Сифэй, Лай. Обучение модели RASAT, интегрирующей реляционную структуру в предварительно обученную модель Seq2Seq для преобразования текста в SQL / Лай Сифэй // Прикладная математика и вопросы управления. – 2024. – № 4. – С. 65–75. DOI 10.15593/2499-9873/2024.4.05

Библиографическое описание согласно ГОСТ Р 7.0.100–2018
Сифэй, Лай. Обучение модели RASAT, интегрирующей реляционную структуру в предварительно обученную модель Seq2Seq для преобразования текста в SQL / Лай Сифэй. – Текст : непосредственный. – DOI 10.15593/2499-9873/2024.4.05 // Прикладная математика и вопросы управления / Applied Mathematics and Control Sciences. – 2024. – № 4. – С. 65–75.

пермский политех

ПРИКЛАДНАЯ МАТЕМАТИКА И ВОПРОСЫ УПРАВЛЕНИЯ
№ 4, 2024
https://ered.pstu.ru/index.php/amcs

Научная статья
DOI: 10.15593/2499-9873/2024.4.05
УДК 004.89

Обучение модели RASAT, интегрирующей реляционную структуру в предварительно обученную модель Seq2Seq для преобразования текста в SQL

Сифэй Лай

Новосибирский Государственный университет, Новосибирск, Российская Федерация

О СТАТЬЕ

Получена: 15 ноября 2024
Одобрена: 24 декабря 2024
Принята к публикации:
28 декабря 2024

Финансирование
Исследование не имело спонсорской поддержки.
Конфликт интересов
Автор заявляет об отсутствии конфликта интересов.
Вклад автора
100 %.

Ключевые слова:
relational attention module; pre-trained model; Transformer Seq2Seq; text to SQL conversion; Spider dataset.

АННОТАЦИЯ

В данном исследовании модуль реляционного внимания интегрируется в предобученную модель Transformer Seq2Seq и осуществляется преобразование вопросов на естественном языке в команды извлечения на языке структурированных запросов (SQL) с помощью экспериментов на наборе данных Spider. Цель этой научной статьи состоит в том, чтобы улучшить точность и эффективность преобразования текста в SQL-запросы, используя механизм реляционного внимания в модели трансформера. Статья представляет модель RASAT (переход SQL на основе реляционного внимания), которая заменяет модуль самовращения в энкодере трансформера на модуль реляционного внимания для обработки задач текст-к-SQL. Этот подход позволяет лучше учитывать семантические связи между сущностями в тексте и генерировать более точные SQL-запросы. Методы исследования включают использование предобученной модели трансформера (T5-small) и ее обучение на наборе данных Spider с введением модуля реляционного внимания. Экспериментальные результаты показывают значительное улучшение показателей точности при преобразовании текста в SQL по сравнению с базовой моделью без реляционного компонента. Экспериментальные результаты демонстрируют, что модель RASAT улучшает производительность по показателю Exact Match на 1,82 % и точность выполнения на 3,26 %. Эти улучшения достигнуты несмотря на то, что количество эпох обучения было ограничено 500 вместо 3072 для базовой модели, что подчеркивает эффективность предложенного подхода даже при ограниченных вычислительных ресурсах. В заключение подчеркиваются перспективы дальнейшего развития метода реляционной модели для улучшения качества систем, связанных с обработкой естественного языка и базами данных.

© **Лай Сифэй** – аспирант, e-mail: laixifei153@gmail.com.

