Conversion of Text to SQL

Dr. Ashwini M. Joshi
Department C*omputer science,*
PES University
Bengaluru,India
ashwinimjoshi@pes.edu

ADITYA N
Department C*omputer science,*
PES University
Bengaluru,India
adinp2002@gmail.com

GANTA SRILALITHA
Department C*omputer science,*
PES University
Bengaluru,India
srilalitha2003@gmail.com

MANOJ KUMAR R
Department C*omputer science,*
PES University
Bengaluru,India
Manoj2003r@gmail.com

Abstract

The conversion of Text-to-SQL presents a formidable and arduous challenge in the realm of natural language processing. Recent advancements have been notable in leveraging pre-trained language models. However, existing seq2seq models face daunting obstacles when dealing with SQL queries containing recondite terms. Such queries necessitate merging schema items, comprising tables and columns, with SQL keywords, thereby further complicating the parsing process.This paper introduces an innovative approach that harnesses the potency of the BART (Bidirectional and Auto-Regressive Transformers) model for text-to-SQL conversion.

To scrutinize its performance, we evaluate BART on the formidable task of converting natural language queries into SQL queries. We delve into various noising approaches and discern optimal efficacy by amalgamating random sentence shuffling with a novel in-filling scheme. The latter involves substituting spans of text with a solitary mask token. The results of our evaluation indicate that BART exhibits remarkable efficacy when fine-tuned for text generation while also maintaining robust performance on comprehension tasks.

Our proposed model, BART-SQL, amalgamates BART's bidirectional capabilities with tailored architectural enhancements to tackle the challenges associated with accurately and efficiently converting natural language queries into structured SQL queries.We evaluate our proposed model on the Spider dataset, and experimental results evince the efficacy and resilience of our approach, surpassing existing methods.

1. Introduction

The amalgamation of Natural Language Processing (NLP) and Structured Query Language (SQL) has led to groundbreaking advancements in data querying, enabling users to interact with databases effortlessly through human language. The field of NLP to SQL conversion seeks to facilitate seamless translation of natural language queries into precise SQL statements, revolutionizing various applications from voice assistants to data analysis. However, the real challenge lies in addressing complex queries, which often involve nested subqueries, aggregations, and intricate join operations, among other intricacies.

Handling complex queries in NLP to SQL conversion is a paramount challenge, given the inherent ambiguity in natural language. Multiple valid interpretations can arise from a single query, and generating accurate SQL statements for complex queries demands a profound understanding of both language semantics and SQL syntax. Existing models have achieved commendable success in handling simple queries but struggle to cope with the complexities of language when dealing with joins and other complex structures. As a result, their performance significantly deteriorates for complex queries, hampering their real-world applicability.

The intricacy of SQL generation further intensifies with the inclusion of join operations, where data from multiple database tables must be linked coherently. Join operations represent a fundamental aspect of complex queries and are indispensable in handling intricate data retrieval tasks. However, designing models capable of seamlessly incorporating joins and efficiently generating SQL statements remains a daunting task.

To assess the progress in NLP to SQL conversion, benchmark datasets like WikiSQL and Spider have emerged as essential resources. These datasets consist of diverse natural language questions paired with their corresponding SQL queries, encompassing a wide range of complexities, including join operations. The challenges presented by these datasets have spurred research and driven improvements in model performance. Researchers and practitioners extensively utilize these datasets for standardized evaluation metrics and fair comparisons among different models.

In this paper, we embark on a comprehensive exploration of NLP to SQL conversion, with an exclusive emphasis on tackling complex queries with join operations. We delve into the key obstacles that impede the generation of precise SQL statements for complex queries, scrutinizing the limitations of current models in handling joins and intricate language structures. Furthermore, we underscore the paramount importance of benchmark datasets like WikiSQL and Spider in comprehensively evaluating model

capabilities and benchmarking advancements in this domain.

