SADGA: Structure-Aware Dual Graph Aggregation Network for Text-to-SQL

Anonymous Author(s)
Affiliation
Address
email

Abstract

The Text-to-SQL task, aiming to translate the natural language of the questions into SQL queries, has drawn much attention recently. One of the most challenging problems of Text-to-SQL is how to generalize the trained model to the unseen databases schema, also known as the cross-domain Text-to-SQL task. The key lies in the generalizability of (i) the encoding method to model the question and the database schema and (ii) the question-schema linking method to learning the mapping between the words in the question and the tables/columns in the database schema. Focusing on the above two key issues, we propose a *Structure-Aware Dual Graph Aggregation Network* (SADGA) for the cross-domain Text-to-SQL task. In SADGA, we adopt the graph structure to provide a unified encoding model for both the natural language question and database schema. Based on the proposed unified modeling, we further devise a structure-aware aggregation method to learn the mapping between the question-graph and schema-graph. The structure-aware aggregation method is featured with *Global Graph Linking*, *Local Graph Linking* and *Dual-Graph Aggregation Mechanism*. We not only study the performance of our proposal empirically but also achieve 3rd place on the challenging Text-to-SQL benchmark Spider.

1 Introduction

Structured Query Language (SQL) has become the standard database query language for a long time, but the difficulty of writing still hinders the non-professional user from using SQL. With the development of semantic parsing and so on technologies, Text-to-SQL, aiming to automatically generate SQL from the natural language question, has achieved a great process and is the focus of NLP [4, 11, 28, 24].

The existing Text-to-SQL approaches are mainly developed for particular domains, that is, both training and inference phases are under the same database schema. However, it is hard for database developers to build their own Text-to-SQL model from scratch because of the high annotation cost for each specific database. Therefore, cross-domain Text-to-SQL, aiming to generalize the trained model to unseen data schema, is proposed as a more promising solution [9, 3, 2, 23, 5, 19, 15]. The core issue of these approaches lies in building the linking between the natural language question and database schema, well-known as question-schema linking problem [9, 23, 15].

There are two categories of efforts to solve the aforementioned question-schema linking problem – matching-based methods and learning-based methods. IRNet[9] is a typical matching-based method, which uses a simple string matching approach to link the question words and tables/columns. Wang et al. [23] is a typical learning-based method, which applies a relation-aware transformer to global learn the linking over the question and schema with pre-defined relations. However, both of the above two categories of methods still suffer from the problem of insufficient generalization ability. There

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

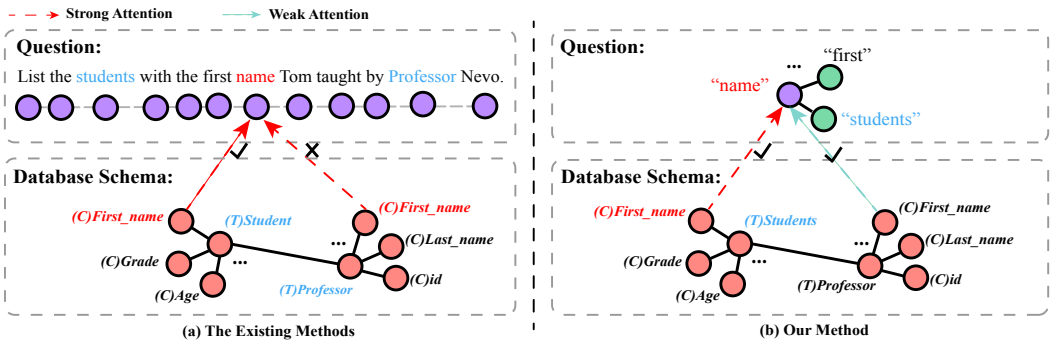

Figure 1: A toy example on the case of "List the students with the first name Tom taught by Professor Nevo".

are two main reasons for the above problem: (1) the structure gap between the encoding process of the questions and database schema: take Figure 1 for example, most of the existing methods learn the representation of the problem by sequential encoders or Transformers, while the representation of the database schema is based on the graph encoder. Such structure gap leads to the trained model can not be efficiently adapted to the unseen schema. (2) the latent association between the question words and the tables/columns of the database schema. Recall the example in Figure 1, existing works highly rely on the pre-defined relationship or self-supervised learning on question-schema linking, causing the latent association between problem words and database item to be undetectable. Such undetected latent associations also lead to the low generalization ability of the trained model.