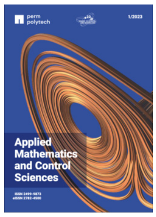
Perm Polytech Style: Xifei Lai. Train the RASAT model that integrates the relational structure into the pre-trained Seq2Seq model to convert text into SQL. *Applied Mathematics and Control Sciences.* 2024, no. 4, pp. 65–75. DOI: 10.15593/2499-9873/2024.4.05

MDPI and ACS Style: Xifei, Lai. Train the RASAT model that integrates the relational structure into the pre-trained Seq2Seq model to convert text into SQL. *Appl. Math. Control Sci.* **2024, 4**, 65–75. https://doi.org/10.15593/2499-9873/2024.4.05

Chicago/Turabian Style: Xifei, Lai. 2024. "Train the RASAT model that integrates the relational structure into the pre-trained Seq2Seq model to convert text into SQL". *Appl. Math. Control Sci.* no. 4: 65–75. https://doi.org/10.15593/2499-9873/2024.4.05

perm polytech

APPLIED MATHEMATICS
AND CONTROL SCIENCES
№ 4, 2024
https://ered.pstu.ru/index.php/amcs



Article
DOI: 10.15593/2499-9873/2024.4.05
UDC 004.89

Train the RASAT model that integrates the relational structure into the pre-trained Seq2Seq model to convert text into SQL

Xifei Lai

Novosibirsk State University, Novosibirsk, Russian Federation

ARTICLE INFO

Received: 15 November 2024
Approved: 24 December 2024
Accepted for publication: 28 December 2024

Funding
This research received no external funding.

Conflicts of Interest
The author declares no conflict of interest.

Author Contributions
100 %.

Keywords:
relational attention module; pre-trained model; Transformer Seq2Seq; text to SQL conversion; Spider dataset

ABSTRACT

In this study the relational attention module is getting integrated into pre-trained Transformer Seq2Seq model and realize the conversion of natural language questions to Structured Query Language (SQL) retrieval commands by conducting experiments on the Spider dataset. The purpose of this scientific article is to improve the accuracy and efficiency of converting text into SQL queries by using the relational attention mechanism in the transformer model. The article presents the RASAT (Relational Attention-based SQL Transformation) model, which replaces the self-rotation module in the transformer encoder with a relational attention module for processing text-to-SQL tasks. this approach allows you to better take into account the semantic relationships between entities in the text and generate more accurate SQL queries. The research methods include the use of a pre-trained transformer model (T5-small) and training it on the Spider dataset with the introduction of a relational attention module. Experimental results show a significant improvement in accuracy indicators when converting text to SQL compared to the basic model without a relational component. The experimental results demonstrate that the RASAT model improves the Exact Match performance by 1.82% and the Execution Accuracy by 3.26%. These improvements are achieved despite the fact that the number of training epochs was limited to 500 instead of 3072 for the basic model, which emphasizes the effectiveness of the proposed approach even with limited computing resources. In conclusion, the prospects for further development of the relational model method to improve the quality of systems related to natural language processing and databases are emphasized.

© **Xifei Lai** – Ph. D. student, e-mail: laixifei153@gmail.com.

Introduction

SQL (Structured Query Language) is a standardized language for managing relational databases and is widely used in database management systems. It is widely used in database management systems to query, update and manage data. The difficulty of learning SQL varies from person to person: it may be relatively easy for someone with a programming or data processing background, but for beginners, understanding the concepts and syntax of SQL may take time and practice.The complexity of SQL lies mainly in its rich syntax and functionality, including, but not limited to, a variety of querying, joining, filtering, sorting, and aggregation operations. As such, SQL is an invaluable skill that provides solid support for areas such as data analysis and management and software development.

The introduction of artificial intelligence technologies, especially deep learning and transformational modeling for natural language processing, has brought new opportunities to address the challenges of SQL learning.

Transformation models have achieved great success in the field of natural language processing due to their excellent sequence modeling capabilities and attention mechanisms. By applying the transformer model to the task of converting text to SQL, we are able to handle the complex structure and semantics of natural language and achieve efficient and accurate transformations. The converter model is able to capture remote dependencies and contextual information, resulting in more accurate generated SQL queries.

This task aims at translating natural language questions into SQL queries.

It is one of the most important semantic parsing tasks of practical significance today as it can significantly reduce the barriers for non-specialized users to interact with databases [1].