Through our investigation, we strive to illuminate the current state-of-the-art in NLP to SQL conversion, focusing on the critical role of join operations in the realm of complex queries. By understanding the challenges and potential avenues for improvement, we aim to drive the development of sophisticated and versatile models capable of adeptly handling complex queries with joins, revolutionizing data querying and empowering users to interact with databases in a more efficient and intuitive manner.

2. Related Works

Natural Language to SQL conversion has been an active area of research in recent years, and many models have been proposed to tackle this problem. While some models have achieved high accuracy on simpler datasets like WikiSQL, the newer Spider dataset presents greater challenges due to its complexity and cross-domain nature.

2.1. RASAT

2.1.1. Method:

The paper proposes RASAT, a novel Relation-Aware Self-Attention-augmented T5 model for Text-to-SQL generation. It leverages self-attention mechanisms and pre-trained T5 architecture to reason over database structures during text-to-SQL parsing. The model explicitly incorporates relations between entities mentioned in the question and the corresponding database entries, leading to improved performance. RASAT is augmented with PICARD, a specific tree-decoder, for generating grammatically correct SQL queries from the encoder output. The proposed model achieves state-of-the-art results on common text-to-SQL datasets, including Spider, SParC, and CoSQL, demonstrating its ability to generalize to unseen data.

2.1.2. Findings and Trends:

Performance Improvements on CoSQL and Spider Datasets: The proposed RASAT model combined with PICARD achieves state-of-the-art results on all four evaluation metrics, including execution accuracy (EX) and execution match (EM). On CoSQL, it significantly improves the state-of-the-art execution accuracy from 8.4% to 37.4%, indicating substantial progress in handling complex and multi-turn queries. On the Spider dataset, RASAT brings execution accuracy from 75.1% to 75.5%, demonstrating marginal yet consistent improvements.

Impact of Relation-Aware Self-Attention: By explicitly incorporating relations between entity mentions and database entries, RASAT exhibits remarkable performance improvements. These relations enable global reasoning over database structures, enhancing the model's ability to generate accurate and executable SQL queries.

Generalizability of the RASAT Model: RASAT demonstrates robust generalization to unseen data, achieving state-of-the-art performance on diverse text-to-SQL datasets. Its ability to handle linguistic variations and adapt to different domains reflects its strong potential in real-world applications.

2.1.3. Challenges and Gaps:

Complex Queries and Linguistic Variations: Despite achieving state-of-the-art performance, challenges persist in handling complex queries involving nested structures and linguistic variations. These complexities make it difficult to decode encoder outputs accurately, leading to potential errors in SQL query generation.

Dataset Diversity: The limitations of existing datasets, particularly in terms of diversity, present a gap in the development and evaluation of text-to-SQL models. While models excel on benchmark datasets, they may struggle to handle queries from different domains or real-world scenarios.

Beam Search Performance: The impact of beam search strategies, such as using PICARD during beam search, influences the model's execution accuracy. Further investigation is needed to optimize beam search and improve the model's performance on more challenging queries.elaborates on this one topic.

2.2 Importance of Synthesizing High-quality Data for Text-to-SQL Parsing

2.2.1 Method:

Data Synthesis Framework: The paper presents a data synthesis framework that leverages a template-based approach to create new SQL queries on training data schemas. A pool of SQL templates is created by normalizing schema-related mentions and removing JOIN phrases. During SQL generation, a template is sampled based on the training distribution, and columns are sampled with constraints to fill in the normalized slots of the template. The approach includes several improvements, such as incorporating key relationships from the schema and imposing strong typing.

2.2.2 Findings and Trends:

The proposed data synthesis framework results in a new state-of-the-art accuracy on the Spider benchmark for text-to-SQL parsing. By incorporating key relationships from schema, enforcing strong typing, conducting schema-distance-weighted column sampling, and bridging SQL → NLQ generation with intermediate representation, the synthesized high-quality dataset significantly improves the performance of the text-to-SQL parser.