Aiming to alleviate the structure gap and the undetected latent association, we propose a *Structure-Aware Dual Graph Aggregation Network* (SADGA) for the Cross-Domain Text-to-SQL task to fully take advantage of the structural information of both the questions and schemas. Different from recent studies using a multi-structured encoder, SADGA adapts to a unified graph neural network encoder to model both natural language problems and database schema. On the question-schema linking cross question-graph and schema-graph, SADGA is featured with *Global Graph Linking*, *Local Graph Linking* and *Dual-Graph Aggregation Mechanism*. In the *Global Graph Linking* phase, the query nodes on Question-graph or Schema-graph calculate the attention with the key node of another graph. In the *Local Graph Linking* phase, the query nodes will calculate the attention with neighbor nodes of the key node cross dual graphs. In the *Dual-Graph Aggregation mechanism*, the above two-phase linking processes are aggregated in a gated-based mechanism to obtain a unified structured representation of question-graph and schema-graph.

The contributions are summarized as follows:

- We propose a unified dual-graph neural network to interactively encode and aggregate structure information of question-graph and schema-graph.
- The structure-aware dual graph aggregation is featured with *Global Graph Linking*, *Local Graph Linking* and *Dual-Graph Aggregation Mechanism*.
- We conduct extensive experiments to study the effectiveness of SADGA. Especially, SADGA outperforms the baseline methods and achieves 3rd place on the challenging Text-to-SQL benchmark Spider[1] [29]. Our source code will be released if our work is accepted.

2 Model Overview

We provide the overview of our proposed method in Figure 2. As shown in the figure, the proposed method follows the typical encoder-decoder framework. There are two components of the encoder, the Structure-Aware Dual Graph Aggregation Network (SADGA) and the Relation-Aware Transformer (RAT) [23].

SADGA consists of the dual-graph construction, dual-graph encoding and structure-aware aggregation. In the workflow of the proposed SADGA, we first construct the question-graph based on the contextual

[1] `https://yale-lily.github.io/spider`

structure and dependency structure of the question and construct the schema-graph based on the natural structure of the database schema. Second, Gated Graph Neural Network is employed to encode the question-graph and schema-graph. Third, the structure-aware aggregation method learns the cross dual-graph mapping through two-stages linking. Finally, the information is aggregated in a gated-based mechanism to obtain a unified representation.

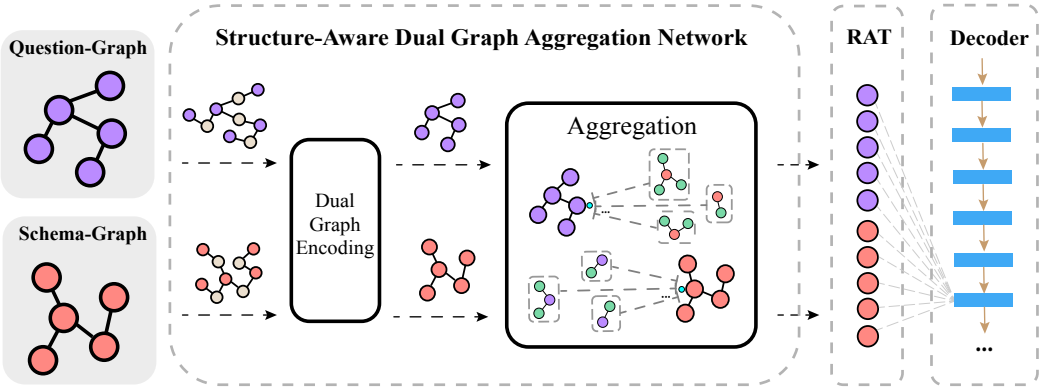

Figure 2: The overview of the proposed method.

RAT tries to further unify the representations learned by our SADGA by encoding the question words and tables/columns with the help of predefined relations. RAT is an extension of Transformer [22], which introduces prior knowledge relations to the self-attention mechanism. Different from the work Wang et al. [23] with more than 50 predefined relations, our work only uses 14 predefined relations, as shown in Table 1. The less predefined relations also benefit the generalization ability of our methods.