I. Related Work

Early approaches typically employ a sketch-based slot filling technique, where various modules are utilized to forecast specific segments of SQL. These methodologies break down the SQL generation task into multiple autonomous sketches and employ diverse classifiers to predict the corresponding parts. Examples include SQLNet [2], SQLOVA [3], RYANSQL [4], among others.

However, many of these approaches are limited to handling simple queries and struggle to generate accurate SQL in complex scenarios like those found in the Spider dataset.

In the realm of text-to-SQL tasks, grappling with multi-table and complex SQL scenarios has propelled the adoption of graph structures to encapsulate intricate relationships.

Illustratively, Global-GNN [5] represents the convoluted database schema through graph representations. LGESQL [6] delves deeper into distinguishing local and non-local relations by leveraging a line graph enhanced hidden module. SADGA [7] leverages contextual and dependency structures to encode question graphs, while schema graph utilizes database schema relations. S2SQL [8] enriches relational graph attention networks (RGAT) [9] with syntactic dependency information.

In 2021, Shaw et al. [10] showed that fine-tuning the pre-trained T5-3B model can produce results comparable to the state-of-the-art at the time.

Based on this finding, Scholak et al. [11] proposed to constrain the autoregressive decoder through incremental parsing during inference time, effectively filtering out dynamic syntax error sequences during the beam search process, thereby significantly improving the quality of the generated SQL.

II. Dataset and Baseline

Spider is a large-scale complex cross-domain semantic parsing and text-to-SQL dataset annotated by 11 Yale students. The goal of the Spider Challenge is to develop natural language interfaces to cross-domain databases [12].

The Spider dataset is a multi-database, multi-table, single-round query Text-to-SQL dataset, containing 10181 questions and 5693 unique complex SQL queries, 200 databases with multiple tables, covering 138 different fields. It has strong practical applications.

Why Spider?

Spider stands out from previous semantic parsing tasks depicted in the spider chart due to its unique characteristics, it occupies the largest area on the chart, marking it as the pioneer in complex, cross-domain semantic parsing and text-to-SQL datasets (shown at Fig. 1).

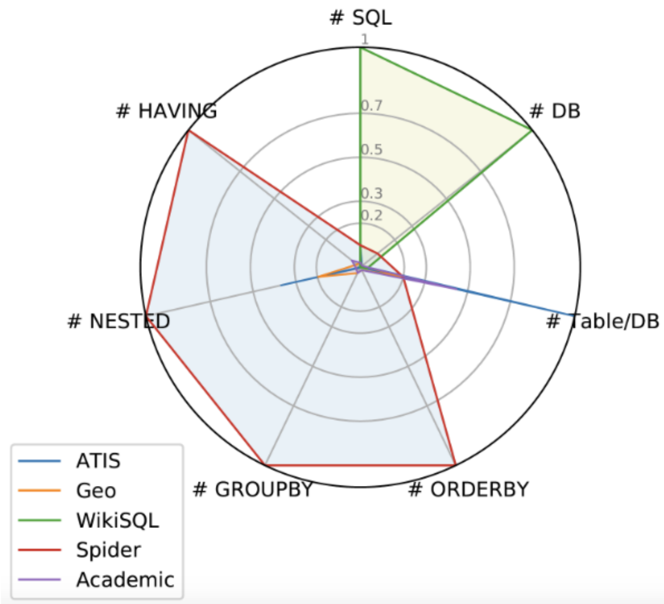


Fig. 1. Advantages of the spider dataset [18]

Database splitting:

206 databases were split into 146 trains, 20 devs, and 40 tests, with all questions from the same database in the same grouping (train/dev/test).

Only train and dev are used in this task.

To better understand the performance of the model on different queries, the SQL queries were categorized into 4 levels: easy, medium, hard, and extra hard. Difficulty is defined based on the number of SQL components, selections and conditions, including more SQL keywords such as GROUP BY, ORDER BY, INTERSECT, nested subqueries, column selections and aggregation statements would be considered as difficult.

