LLM-based contextual reasoning, which is discussed in this section.

We use the best performing base models, i.e., fine-tuned T5 for Text-to-SQL and OmniTab for E2E TQA in the ensemble model.

4.1 Selector Designs

Feature-based Selector. SYNTQA (RF) trains a random forest classifier to make the selection.[2] We design the following features to train the classifier: *question characteristics* (e.g., question word and length), *table characteristics* (e.g., table size, header and question overlapping, and truncation), *Text-to-SQL answer characteristics* (e.g., confidence, query execution, and queried answer data type), and *E2E TQA answer characteristics* (e.g., confidence and length). The full list of features and training details are included in Appendix E.

LLM-based Selector. SYNTQA (GPT) does not require training data thanks to LLMs' remarkable few-shot capabilities. For comparison, we evaluate LLMs' answer selection capability via direct prompting in Table 1.[3] Furthermore, we propose a heuristic-enhanced prompting strategy to elevate the SOTA performance to 74.4% and 93.6% on WTQ and WIKISQL (see details in Appendix F).

4.2 Results

According to Table 1 (Bottom), our ensemble models exhibit substantial improvement over individual models. They achieve comparable performance with recent tool-based LLMs on WTQ while saving computational costs, e.g., Dater (Ye et al., 2023) 65.9% and Mix SC (Liu et al., 2024) 73.6%. As our findings are orthogonal to these methods, we demonstrate a case integrating the concept of Mix SC in Appendix G. The effectiveness can be attributed to selectors' high success rate (nearly 80%) in selecting correct answers. Notably, the confidence of Text-to-SQL and E2E TQA models is the most impactful feature for SYNTQA (RF).

[2]We evaluate various classic classifiers and identify random forest as the top performer in Appendix E.3.

[3]We employ `gpt-3.5-turbo-0125` for the evaluation.

4.3 SQL Annotation Efficiency

Since manually creating SQL annotations can be costly (Shi et al., 2020), we conducted experiments to study how the accuracy improvement varies with different amounts of SQL annotations, using the feature-based selector in the WTQ dataset. The answers are assumed to be always fully available, leading to a stable performance of E2E TQA.

As shown in Figure 4, 10% of SQL annotations (∼900) enhanced E2E TQA accuracy by 5%. The improvement potential and actual improvement continue to grow with the increase of the SQL annotation amount. Trade-offs can be made between the performance improvement and annotation amounts depending on the use case.

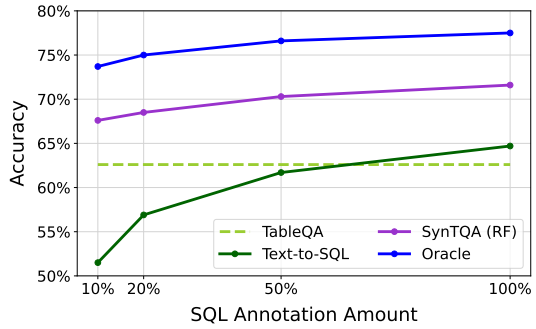

Figure 4: WIKISQL test set accuracy versus the percentage amount of SQL annotations provided by SQUALL. Even an inferior Text-to-SQL model trained with a more limited set of SQL annotations can substantially enhance the E2E TQA model.

4.4 Robustness Analysis

In addition to individual Text-to-SQL and E2E TQA models such as previous works (Pi et al., 2022; Singha et al., 2023), we evaluate our ensemble approach SYNTQA (RF) with adversarial perturbations such as replacing key question entities and adding table columns. The evaluation is performed on the ROBUT-WIKISQL dataset (Zhao et al., 2023). We find that different models exhibited degradation on distinct adversarial samples. Employing model assembling mitigates the performance degradation experienced by individual models significantly (see details in Table 5).

5 Other Related Work

Mixture-of-Experts. Since proposed by (Jacobs et al., 1991), Mixture-of-Experts has been applied in a wide fields of machine learning (Li et al., 2022; Gururangan et al., 2023). We follow the same concept and route the experts from the sample level (Puerto et al., 2021; Si et al., 2023), i.e. selecting an expert model for each test instance.