In the decoder, we follow the tree-structured architecture of Yin and Neubig [26], which transforms the SQL to an abstract syntax tree in depth-first traversal order. Then, we apply an LSTM to output a sequence of actions that generates the corresponding SQL syntax abstraction tree. These actions are either schema-independent (the grammar rule) or schema-specific (tables/columns). The reader may refer Yin and Neubig [26] for the details.

3 Structure-Aware Dual Graph Aggregation Network

In this section, we will delve into the Structure-Aware Dual Graph Aggregation Network (SADGA). The structure-aware aggregation method consists of *Global Graph Linking*, *Local Graph Linking* and *Dual-Graph Aggregation Mechanism*. These three steps introduce global and local structure information on question-schema linking. The details of each component are as follows.

Table 1: The structural relations for Dual Graph Construction and Aggregation.

	Node A	Node B	Relation
Question-Graph Construction	Word	Word	1-order Word Distance 2-order Word Distance Dependency Parsing
Schema-Graph Construction	Column	Column	Table Match Foreign-Primary Key
	Column	Table	Foreign Key Primary Key Table-Column Match
	Table	Table	Foreign Key
Cross-Graph	Word	Table	Exact Match Partial Match
	Word	Column	Exact Match Partial Match Value Match

3.1 Dual Graph Construction

In SADGA, we adopt a unified dual graph structure to model the structure of the question, the schema and the predefined linking between the words and tables/columns. The details of the generation of question-graph, schema-graph and predefined cross-graph linkings are as follows.

Question-Graph A question-graph can be presented by $\mathcal{G}_Q = (Q, R_Q)$, where the node set Q presents the words in the question and the set R_Q presents the dependencies among the words. There are three different types of dependency. They are the 1-order word distance dependency, the 2-order

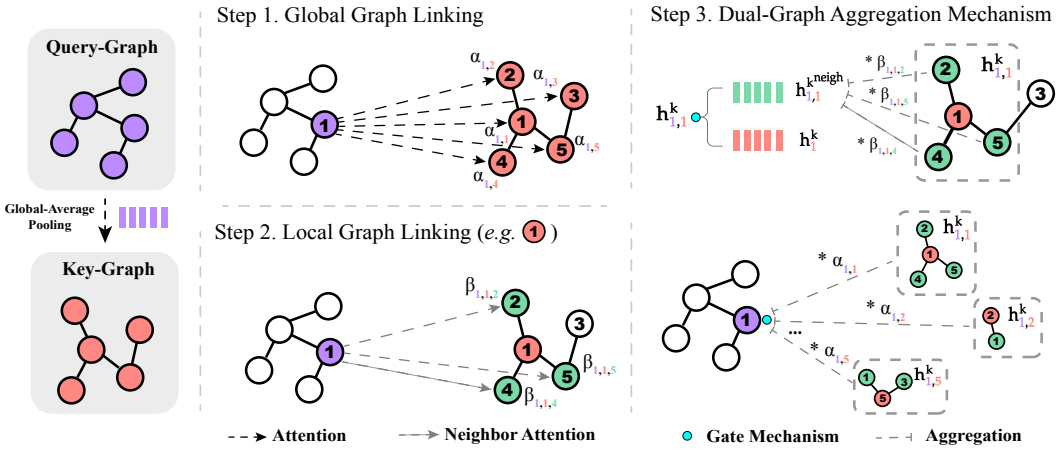

Figure 3: The aggregation procedure. We show the case when the 1st node in the question-graph acts as the query node. The query node attends not only the key node but also the neighbor nodes of the key node.

word distance dependency and the parsing based dependency. For the parsing based dependency, we extract dependency parsing link via Stanford CoreNLP toolkit [16].

Schema-Graph Similarly, a schema-graph can be presented by $\mathcal{G}_S = (S, R_S)$, where the node set S presents the tables/columns in the database schema and the edge set R_S presents the structural relations among the tables/columns in the schema. We use five typical relations in the database. They are the table match and foreign primary key for the column-column pairs, and the foreign primary key, primary key and table-column match for the column-table pairs.