Evaluation Metrics:

The accuracy of the system is measured based on the difficulty of the query, and an official evaluation script is released along with the corpus (shown at Fig. 2).

Exact Match: the predicted query is correct when and only when each component is correct.

Execution accuracy: exact matching may lead to false negative [not understood] evaluations, so execution accuracy is also taken into account, similarly if the returned result is the same as the standard but with different semantics, false positive errors may be reported, which can complement each other as well; and finally if there is a JOIN and a GROUP in the query statement, the then the evaluation can accept multiple keys.

Thus the execution accuracy is always a bit higher than exact matching.

Easy

What is the number of cars with more than 4 cylinders?

```
SELECT COUNT(*)
FROM cars_data
WHERE cylinders > 4
```

Meidum

For each stadium, how many concerts are there?

```
SELECT T2.name, COUNT(*)
FROM concert AS T1 JOIN stadium AS T2
ON T1.stadium_id = T2.stadium_id
GROUP BY T1.stadium_id
```

Hard

Which countries in Europe have at least 3 car manufacturers?

```
SELECT T1.country_name
FROM countries AS T1 JOIN continents
AS T2 ON T1.continent = T2.cont_id
JOIN car_makers AS T3 ON
T1.country_id = T3.country
WHERE T2.continent = 'Europe'
GROUP BY T1.country_name
HAVING COUNT(*) >= 3
```

Extra Hard

What is the average life expectancy in the countries where English is not the official language?

```
SELECT AVG(life_expectancy)
FROM country
WHERE name NOT IN
    (SELECT T1.name
     FROM country AS T1 JOIN
     country_language AS T2
     ON T1.code = T2.country_code
     WHERE T2.language = "English"
       AND T2.is_official = "T")
```

Fig. 2. SQL query examples in 4 hardness levels [18]

III. Model

The structure of the RASAT model depicted in Figure 3, is straightforward in terms of architecture. It utilizes the T5 model as its foundation, replacing the self-attention modules in the encoder with relation-aware self-attentions.

The encoder's input comprises questions (Q), a database schema (S) consisting of tables (T) and their columns (C), along with the database name (S), and mentions of database content, all accompanied by essential separators. The approach to serializing these inputs largely aligns with Shaw et al. [10] and Scholak et al. [11]. Formally, it can be expressed as:

$$X = Q|S|t_1 : c_{11}[v], \dots, c_{1|T_1|} \,|\, t_2 : c_{21}, \dots, \tag{1}$$

where t_i is the table name, c_{ij} is the j-th column name of the i-th table.

The v $\in$ V showing after column c_{11} is the database content belonging to the column that has n-gram matches with the tokens in the question.

As for delimiters, I use | to note the boundaries between Q, S, and different tables in the schema.

Within each table, I use: to separate between table name and its columns. Between each column, '|' is used as the delimiter.

For multi-turn scenarios, the model prepends previous questions to the beginning of the sequence and trim tokens from the front when the sequence reaches 512 in length. This ensures continuity while managing sequence length constraints:

$$X = \underline{Q_1 |Q_2| \dots | Q_t |S| t_1 : c_{11}[v], \dots,} \tag{2}$$

where use '|' as delimiters.

Following this, I introduced various types of relations in the form of triplets, establishing links between tokens in the serialized input, which naturally transforms the input sequence into a graph (Fig. 3). Furthermore, since almost all relation triplets have a head and tail that correspond to words or phrases, while the T5 model operates at the subword level, it also implements relation propagation to map these relations to the subword level.

To fine-tune this model, RASAT inherits all parameters from T5 and randomly initializes additional relation embeddings introduced by relation-aware self-attention. The overall increase in parameters is less than 0.01 %.