Tool-based LLMs. With LLMs' strong textual reasoning and tool-use capabilities, recent Table QA methods (Cheng et al., 2023; Ye et al., 2023; Liu et al., 2024) call executable programs (e.g., SQL and Python) as needed to retrieve relevant contexts, facilitating reasoning. We provide an alternative ensemble approach that does not rely on computationally expensive LLMs.

6 Conclusion

This study delved into the comparative analysis of two Table QA approaches: Text-to-SQL and E2E TQA. Results indicate Text-to-SQL's proficiency in arithmetic operations and long tables and E2E TQA's advantages in resolving ambiguity and complexity in the question and table. We enhance performance on Table QA datasets by combining models through answer selectors. We plan to extend the method to more challenging problems such as hybrid TQA (Chen et al., 2020; Zhu et al., 2021).

Limitations

Although OmniTab is pre-trained for E2E TQA, T5 is not a model specifically designed for Text-to-SQL. Most Text-to-SQL models are tailored for the Spider dataset (Wang et al., 2018; Rubin and Berant, 2021; Scholak et al., 2021). Table or passage retrievers (Karpukhin et al., 2020; Herzig et al., 2021) can be applied to select certain rows and columns before truncating the long tables which might improve E2E TQA performance. As for SYNTQA (GPT), we constrain GPT to select an answer from candidates, which abandons its capability to provide a different answer when both candidates are wrong. In more challenging datasets which necessitate both textual and tabular data (Chen et al., 2020; Zhu et al., 2021), our method may not be as flexible and effective as tool-based LLMs (Li et al., 2023b; Asai et al., 2024).

Ethics Statement

SYNTQA were developed using WTQ (Pasupat and Liang, 2015), SQUALL (Shi et al., 2020), WIKISQL (Zhong et al., 2017), and ROBUT (Zhao et al., 2023), which are publicly available under

the licenses of CC BY-SA 4.0[4], BSD 3-Clause[5], and MIT[6]. We used 4 NVIDIA Quadro RTX8000 GPUs to fine-tune models. SYNTQA (RF) and SYNTQA (GPT) were constructed and executed solely using CPU. SYNTQA (GPT) relies on OpenAI API and using other GPT versions will lead to varied performance. No manual annotation and human study are involved in this study.

Acknowledgements

This research is supported by the RIE2025 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) (Award I2301E0026), administered by A*STAR, as well as supported by Alibaba Group and NTU Singapore. Siyue Zhang and Chen Zhao were supported by Shanghai Frontiers Science Center of Artificial Intelligence and Deep Learning, NYU Shanghai. This work was supported in part through the NYU IT High Performance Computing resources, services, and staff expertise.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Self-reflective retrieval augmented generation. In *The Twelfth International Conference on Learning Representations*.

Wenhu Chen. 2023. Large language models are few(1)-shot table reasoners. *Preprint*, arXiv:2210.06710.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020. HybridQA: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Zhoujun Cheng, Tianbao Xie, Peng Shi, Chengzu Li, Rahul Nadkarni, Yushi Hu, Caiming Xiong, Dragomir Radev, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Tao Yu. 2023. Binding language models in symbolic languages. In *The Eleventh International Conference on Learning Representations*.

Julian Eisenschlos, Syrine Krichene, and Thomas Müller. 2020. Understanding tables with intermediate pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Dawei Gao, Haibin Wang, Yaliang Li, Xiuyu Sun, Yichen Qian, Bolin Ding, and Jingren Zhou. 2024. Text-to-sql empowered by large language models: A benchmark evaluation. *Proceedings of the VLDB Endowment*.

Jiaqi Guo, Zecheng Zhan, Yan Gao, Yan Xiao, Jian-Guang Lou, Ting Liu, and Dongmei Zhang. 2019. Towards complex text-to-SQL in cross-domain database with intermediate representation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Vivek Gupta, Pranshu Kandoi, Mahek Vora, Shuo Zhang, Yujie He, Ridho Reinanda, and Vivek Srikumar. 2023. TempTabQA: Temporal question answering for semi-structured tables. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Suchin Gururangan, Margaret Li, Mike Lewis, Weijia Shi, Tim Althoff, Noah A. Smith, and Luke Zettlemoyer. 2023. Scaling expert language models with unsupervised domain discovery. *Preprint*, arXiv:2303.14177.