Cross-Graph We also introduce cross-graph relations to capture the connection between question-graph and schema-graph. There are two kinds of rules to generate the relations. We use both the exact match rule and partial match rule about the word-table node pairs to build the relations. For the word-column node pairs, we use the exact match, partial match and value match.

All the rules used in the construction of the dual-graph are summarized in Table 1.

3.2 Dual-Graph Encoding

We employ Gated Graph Neural Network(GGNN)[14] to encode the graph representation of dual-graph by performing message propagation among their self-structure before building the two-stage graph linking cross dual-graph. Inspired by Beck et al. [1], instead of representing multiple relations in the edge, we represent the predefined relations of Table 1 in the node. In addition, we define three basic edge types for GGNN updating, i.e., bidirectional and self-loop.

3.3 Structure-Aware Aggregation

Following with dual-graph encoding, we devise a structure-aware aggregation method on question-schema linking between question graph $\mathcal{G}_Q$ and schema graph $\mathcal{G}_S$. The aggregation process is formulated as

$$\mathcal{G}'_Q = \text{GraphAggr}(\mathcal{G}_Q, \mathcal{G}_S), \quad \mathcal{G}'_S = \text{GraphAggr}(\mathcal{G}_S, \mathcal{G}_Q), \tag{1}$$

As shown in Eq. 1, the structure-aware aggregation method is applied to respectively to aggregate information from schema-graph $\mathcal{G}_S$ and question-graph $\mathcal{G}_Q$ to another graph. Thus, we illustrate the detailed approach in the manner of the query-graph $\mathcal{G}_q$ and key-graph $\mathcal{G}_k$, i.e.,

$$\mathcal{G}'_q = \text{GraphAggr}(\mathcal{G}_q, \mathcal{G}_k), \tag{2}$$

Let $\mathcal{G}_q = \{\boldsymbol{h}_i^q\}_{i=1}^m$ be the query graph $\mathcal{G}_q$ consisting of a set of node embedding h_i^q and $\mathcal{G}_k = \{\boldsymbol{h}_j^k\}_{j=1}^n$ be the key graph $\mathcal{G}_k$ consisting of a set of node embedding h_i^k, which both learned by dual-graph

encoding. Figure 3 shows the whole procedure of structure-aware aggregation method. First, we use global-average pooling on the node embedding h_i^q of query-graph $\mathcal{G}_q$ to get the global query-graph embedding q. Then, in order to capture globally relevant information and receives the query-aware context representation, the key node embedding h_j^k is updated as

$$\boldsymbol{q} = \frac{1}{m} \sum\nolimits_{i=1}^{m} \boldsymbol{h}_i^q, \tag{3}$$

$$e_j = \theta \left(\boldsymbol{q}^T \boldsymbol{W}_g \boldsymbol{h}_j^k \right), \tag{4}$$

$$\boldsymbol{h}_j^k = (1 - e_j) \boldsymbol{W}_{qg} \boldsymbol{q} + e_j \boldsymbol{W}_{kg} \boldsymbol{h}_j^k, \tag{5}$$

where $\boldsymbol{W}_g$, $\boldsymbol{W}_{qg}$, $\boldsymbol{W}_{kg}$ are trainable parameters and θ is a sigmoid function. e_j represents the relevance score between the j-th key node and the global query-graph. The above aggregation process is inspired by Zhang et al. [33], proposed structure-aware aggregation method to further introduce global and local structural information through three primary phases, including *Global Graph Linking*, *Local Graph Linking* and *Dual-Graph Aggregation Mechanism*.

Global Graph Linking *Global Graph Linking* is to learn the mapping relationship between each query node and the global structure of the key graph. Inspired by the relation-aware attention [23], we calculate the global attention score $\alpha_{i,j}$ between the query node embedding h_i^q and the key node embedding h_j^k as follows:

$$s_{i,j} = \sigma \left(\boldsymbol{h}_i^q \boldsymbol{W}_q \left(\boldsymbol{h}_j^k + \boldsymbol{R}_{ij}^E \right)^T \right), \alpha_{i,j} = \mathrm{softmax}_j \left\{ s_{i,j} \right\}, \tag{6}$$

where σ is nonlinear activation function and $\boldsymbol{R}_{ij}^E$ is a learned feature to represent the predefined relation between i-th query node and j-th key node.