The entire model architecture inherits the seq2seq design from T5, comprising multiple layers of encoders and decoders. In the encoder, the self-attention modules are replaced with relation-aware self-attention, which incorporates two additional relation embedding lookup tables R_k and R_v. We transform the sequential input into an interaction graph by introducing various relation types and adapting them to the subword level through relation propagation. During the forward pass, the relation-aware self-attention modules access the relations of each token via the interaction graph and fetch the corresponding relation embeddings from the lookup tables R_k and R_v.

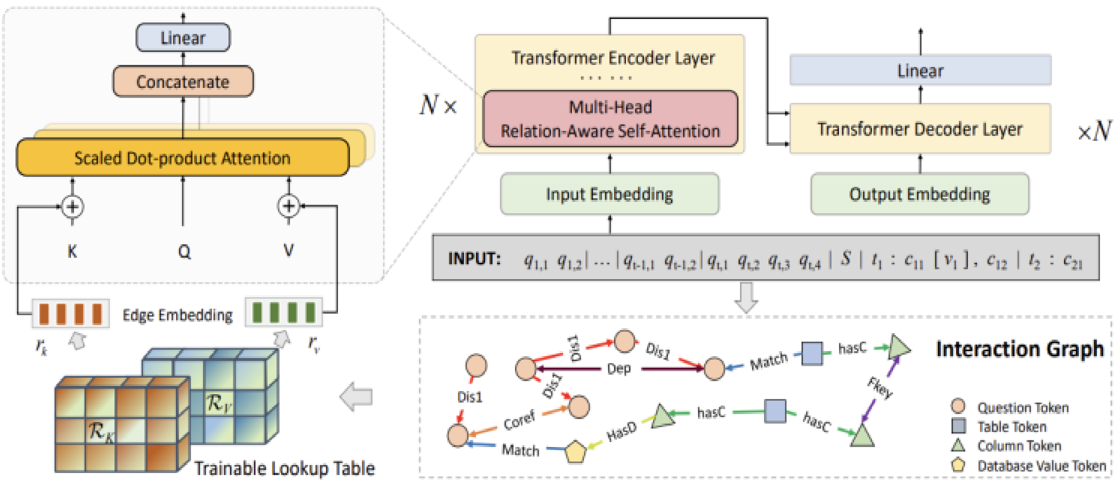


Fig. 3. The overview of the RASAT model [18]

A. Relation-aware Self-Attention: Relation-aware self-attention [13] augments the vanilla self-attention [15] by introducing relation embeddings into the key and value entries. Assume the input to the self attention is a sequence of n embeddings $X = \sum_{i=1}^{n} x_i$, where $x_i \in R^{d_x}$, then it calculates its output z as (|| means concatenate operation):

$$\alpha_{ij}^{(h)} = soft\max\left(\frac{(x_i W_K^{(h)}\left(x_j W_K^{(h)} + r_{ij}^K\right)^T}{\sqrt{d_z / H}}\right),$$

$$z_i = \|_{h=1}^{H}\left[\sum_{j=1}^{n} \alpha_{ij}^{(h)}\left(x_j W_V^{(h)} + r_{ij}^V\right)\right], \tag{3}$$

where H is the number of heads, and $W_Q^{(h)}$, $W_K^{(h)}$, $W_V^{(h)}$ are learnable weights. The r_{ij}^K, r_{ij}^V are two different relation embeddings used to represent the relation r between the i-th and j-th token.

IV. Baseline

The baseline model is to use the original Transformer model, which treats text-to-SQL tasks as processing text-to-text tasks [15].

On this basis, this task consists of replacing the self-attention module in the encoder of the Transformer model with a relational attention module for text-to-SQL transformation. Compared with the baseline model, the RASAT model introduces the relational attention mechanism, which makes the model more accurate and efficient in understanding the semantic relationship between text and SQL. The relational attention module better captures the associations between entities in the text and thus generates more accurate SQL queries. As a result, the RASAT model has higher performance and accuracy in handling text-to-SQL conversion tasks, providing a more intelligent and efficient solution for database querying and management.