Jonathan Herzig, Thomas Müller, Syrine Krichene, and Julian Eisenschlos. 2021. Open domain question answering over tables via dense retrieval. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. TaPas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Mohit Iyyer, Wen-tau Yih, and Ming-Wei Chang. 2017. Search-based neural structured learning for sequential question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. 1991. Adaptive mixtures of local experts. *Neural Computation*.

Zhengbao Jiang, Yi Mao, Pengcheng He, Graham Neubig, and Weizhu Chen. 2022. OmniTab: Pretraining with natural and synthetic data for few-shot table-based question answering. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Jinyang Li, Binyuan Hui, Reynold Cheng, Bowen Qin, Chenhao Ma, Nan Huo, Fei Huang, Wenyu Du, Luo Si, and Yongbin Li. 2023a. Graphix-t5: mixing pre-trained transformers with graph-aware layers for text-to-sql parsing. In *Proceedings of the Thirty-Seventh*

[4] https://creativecommons.org/licenses/by-sa/4.0/

[5] https://opensource.org/license/bsd-3-clause

[6] https://opensource.org/license/mit

AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence.

Margaret Li, Suchin Gururangan, Tim Dettmers, Mike Lewis, Tim Althoff, Noah A. Smith, and Luke Zettlemoyer. 2022. Branch-train-merge: Embarrassingly parallel training of expert language models. *Preprint*, arXiv:2208.03306.

Peng Li, Yeye He, Dror Yashar, Weiwei Cui, Song Ge, Haidong Zhang, Danielle Rifinski Fainman, Dongmei Zhang, and Surajit Chaudhuri. 2024. Table-gpt: Table fine-tuned gpt for diverse table tasks. *Proceedings of the ACM on Management of Data*.

Xingxuan Li, Ruochen Zhao, Yew Ken Chia, Bosheng Ding, Shafiq Joty, Soujanya Poria, and Lidong Bing. 2023b. Chain-of-knowledge: Grounding large language models via dynamic knowledge adapting over heterogeneous sources. In *The Twelfth International Conference on Learning Representations*.

Xi Victoria Lin, Richard Socher, and Caiming Xiong. 2020. Bridging textual and tabular data for cross-domain text-to-SQL semantic parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Qian Liu, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, and Jian-Guang Lou. 2022. TAPEX: Table pre-training via learning a neural SQL executor. In *International Conference on Learning Representations*.

Tianyang Liu, Fei Wang, and Muhao Chen. 2024. Rethinking tabular data understanding with large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*.

Xinyu Pi, Bing Wang, Yan Gao, Jiaqi Guo, Zhoujun Li, and Jian-Guang Lou. 2022. Towards robustness of text-to-SQL models against natural and realistic adversarial table perturbation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*.

Mohammadreza Pourreza and Davood Rafiei. 2024. Din-sql: decomposed in-context learning of text-to-sql with self-correction. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*.

Haritz Puerto, Gozde Gul cSahin, and Iryna Gurevych. 2021. Metaqa: Combining expert agents for multi-skill question answering. *Proceedings of The 17th Conference of the European Chapter of the Association for Computational Linguistics*.

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

Ohad Rubin and Jonathan Berant. 2021. SmBoP: Semi-autoregressive bottom-up semantic parsing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. 2021. PICARD: Parsing incrementally for constrained auto-regressive decoding from language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Tianze Shi, Chen Zhao, Jordan Boyd-Graber, Hal Daumé III, and Lillian Lee. 2020. On the potential of lexico-logical alignments for semantic parsing to SQL queries. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Chenglei Si, Weijia Shi, Chen Zhao, Luke Zettlemoyer, and Jordan Boyd-Graber. 2023. Getting more out of mixture of language model reasoning experts. In *Findings of the Association for Computational Linguistics: EMNLP 2023*.

Ananya Singha, José Cambronero, Sumit Gulwani, Vu Le, and Chris Parnin. 2023. Tabular representation, noisy operators, and impacts on table structure understanding tasks in llms. In *Table Representation Learning Workshop at International Conference on Neural Information Processing System*.