Local Graph Linking *Local Graph Linking* is designed to detect latent associations by introducing local structure information on dual-graph linking. In this phase, the query nodes will calculate the attention with nighbor nodes of the key node cross dual graphs. Specifically, we calculate the local attention score $\beta_{i,j,t}$ between i-th query node and t-th neighbor of j-th key node, formulated as

$$o_{i,j,t} = \sigma \left(\boldsymbol{h}_i^q \boldsymbol{W}_{nq} \left(\boldsymbol{h}_t^k + \boldsymbol{R}_{it}^E \right)^T \right), \beta_{i,j,t} = \mathrm{softmax}_t \left\{ o_{i,j,t} \right\} (t \in \mathcal{N}_j), \tag{7}$$

where $\mathcal{N}_j$ represents the neighbors of the j-th key node.

Dual-Graph Aggregation Mechanism *Global Graph Linking* and *Local Graph Linking* phase process are aggregated with *Dual-Graph Aggregation Mechanism* to obtain the unified structured representation of the nodes in the key-graph. First, we aggregate the neighbor information with the local attention scores $\beta_{i,j,t}$, and then apply a gate function to extract essential features among the key node self and the neighbor information, i.e.,

$$\boldsymbol{h}_{i,j}^{k^{\mathrm{neigh}}} = \sum_{t=1}^{T} \beta_{i,j,t} \boldsymbol{h}_t^k, \quad \boldsymbol{h}_{i,j}^{k^{\mathrm{self}}} = \boldsymbol{h}_j^k, \tag{8}$$

$$\mathrm{gate}_{i,j} = \theta \left(\boldsymbol{W}_{ng} \left[\boldsymbol{h}_{i,j}^{k^{\mathrm{self}}} ; \boldsymbol{h}_{i,j}^{k^{\mathrm{neigh}}} \right] \right), \tag{9}$$

$$\boldsymbol{h}_{i,j}^k = (1 - \mathrm{gate}_{i,j}) * \boldsymbol{h}_{i,j}^{k^{\mathrm{self}}} + \mathrm{gate}_{i,j} * \boldsymbol{h}_{i,j}^{k^{\mathrm{neigh}}}, \tag{10}$$

where $\boldsymbol{h}_{i,j}^{k^{\mathrm{neigh}}}$ represents the neighbor context vector and $\boldsymbol{h}_{i,j}^k$ indicates the j-th key node neighbor-aware feature toward i-th query node. Finally, each query node aggregates the structure-aware information from all key nodes with the global attention score $\alpha_{i,j}$:

$$\boldsymbol{h}_i^{q^{\mathrm{new}}} = \sum\nolimits_{j=1}^{n} \alpha_{i,j} \left(\boldsymbol{h}_{i,j}^k + \boldsymbol{R}_{ij}^E \right), \tag{11}$$

$$\mathrm{gate}_i = \theta \left(\boldsymbol{W}_{\mathrm{gate}} \left[\boldsymbol{h}_i^q ; \boldsymbol{h}_i^{q^{\mathrm{new}}} \right] \right), \tag{12}$$

$$\boldsymbol{h}_i^{q^{\mathrm{aggr}}} = (1 - \mathrm{gate}_i) * \boldsymbol{h}_i^q + \mathrm{gate}_i * \boldsymbol{h}_i^{q^{\mathrm{new}}}, \tag{13}$$

where gate_i indicates how much information the query node should receive from the key-graph. Consequently, we obtain the final query node representation $\boldsymbol{h}_i^{q^{\mathrm{aggr}}}$ with the structure-aware information of the key-graph.

Table 2: Results on the Spider development and test set.