V. Limitation

Due to the limitations of the author's computing resources it was impossible to utilize the large T5-3B model, and therefore this task could not be optimized. Model was trained on an A100 GPU (80G) for about 1 day. The model truncates the source sequence to 512, which can lead to loss of information. This can lead to information loss when the sample input is long. The model is only applicable to English because it has more analytical tools and resources than other languages.

Conclusion

Table illustrates results as well as the results on the baseline model. By integrating relational attention into a pre-trained Transformer Seq2Seq model using the Spider dataset, the RASAT model (based on T5-small) achieves significant improvements in the text-to-SQL transformation task compared to the baseline model (T5-small).

Specifically, the RASAT model improves 1.82 % on Exact Match (EM), from 47.2 % to 49.03 %, and 3.26 % on Execution Accuracy (EX), from 47.8 % to 51.06 %.

These results show that by replacing the self-attention module in the Transformer encoder with a relational attention module, the RASAT model is more accurate and efficient in understanding the semantic relationship between text and SQL. The relational attention module is able to better capture the associations between entities in text, which makes the generated SQL queries more accurate and provides a more intelligent and efficient solution for database querying and management.

It is worth mentioning that the baseline model has been trained for 3072 epochs, but due to the limitation of computational resources, my RASAT model has only been trained for 500 epochs on A100 GPUs, and the batch_size has been reduced to the original general one, which is far less than the number of rounds of training of the baseline model, but it can be clearly seen that the result has been better than that of the baseline model.

Result for T5 model and RASAT model on Spider dev set.
The performance of T5 baselines are from [17]

Approach	EM	EX
T5-small	47.2 %	47.8 %
RASAT (T5-small)	49.03 % (+1.82 %)	51.06 % (+3.26 %)

Список литературы

1. Kamath, A. A Survey on Semantic Parsing / A. Kamath, R. Das // arXiv. – 2018. – 1812.00978. – DOI 10.48550/arXiv.1812.00978

2. Xu, X. SQLnet: Generating Structured Queries from Natural Language without Reinforcement Learning / X. Xu, C. Liu, D. Song // arXiv. – 2017. – 1711.04436. – DOI 10.48550/arXiv.1711.04436

3. A Comprehensive Exploration on wikiSQL with Table-aware Word Contextualization / W. Hwang, J. Yim, S. Park, M. Seo // arXiv. – 2019. – 1902.01069. – DOI 10.48550/ arXiv.1902.01069

4. RYANSQL: Recursively Applying Sketch-based Slot Fillings for Complex Text-to-SQL in Cross-Domain Databases / D.H. Choi, M. Ch. Shin, E.G. Kim, D. R. Shin // Computational Linguistics. – 2021. – Vol. 47, № 2. – P. 309–332. – DOI 10.1162/coli_a_00403

5. Bogin, B. Global Reasoning over Database Structures for Text to-SQL Parsing / B. Bogin, M. Gardner, J. Berant // Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLPIJCNLP), 3–7 November 2019, Hong Kong, China. – Association for Computational Linguistics, 2019. – P. 3659–3664. – DOI 10.18653/v1/D19-1378

6. LGESQL: Line Graph Enhanced Text-to-SQL Model with Mixed Local and Non-Local Relations / R. Cao, L. Chen, Z. Chen, Y. Zhao, S. Zhu, K. Yu // Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 1–6 August 2021, Online. – Association for Computational Linguistics, 2021. – P. 2541–2555. – DOI 10.18653/v1/2021.acl-long.198

7. SADGA: Structure-Aware Dual Graph Aggregation Network for Text-to-SQL / R. Cai, J. Yuan, B.Xu, Z. Hao // arXiv. – 2021. – 2111.00653. – DOI 10.48550/arXiv.2111.00653