Bailin Wang, Richard Shin, Xiaodong Liu, Oleksandr Polozov, and Matthew Richardson. 2020. RAT-SQL: Relation-aware schema encoding and linking for text-to-SQL parsers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Chenglong Wang, Po-Sen Huang, Alex Polozov, Marc Brockschmidt, and Rishabh Singh. 2018. Execution-guided neural program decoding. *CoRR*, abs/1807.03100.

Yunhu Ye, Binyuan Hui, Min Yang, Binhua Li, Fei Huang, and Yongbin Li. 2023. Large language models are versatile decomposers: Decompose evidence and questions for table-based reasoning. In *Special Interest Group on Information Retrieval*.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2018. Spider: A large-scale human-labeled

dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Tao Yu, Rui Zhang, Michihiro Yasunaga, Yi Chern Tan, Xi Victoria Lin, Suyi Li, Heyang Er, Irene Li, Bo Pang, Tao Chen, Emily Ji, Shreya Dixit, David Proctor, Sungrok Shim, Jonathan Kraft, Vincent Zhang, Caiming Xiong, Richard Socher, and Dragomir Radev. 2019. SParC: Cross-domain semantic parsing in context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Yilun Zhao, Linyong Nan, Zhenting Qi, Rui Zhang, and Dragomir Radev. 2022. ReasTAP: Injecting table reasoning skills during pre-training via synthetic reasoning examples. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

Yilun Zhao, Chen Zhao, Linyong Nan, Zhenting Qi, Wenlin Zhang, Xiangru Tang, Boyu Mi, and Dragomir Radev. 2023. RobuT: A systematic study of table QA robustness against human-annotated adversarial perturbations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*.

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

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*.

A Evaluation Implementation Details

To fully utilize the table-question-answer triplets from WTQ and SQL annotations from SQUALL, we augmented the random splits generated by SQUALL with additional WTQ samples that were not annotated within SQUALL. In the evaluation, we used the split of `train-1` for fine-tuning Text-to-SQL and the corresponding augmented split to fine-tune E2E TQA. Then, both fine-tuned models are evaluated by the augmented `dev-1` set. Specifically, the training set comprises 11,340 WTQ samples, with SQL annotations present in 9,032 of them. As for WIKISQL, we employed the full dataset with 56,640 table-question-SQL query training samples. Answers were extracted following the approach outlined in (Liu et al., 2022). We used the default split for the evaluation, named as `train-0` and `dev-0`. For model fine-tuning, we maintained the same parameters as original papers, running 50 and 10 epochs for WTQ and WIKISQL and selecting the best checkpoint based on the validation accuracy.

B Statistics of Error Cases

We analyse 200 error cases for Text-to-SQL and E2E TQA models. The detailed breakdown is shown in Figure B. The remaining percentage points correspond to other errors, such incorrect labels.

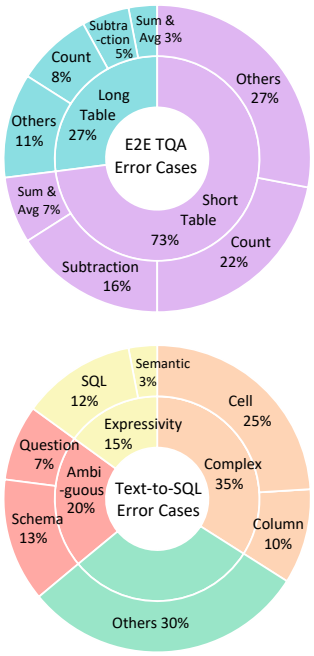

Figure 5: Breakdown of E2E TQA error cases (top) and Text-to-SQL error cases (bottom).