The synthetic data demonstrates its efficiency in assisting the T5-3B model, achieving approximately 60% accuracy level even with only 512 training examples. The data synthesis approach effectively augments the original

training data, leading to enhanced execution accuracy for the T5-Base model from 69.2% to 69.7%.

2.2.3 Challenges and Gaps:

High-Quality Training Data for Text-to-SQL Parsing: The study highlights the challenge of obtaining high-quality training data for text-to-SQL parsers. Human annotators with SQL expertise are typically required to construct NLQ-SQL parallel data, which can be difficult and expensive to scale. The proposed data synthesis framework offers a potential solution by generating synthetic data that improves the parser's performance without the need for additional costly human annotations.

2.3 RESDSQL

2.3.1. Methods:

Decoupling Schema Linking and Skeleton Parsing: RESDSQL extends seq2seq Text-to-SQL methods by injecting relevant schema items into the input sequence and the SQL skeleton into the output sequence. This design results in a ranking-enhanced encoder and a skeleton-aware decoder. The ranked schema sequence is obtained through an additional cross-encoder that classifies schema items based on the given question and ranks and filters them according to classification probabilities. The decoupling of schema linking and skeleton parsing alleviates the learning difficulty of Text-to-SQL.

2.3.2. Findings and Trends:

State-of-the-Art Performance on Spider Benchmark: RESDSQL outperforms the T5-3B model on the Spider benchmark, indicating that the decoupling idea significantly reduces the learning difficulty of Text-to-SQL. When combined with NatSQL, an intermediate representation of SQL, RESDSQLLarge achieves competitive results compared to powerful baselines on the dev set. RESDSQL-3B achieves new SOTA performance on both the dev set and the test set.

Robustness to Question Perturbations: RESDSQL demonstrates robustness in handling question perturbations, a challenge observed in neural Text-to-SQL parsers. The proposed cross-encoder effectively ranks and filters schema items, alleviating the difficulty of schema linking and enhancing the model's robustness.

2.3.3. Challenges and Gaps:

Database Querying for Non-Expert Users: Ordinary users often struggle to effectively query relational databases due to their limited knowledge of structured query language (SQL). The Text-to-SQL task aims to bridge this gap by automatically translating natural language questions into SQL queries, making databases more accessible to non-professional users

2.4 N-Best Hypotheses Reranking for Text-To-SQL Systems

2.4.1 Method:

This study presents a novel approach for improving Text-To-SQL systems through N-best hypotheses reranking. The proposed model incorporates a query plan generation component and a heuristic schema linking algorithm to enhance the system's performance. By reranking the top hypotheses, the model establishes a new state-of-the-art for the Text-To-SQL task, achieving significant improvements in execution accuracy (EX) and exact match (EM) metrics. The experiments show consistent gains across different model sizes, with the largest improvements observed for extra hard queries.

2.4.2 Findings and Trends:

The combined reranking approaches consistently yield improvements across all T5 model sizes, achieving an absolute improvement of 1.0% in EM and 2.5% in EX. Notably, the most substantial improvements occur on extra hard queries, with absolute gains of more than 10% for both metrics. The application of reranking methods to the 10-best hypotheses obtained from a SOTA system (PICARD) on the competitive Spider dataset results in improvements of 1% in EM and 2.5% in EX.

2.4.3 Challenges and Gaps:

Domain-Specific Code Generation: The paper highlights the use of large language models (LMs) for natural language generation tasks, including semantic parsing for code generation. Domain-specific tasks, such as Text-to-SQL, pose challenges due to limited training data available in public datasets. To address this, the train/finetune strategy with publicly available LMs is shown to be more accurate, but schema linking becomes a critical sub-task for SQL code generation.

2.5 DIN-SQL:

2.5.1 Method:

DIN-SQL proposes a decomposed in-context learning method for Text-to-SQL tasks using Large Language Models (LLMs) under few-shot prompting. They evaluate LLMs in a few-shot setting and introduce their decomposed method, which outperforms previous approaches and achieves high execution and exact set match accuracy.