8. S2SQL: Injecting syntax to question-schema interaction graph encoder for text-to-SQL parsers / B. Hui, R. Geng, L. Wang, B. Qin, B. Li, J. Sun, Y. Li // arXiv. – 2022. – 2408.03256. – DOI 10.48550/arXiv.2408.03256

9. Relational Graph Attention Network for Aspect-based Sentiment Analysis / K. Wang, W. Shen, Y. Yang, X. Quan, R. Wang // Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 5–10 July 2020, Online. – Association for Computational Linguistics, 2020. – P. 3229–3238. – DOI 10.18653/v1/2020.acl-main.295

10. Compositional Generalization and Natural Language Variation: Can a Semantic Parsing Approach Handle Both? / P. Shaw, M.-W. Chang, P. Pasupat, K. Toutanova // Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 1–6 August 2021, Online – Association for Computational Linguistics, 2021. – P. 922–938. – DOI 10.18653/v1/2021.acl-long.75

11. Scholak, T. PICARD: Parsing Incrementally for Constrained Auto-Regressive Decoding from Language Models / T. Scholak, N. Schucher, D. Bahdanau // Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Online and Punta Cana, Dominican Republic, 7–11 November 2011. – Association for Computational Linguistics, 2021. – P. 9895–9901 – DOI 10.18653/v1/2021.emnlp-main.779

12. Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task / T. Yu, R. Zhang, K. Yang [et al.] // Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, Belgium, 31 October – 4 November 2018. – Association for Computational Linguistics, 2018. – P. 3911–3921. – DOI 10.18653/v1/D18-1425

13. Shaw, P. Self-Attention with Relative Position Representations / P. Shaw, J. Uszkoreit, A. Vaswani // Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), New Orleans, Louisiana, 1–6 June 2018. – Association for Computational Linguistics, 2018. – P. 464–468. – DOI 10.18653/v1/N18-2074.

14. Attention is All you Need / V. Ashish, N. M. Shazeer, N. Parmar [et al.] // Advances in Neural Information Processing Systems. – 2017. – Vol. 30. – 11 p.

15. UnifiedSKG: Unifying and Multi-Tasking Structured Knowledge Grounding with Text-to-Text Language Models / T. Xie, C. H. Wu, P. Shi [et al.] // arXiv. – 2022. – 2201.05966. – DOI 10.48550/arXiv.2201.05966

16. Yale Semantic Parsing and Text-to-SQL Challenge: сайт [Электронный ресурс]. – URL: https://yale-lily.github.io/spider (accessed 15.11.2024)

17. RASAT: Integrating Relational Structures into Pretrained Seq2Seq Model for Text-to-SQL / J. Qi, J. Tang, Z. He [et al.] // arXiv. – 2022. – 2205.06983. – DOI 10.48550/arXiv.2205.06983

18. Giordani, A. Corpora for Automatically Learning to Map Natural Language Questions into SQL Queries. / A. Giordani, A. Moschitti // Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, 17–23 May 2010, Valletta, Malta. – European Language Resources Association (ELRA), 2010. – P. 2336–2339.

References

1. Kamath A., Das R. A Survey on Semantic Parsing. *arXiv*, 2018, 1812.00978. – DOI 10.48550/arXiv.1812.00978

2. Xu X., Liu C., Song D. SQLnet: Generating Structured Queries from Natural Language without Reinforcement Learning. arXiv, 2017, 1711.04436. – DOI 10.48550/arXiv.1711.04436

3. Hwang W., Yim J., Park S., Seo M. A Comprehensive Exploration on wikiSQL with Table-aware Word Contextualization. *arXiv*, 2019, 1902.01069. – DOI 10.48550/arXiv.1902.01069

4. Choi D.H., Shin M.Ch., Kim E.G., Shin D.R. RYANSQL: Recursively Applying Sketch-based Slot Fillings for Complex Text-to-SQL in Cross-Domain Databases. *Computational Linguistics*, 2021, vol. 47, no. 2, pp. 309–332. – DOI 10.1162/coli_a_00403