C Table Size Impact Analysis

This section analyzes how table size, measured by row numbers, influences the performance of various methods on WTQ. We investigate the impact of row count on the average accuracy of E2E TQA, Text-to-SQL, and an ensemble approach. Our findings reveal a consistent trend of decreasing accuracy as the number of rows increases. Notably, E2E TQA experiences a more pronounced decline in accuracy compared to Text-to-SQL. Traditional Text-to-SQL methods typically rely solely on table schema for SQL generation, leading to consistent accuracy regardless of the number of rows. However, the decline of Text-to-SQL accuracy shown in Figure 3 implies that table content may also play a role in SQL generation. Besides, it is evident that E2E TQA deteriorates much more severely than Text-to-SQL which can be attributed to the lack of the retrieval system (i.e., table row and column selection) and the the complexity of handling long-context data. Last but not least, the ensemble approach is observed to be effective to mitigate the accuracy drop caused by the table size.

D LLM-based Table QA Models

This section presents the evaluation of LLM-based E2E TQA and Text-to-SQL models. To optimize the cost, we use `gpt-3.5-turbo-0125` for all models including E2E TQA, Text-to-SQL, and SYNTQA (GPT) selector. For LLM-based E2E TQA, we follow the direct prompting (zero-shot) approach implemented by (Liu et al., 2024). For LLM-based Text-to-SQL, we incorporate 8 examples from `dev` set in the prompting to showcase the target style of SQL queries.

Table 2 demonstrates that GPT-3.5 exhibits limited proficiency in table understanding, as evidenced by significantly lower accuracy of both GPT-based Text-to-SQL and E2E TQA models compared to fine-tuned small models. However, it is evident that GPT-based Text-to-SQL and E2E TQA models also response correctly to different questions, mirroring the findings observed between T5 and OmniTab. The gap between the oracle accuracy and individual model accuracy suggests the substantial improvement potential by aggregation.

Model	WTQ	WikiSQL
Text-to-SQL Models		
T5	67.6	90.8
GPT	50.0	82.2
E2E TQA Models		
OmniTab	66.3	88.3
GPT	56.8	62.6
Ensemble Models		
SYNTQA (GPT)	65.2	84.4
SYNTQA (Oracle)	75.2	87.6

Table 2: Accuracy on subsets of WTQ and WIKISQL. SYNTQA aggregates LLM-based Text-to-SQL and E2E TQA models via the LLM-based selector. Oracle result indicates the maximum potential of mixing LLM-based Text-to-SQL and E2E TQA models.

E Feature-based Selector Implementation

E.1 Classifier Features

Below we list all the features used to train our random forest classifier for selecting the final output answer based on model predictions.

- **Question Characteristics**: question word, question length, and the number of numerical values in the question.
- **Table Characteristics**: the number of rows and columns in the table, the number of overlap words between the table header and question, and a boolean value implying whether the table is truncated in the model input.
- **Text-to-SQL Answer Characteristics**: with regard to the predicted and revised SQL query, it includes the generation probability normalized by length, and the number of preprocessed columns used in the query (e.g., `_parsed`, `_first`, and `_list` in SQUALL); concerning the queried answers from the table, it consists of the query execution status (i.e., successful or not), the number of queried answers, and the data types of queried answers (i.e., string or number).
- **E2E TQA Answer Characteristics**: the generation probability normalized by length, the number of predicted answers, answer data types, a boolean value indicating whether the E2E TQA answer is a sub-string of the Text-to-SQL answer, and another boolean indicator checking if the E2E TQA answer is a sub-string of the model input.

E.2 Training Details

Error case samples, where one model is correct and the other one is erroneous, are essential for effectively training the random forest classifier. Thus, we trained one Text-to-SQL model and one E2E TQA model at a time for each dataset splitting (in total 5 splits). We gathered error cases from each validation set. As WIKISQL does not provide 5 random splits as SQUALL, 4 additional unique `dev` sets with a similar amount of samples as the original `dev` set were extracted from the `train` set.

E.3 Comparisons Among Classifiers

We investigate various classic classification methods for answer selection in SYNTQA: linear regression (LR), k-nearest neighbors (kNN), support vector machine (SVM), multilayer perceptron (MLP), and random forest (RF). As shown in Table 3, RF attains the best performance in answer selection.

Model	LR	kNN	SVM	MLP	RF
Accuracy	70.8	66.8	70.1	70.0	**71.6**

Table 3: Classification accuracy of different machine learning methods in SYNTQA on the test set of WTQ dataset. The best performance is highlighted in **bold**.