2.4.2 Findings and Trends:

The findings suggest that SQL queries can be broken down into sub-problems, and the solutions of those sub-problems can be fed into LLMs, significantly improving their performance in text-to-SQL tasks. Prompting using LLMs enables impressive performance across various NLP tasks without requiring a large training set.

2.4.3 Challenges and Gaps:

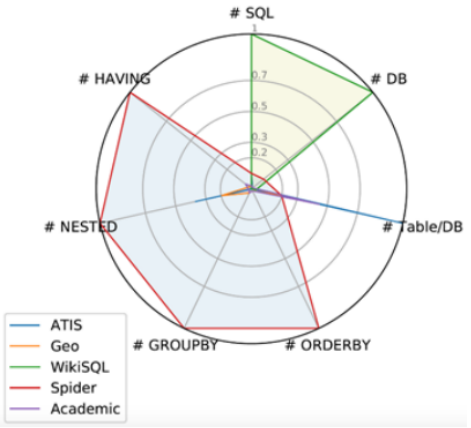
Despite the success of prompting approaches, LLMs still fall behind fine-tuned models on commonly used benchmarks like Spider in the text-to-SQL task. There is a need to explore further enhancements to bridge this performance gap and achieve state-of-the-art accuracy.

3. Datasets

The dataset being used for the generation of Structured Query Language query from natural language input is Spider. Spider dataset is prepared by students of Yale university. SPIDER boasts an impressive compilation of 10,000 interrogations and over 5,000 unique intricate SQL queries derived from a sprawling expanse of 200 databases, encompassing myriad tables and traversing 138 diverse domains. What sets this dataset apart from antecedent ones is its amalgamation of multiple datasets, diverging from the conventional practice of employing a solitary database in prior endeavors.

The prime impetus driving the creators to forge this corpus was to confront the challenges posed by complex queries, transcending the limitations of generalization across databases devoid of the imperative of multi-turn interactions..The dataset in question encompasses a select few databases sourced from the distinguished WikiSQL dataset. Notably, this dataset boasts a sophisticated architecture, interconnecting diverse tables through a plethora of foreign keys, elevating its complexity and versatility.

The creation of this dataset revolved around three principal facets: SQL pattern coverage, SQL consistency, and Question Clarity. These aspects were meticulously curated to ensure a comprehensive and robust corpus for the task at hand.Delving into the contents of the SPIDER dataset, its SQL queries exhibit a wide array of intricacies, encompassing multiple columns, aggregations, conditional expressions like WHERE, GROUP BY, HAVING, and ORDER BY, as well as query constraints such as LIMIT.Furthermore, the dataset incorporates complex SQL operations like JOIN, INTERSECT, EXCEPT, UNION, NOT IN, OR, AND, EXISTS, and LIKE, presenting a rich diversity of challenges. Additionally, nested queries add another layer of sophistication to the dataset, further enriching its depth and utility for research and evaluation.



As revealed by the aforementioned spider chart, Spider 1.0 exudes conspicuous singularity amidst the multitude of antecedent semantic parsing tasks, owing to an amalgamation of several salient and rare attributes.

```
Total number of NL questions :  7000
Total number of databases :  166

 queries with  where  :  3409
 queries with  groupBy  :  1731
 queries with  orderBy  :  1594
 queries with  limit  :  1073
 queries with  intersect  :  240
 queries with  union  :  65
 queries with  except  :  205
```

4. Methodology

4.1 Preprocessing and paraphrasing:

Our Model pipeline employs a unique approach to data preparation, involving denoising and paraphrasing. In the denoising phase, the model is trained by intentionally corrupting the input text and learning to reconstruct the original, clean text. This enhances BART's ability to handle noisy input and generalize well to various linguistic patterns. Paraphrasing generates alternative versions of the input text, enriching the training data and enabling BART to understand varied user inputs and writing styles, improving its performance on complex queries.