5. Bogin B., Gardner M., Berant J. Global Reasoning over Database Structures for Text to-SQL Parsing. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLPIJCNLP), 3-7 November 2019, Hong Kong, China, Association for Computational Linguistics, 2019, pp. 3659–3664. – DOI 10.18653/v1/D19-1378

6. Cao R., Chen L., Chen Z., Zhao Y., Zhu S., Yu K. LGESQL: Line Graph Enhanced Text-to-SQL Model with Mixed Local and Non-Local Relations. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 1-6 August 2021, Online, Association for Computational Linguistics, 2021, pp. 2541–2555. – DOI 10.18653/v1/2021.acl-long.198

7. Cai R., Yuan J., Xu B., Hao Z. SADGA: Structure-Aware Dual Graph Aggregation Network for Text-to-SQL. *arXiv*, 2021, 2111.00653. – DOI 10.48550/arXiv.2111.00653

8. Hui B., Geng R., Wang L., Qin B., Li B., Sun J., Li Y. S2SQL: Injecting syntax to question-schema interaction graph encoder for text-to-SQL parsers. *arXiv*, 2022, 2408.03256. – DOI 10.48550/arXiv.2408.03256

9. Wang K., Shen W., Yang Y., Quan X., Wang R. Relational Graph Attention Network for Aspect-based Sentiment Analysis. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 5-10 July 2020, Online. – Association for Computational Linguistics, 2020, pp. 3229–3238. – DOI 10.18653/v1/2020.acl-main.295

10. Shaw P., Chang M.-W., Pasupat P., Toutanova K. Compositional Generalization and Natural Language Variation: Can a Semantic Parsing Approach Handle Both? In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 1–6 August 2021, Online, Association for Computational Linguistics, 2021, pp. 922–938. – DOI 10.18653/v1/2021.acl-long.75

11. Scholak T., Schucher N., Bahdanau D. PICARD: Parsing Incrementally for Constrained Auto-Regressive Decoding from Language Models. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Online and Punta Cana, Dominican Republic, 7–11 November 2011, Association for Computational Linguistics, 2021, pp. 9895–9901 – DOI 10.18653/v1/2021.emnlp-main.779

12. T. Yu, R. Zhang, K. Yang et al. Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, Belgium, 31 October – 4 November, 2018, Association for Computational Linguistics, 2018, pp. 3911–3921. – DOI 10.18653/v1/D18-1425

13. Shaw P., Uszkoreit J., Vaswani A. Self-Attention with Relative Position Representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), New Orleans, Louisiana, 1–6 June 2018, Association for Computational Linguistics, 2018, pp. 464–468. – DOI 10.18653/v1/N18-2074.

14. Ashish V., Shazeer N.M., Parmar N. et al. Attention is All you Need. *Advances in Neural Information Processing Systems*, 2017, vol. 30, 11 p.

15. Xie T., Wu C.H., Shi P. et al. UnifiedSKG: Unifying and Multi-Tasking Structured Knowledge Grounding with Text-to-Text Language Models. *arXiv*, 2022, 2201.05966. – DOI 10.48550/arXiv.2201.05966

16. Yale Semantic Parsing and Text-to-SQL Challenge, available at: https://yale-lily.github.io/spider (accessed 15.11.2024)

17. Qi J., Tang J., He Z. et al. RASAT: Integrating Relational Structures into Pretrained Seq2Seq Model for Text-to-SQL. *arXiv*, 2022, 2205.06983. – DOI 10.48550/arXiv.2205.06983

18. Giordani A., Moschitti A. Corpora for Automatically Learning to Map Natural Language Questions into SQL Queries. In Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, 17-23 May 2010, Valletta, Malta, European Language Resources Association (ELRA), 2010, pp. 2336–2339.