F Heuristic Enhanced SYNTQA (GPT)

Apart from the direct prompting approach presented in the paper, we also develop a heuristic-enhanced prompting strategy for SYNTQA (GPT) and test it with `gpt-4-0125-preview`. The main idea is to leverage additional LLM-based modules to reduce the the necessity of complex reasoning on the question, table, and answer candidates. The designed heuristic is demonstrated in Figure 6 and the full prompts refer to the following subsections. As a result, the heuristic-enhanced prompting strategy achieves 89% and 87.1% accuracy in selecting the correct answer on WTQ and WIKISQL respectively. Correspondingly, it attains Table QA accuracy of 74.4% and 93.6% on WTQ and WIKISQL, further elevating the SOTA Table QA performance.

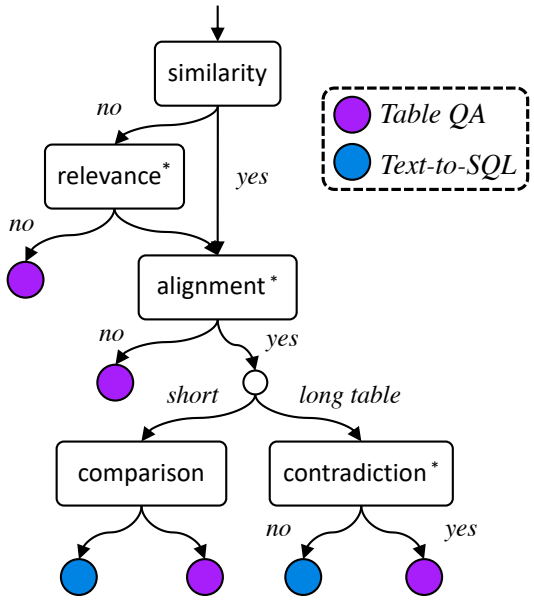

Figure 6: Design of LLM-based Selector. *Similarity* module examines if Text-to-SQL and E2E TQA answers are similar entities. *Relevance* module checks if the Text-to-SQL answer is relevant to the question. *Alignment* module inspects if the number of entities in Text-to-SQL answer corresponds to the question. *Comparison* module chooses the correct answer from two models. *Contradiction* module identifies if there is contradiction between the truncated table and Text-to-SQL answer (*indicates only using the Text-to-SQL answer).

F.1 Similarity Module

Assess the Response A and the Response B to the Question. Answer yes if the Response A and the Response B belong to the similar type of entities related to the Question, or no if they are distinct type of things. Answer yes if they are both names.

[QUESTION] which team did liverpool play against?

[RESPONSE A] coventry city

[RESPONSE B] new england patriots (4)

[ANSWER] yes

F.2 Relevance Module

Assess whether the Response contains the entities or answers asked by the Question. Answer yes if the Question asks for names or persons. Answer yes if the Response is relevant to the Question, or no if the Response is not expected by the Question. It does not matter if the response is correct or not.

[QUESTION] which city has the largest number of people in camarines sur?

[RESPONSE] 25px

[ANSWER] no

F.3 Alignment Module

Assess whether the Response contains the entities or answers asked by the Question. Answer yes if the Question asks for names or persons. Answer yes if the Response is relevant to the Question, or no if the Response is not expected by the Question. It does not matter if the response is correct or not.

[QUESTION] which city has the largest number of people in camarines sur?

[RESPONSE] 25px

[ANSWER] no

F.4 Comparison Module

You will get a table, a question, and two responses.
Based on this table, choose the more correct answer from A or B.
If A and B are the same, choose the one that is more natural to the question and favored by humans.
If neither response is correct, choose A.
If the table does not have enough information, choose A.
Let's think step by step, and then give the final answer.
Ensure the final answer format is either "Final Answer: A" or "Final Answer: B", no other form.

[TABLE] header : date announced | plant name | employees row 1 : january 23, 2006 | st. louis assembly | 1445 row 2 : january 23, 2006 | atlanta assembly | 2028 row 3 : january 23, 2006 | batavia transmission | 1745 row 4 : january 23, 2006 | windsor casting | 684 row 5 : total | total | 5902

[QUESTION] how many plants have at least 1,500 employees?