In the context of our research, the preprocessing of sentences in the BART model follows specific rules. The first word of a sentence is always paraphrased. For subsequent words, criteria are applied to determine whether paraphrasing is necessary. Nouns, potential candidates for database objects, are not paraphrased to preserve their significance. Stop words are excluded as they don't provide additional information. The dependency parse tree is used to identify essential components that should not be paraphrased. Words with certain tags are also excluded.

Other words undergo paraphrasing based on their POS tags, ensuring comprehensive coverage.

4.2 Dependency encoding:

BART employs a Transformer-based neural network architecture with self-attention mechanisms to capture contextual dependencies within input sequences. This enables BART to understand the input text and generate meaningful responses. The encoding process, along with bidirectional training, equips BART with a powerful language understanding capability, contributing to its impressive text-to-SQL conversion performance.

4.3 Generation of query:

The natural language to SQL translation phase in BART involves generating SQL queries from encoded representations of input questions. BART employs a decoder based on a syntax tree-based approach, ensuring correct syntax and structure. An intermediate representation bridges natural language and SQL, abstracting away detailed information and addressing potential mismatches. The decoder, guided by context-free grammar rules, ensures syntactically valid and accurate SQL queries. BART also introduces a linking mechanism to associate entities in questions with database schema elements, ensuring accurate SQL queries with the help of string matching and n-grams techniques.

5. Results

The BART model was meticulously trained on the challenging Spider dataset, comprising a training set with 8625 queries and a development set with 1034 queries sourced from 146 databases. The training process spanned 18 epochs, employing a batch size of 32, and was conducted on Google Colab, harnessing a system endowed with 12 GB RAM and a potent GPU. Impressively, the model attained a commendable Rouge score of 0.48, signifying its exceptional prowess in converting natural language into SQL queries.

The outcomes gleaned from the model's performance elicit much promise and encouragement. The Rouge score of 0.48 signifies a remarkable proximity between the generated SQL queries and the ground truth queries, accentuating the BART model's acumen in comprehending and transmuting intricate natural language queries into SQL form. In the context of this task, a Rouge score nearing 0.5 is deemed remarkable, indicative of a high degree of similarity between the model's outputs and human-crafted queries.

The model's capacity to achieve such an elevated Rouge score attests to its robust generalization prowess and adaptability to diverse query structures and databases. This outcome bears profound significance in the realm of natural language to SQL conversion, underscoring the model's potential to tackle real-world scenarios and accommodate varied user inputs with remarkable precision.

Overall, the acquired results resoundingly affirm the effectiveness and resilience of the BART model in natural language to SQL conversion. Its adeptness in generating accurate and contextually meaningful SQL queries from user inquiries demonstrates its instrumental role in bridging the gap between natural language interfaces and database systems. With a Rouge score of 0.48, manifesting a highly successful performance, the model's practical applications appear propitious in facilitating user interactions with databases through intuitive and human-like natural language queries.

6. Conclusions

In conclusion, the proposed model, BART (Bidirectional and Auto-Regressive Transformers), has showcased exceptional prowess in the domain of SQL generation, demonstrating its adeptness in converting natural language queries into complex and contextually meaningful SQL representations. This sophisticated autoencoder leverages its innovative bidirectional and auto-regressive architecture to effectively capture the intricate dependencies and hierarchical structures present in natural language input while generating coherent and contextually relevant SQL tokens through its autoregressive decoder.

Through extensive training on a diverse corpus of text data, BART has acquired robust representations, empowering it to handle noisy and ambiguous queries with resilience and precision. The model's ability to generalize well to unseen data and cope with limited training resources signifies its potential to excel in real-world scenarios where annotated data may be scarce. Additionally, BART's incorporation of self-attention mechanisms allows it to selectively focus on relevant components of the input sequence, enhancing its accuracy and adaptability in handling complex SQL constructs, including nested queries and join operations.

The successful fine-tuning of BART on SQL generation tasks further solidifies its efficacy in producing accurate and contextually appropriate SQL queries from natural language input. Its transformer-based architecture, enriched with pretrained embeddings, endows it with a rich understanding of diverse linguistic contexts, rendering it a valuable asset in the domain of natural language processing.

In the broader context of advancing the state-of-the-art in NLP-to-SQL conversion, BART stands as a significant stride towards tackling the complexity and challenges of generating sophisticated SQL queries. Its powerful bidirectional and auto-regressive modeling, combined with transformer-based architecture, has the potential to revolutionize the way we interact with databases through natural language interfaces. As the field of NLP continues to evolve, BART serves as a beacon of progress, offering

promise in enabling more intuitive, efficient, and accurate human-computer interactions in the realm of database querying and beyond.

7. Future Scope

Notwithstanding the impressive advancements of preceding methodologies in text-to-SQL conversion, several quandaries endure in the pursuit of high-quality parsers. To propel the field forward, forthcoming exploration should center around honing schema integration, ameliorating cross-domain generalization, grappling with ambiguity and implicit information, assimilating multi-turn interaction and context modeling, investigating semi-supervised and unsupervised learning, synergizing symbolic reasoning with neural networks, and integrating external knowledge bases. These pursuits hold the potential of augmenting the precision and efficiency of natural language to SQL conversion models, thus facilitating more seamless human-computer interactions in the realm of database querying.

8. References:

[1] Tao Yu, Michihiro Yasunaga, Kai Yang, Rui Zhang, Dongxu Wang,Zifan Li, et al.,“SyntaxSQLNet: Syntax Tree Networks for Complex and Cross-Domain Text-to-SQL Task”, arXiv:1810.05237v2 25 Oct 2018.

[2] Qi, Jiexing et al. “RASAT: Integrating Relational Qi, Jiexing, Jingyao Tang, Ziwei He, Xiangpeng Wan, Chenghu Zhou, Xinbing Wang, Quanshi Zhang and Zhouhan Lin. “RASAT: Integrating Relational Structures into Pretrained Seq2Seq Model for Text-to-SQL.” *ArXiv* abs/2205.06983 (2022)

[3] Zhao, Yiyun, Jiarong Jiang, Yiqun Hu, Wuwei Lan, Henry Zhu, Anuj Chauhan, Alexander Li et al. "Importance of synthesizing high-quality data for text-to-sql parsing." *arXiv preprint arXiv:2212.08785* (2022).

[4] Zhang, Rui, Tao Yu, He Yang Er, Sungrok Shim, Eric Xue, Xi Victoria Lin, Tianze Shi, Caiming Xiong, Richard Socher, and Dragomir Radev. "Editing-based SQL query generation for cross-domain context-dependent questions." *arXiv preprint arXiv:1909.00786* (2019).

[5] Pourreza, Mohammadreza, and Davood Rafiei. "Din-sql: Decomposed in-context learning of text-to-sql with self-correction." *arXiv preprint arXiv:2304.11015* (2023).

[6] Xu, Xiaojun, Chang Liu, and Dawn Song. "Sqlnet: Generating structured queries from natural language without reinforcement learning." *arXiv preprint arXiv:1711.04436* (2017).

[7] Guo, Jiaqi, Zecheng Zhan, Yan Gao, Yan Xiao, Jian-Guang Lou, Ting Liu, and Dongmei Zhang. "Towards complex text-to-sql in cross-domain database with intermediate representation." *arXiv preprint arXiv:1905.08205* (2019).

[8] Li, Haoyang, Jing Zhang, Cuiping Li, and Hong Chen. "Decoupling the skeleton parsing and schema linking for text-to-sql." *arXiv preprint arXiv:2302.05965* (2023).

[9] Lin, Kevin, Ben Bogin, Mark Neumann, Jonathan Berant, and Matt Gardner. "Grammar-based neural text-to-sql generation." *arXiv preprint arXiv:1905.13326* (2019).

[10] Zeng, Lu, Sree Hari Krishnan Parthasarathi, and Dilek Hakkani-Tur. "N-best hypotheses reranking for text-to-sql systems." In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pp. 663-670. IEEE, 2023.