[RESPONSE A] 3

[RESPONSE B] 2

[ANSWER] to find the number of plants with more than 1500 people employees, we need to look at the employees column and count the entries that exceed 1500.
1. st. louis assembly (1445<1500)
2. atlanta assembly (2028>1500)
3. batavia transmission (1745>1500)
4. windsor casting (684<1500)
5. total (5902>1500)
because total is an aggregation amount not a real plant, the plants have at least 1500 employees are:
- atlanta assembly
- batavia transmission
that makes a total of 2 plants.
final answer: B

F.5 Contradiction Module

We implemented one type of contradiction scenarios regarding the question for counting entities in the table when candidate answers are small integer numbers. In the event that this module detects a higher count of entities within the truncated table than reflected in the Text-to-SQL response, it is deemed a contradiction, indicating a high likelihood of errors within the response.

[TABLE] header : canton | district | establish date row 1 : esch-sur-alzette | luxembourg | 4 august 1907 row 2: diekirch | diekirch | 4 august 1907 row 3: vianden | diekirch | 1 may 1922

[QUESTION] how many cantons are established in 1907?

[ANSWER] 2

G Integrating Self-Consistency

Following (Liu et al., 2024), we incorporate the Self-Consistency in our Text-to-SQL and E2E TQA base models. To generate 5 candidate answers for each model, we perturb the input table schema for the Text-to-SQL model and conduct top-k sampling ($k = 50$) for the E2E TQA model. Among five candidates, we choose one following the rule of maximum voting. Lastly, our RF classifier determined the final answer based on designed features.

Model	**WTQ**
Text-to-SQL Models	
T5	64.7
T5 + SC	65.2
E2E TQA Models	
OmniTab	62.6
OmniTab + SC	62.9
Ensemble Models	
SYNTQA (RF)	71.6
SYNTQA (RF) + SC	71.8

Table 4: Accuracy on WTQ test set. Self-Consistency can further improve the performance of both individual models and the ensemble model (SC: Self-Consistency).

H Robustness Analysis

Level	Perturbation Type	Text-to-SQL		E2E TQA		SYNTQA (RF)	
		ACC	R-ACC	ACC	R-ACC	ORACLE	ACC
Table Header	Synonym Replacement	84.7 / 72.6 (-12.1)	82.9	84.7 / 73.0 (-11.7)	83.4	93.1 / 86.5 (-6.6)	**79.6** (+6.6)
	Abbreviation Replacement	84.4 / 76.2 (-8.2)	87.0	84.2 / 74.3 (-9.9)	85.7	92.9 / 87.5 (-5.4)	**81.2** (+5.0)
Table Content	Column Extension	89.6 / 48.7 (-40.9)	52.9	91.6 / 54.8 (-36.8)	59.1	95.5 / 58.5 (-37.0)	**56.3** (+1.5)
	Column Adding	81.0 / 79.7 (-1.3)	94.7	81.5 / 70.3 (-11.2)	83.4	90.7 / 87.5 (-3.2)	**83.8** (+4.1)
Question	Word-Level Paraphrase	87.3 / 63.7 (-23.6)	70.6	88.3 / 66.0 (-22.3)	72.9	94.3 / 73.8 (-20.5)	**68.8** (+2.8)
	Sentence-Level Paraphrase	83.6 / 71.5 (-12.1)	81.3	83.8 / 72.3 (-11.5)	83.1	92.2 / 83.7 (-8.5)	**78.0** (+5.7)
Mix	—	87.0 / 60.3 (-26.7)	66.8	88.5 / 63.4 (-25.1)	69.5	94.0 / 72.0 (-22.0)	**66.8** (+3.4)

Table 5: Robustness evaluation results of Text-to-SQL, E2E TQA, and SYNTQA models on ROBUT-WIKISQL. ACC represents the *Pre-* and *Post-perturbation Accuracy*; R-ACC represents the *Robustness Accuracy*. **Bold** numbers indicate the highest *Post-perturbation Accuracy* in each perturbation type. Red numbers show the accuracy degeneration due to the perturbation. Green numbers demonstrate the improvement over the best individual model.