Approach	Dev	Test	Approach	Dev	Test
GNN [3]	40.7	39.4	RATSQL-HPFT + BERT-Large	69.3	64.4
Global-GNN [2]	52.7	47.4	YCSQL + BERT-Large	-	65.3
IRNet v2 [9]	55.4	48.5	DuoRAT + BERT-Large [20]	69.4	65.4
RAT-SQL [23]	62.7	57.2	RAT-SQL + BERT-Large [23]	69.7	65.6
SADGA	**64.7**	-	**SADGA + BERT-Large**	**71.6**	**66.7**
EditSQL + BERT-base [32]	57.6	53.4	ShadowGNN + RoBERTa [5]	72.3	66.1
GNN + Bertrand-DR [12]	57.9	54.6	RAT-SQL + STRUG [7]	72.6	68.4
IRNet v2 + BERT-base [9]	63.9	55.0	RAT-SQL + GraPPa [30]	**73.4**	69.6
RAT-SQL + BERT-base [23]	65.8	-	RAT-SQL + GAP [21]	71.8	69.7
SADGA + BERT-base	**69.0**	-	**SADGA + GAP**	73.1	**70.1**

4 Experiments

In this section, we conduct experiments on Spider dataset, the benchmark of cross-domain Text-to-SQL, [29] to evaluate the effectiveness of our model.

4.1 Experiment Setup

Dataset and Metrics The Spider has been so far the most challenging benchmark on cross-domain Text-to-SQL, which contains 9 traditional specific-domain datasets, such as ATIS [18, 6], GeoQuery [31], WikiSQL [34], IMDB [25] etc. It is split into training set(8659 examples), development set(1034 examples) , and test set(2147 examples), which are distributed across 146, 20 and 40 databases respectively. Since the fair competition, the Spider official has not released the test set for evaluation. Instead, participants must submit the model to obtain the test accuracy for the official non-released test set through the submission scripts provided officiallyYu et al. [29].

Embedding Initialization In our model, the embeddings of question words and tables/columns are initialized by the pre-training methods. Specifically, in terms of the pretrained vector, Glove [17] is a common choice for the embedding initialization. And in terms of the pretrained model, BERT [8], the general pretrained model, is the mainstream embedding-initialization method. In detail, BERT-base, BERT-Large are applied according to the model scale. In addition, other pretrained models are adopted as well. For example, the specific-domain pretrained models, i.e., GAP [21], GraPPa [30], STRUG [7] are applied for better taking advantage of prior Text-to-SQL knowledge.

Implementation We trained our models on one machine with a single NVIDIA GTX 3090 GPU. We follow the original hyperparameters of RAT-SQL [23] that uses batch size 20, initial learning rate 7×10^{-4}, max steps 40,000 and the Adam optimizer [13]. For BERT, the initial learning rate is adjusted to 2×10^{-4} and the training max step is increased to 90,000. We also apply a separate learning rate of 3×10^{-6} to fine-tune BERT. For GAP, we follow the original settings in Shi et al. [21] except that the training max step is set to 60,000. In addition, we stack three structure-aware dual graph aggregation layers followed by four RAT layers.

Table 3: The BERT-Large Results on Spider development set and test set by hardness levels defined by Yu et al. [29].

Model	Easy	Medium	Hard	Extra Hard	All
Dev:					
RAT-SQL	86.4	**73.6**	62.1	42.9	69.7
SADGA	**90.3**	72.4	**63.8**	**49.4**	**71.6**
Test:					
RAT-SQL	83.0	71.3	**58.3**	38.4	65.6
SADGA	**85.1**	**72.1**	57.0	**41.7**	**66.7**

4.2 Overall Performance

The exact match accuracy results are presented in Table 2. Noted that the test results of **SADGA** and **SADGA+BERT-base** are blank for the community submission limitation. And the results of the baselines are obtained from the official leader-board. As shown as the table, the proposed

SADGA model is competitive with the baselines in the identical sub-table. Specifically, regarding the development set, our raw SADGA and the one with BERT-base and BERT-Large enhancing (SADGA+BERT-base, SADGGA+BERT-Large), outperform their corresponding baselines. And with the GAP enhancing, our model(SADGA+GAP) is competitive with its best baseline, i.e., RAT-SQL+GraPPa, as well. While regarding the test set, our models, only available for the BERT-Large one and the GAP one, also surpass their competitors. Util our submission, our best model(SADGA+GAP) has achieved the 3rd on the overall leader-board.

To better demonstrate the effectiveness, our SADGA is evaluated on the development set and test set, according to the parsing difficulty level defined by Yu et al. [29], compared with RAT-SQL. As shown as Table 3. our SADGA outperforms the baseline on the Extra-Hard level by 6.5% and 3.3% on dev set and test set respectively, which implies that our model can handle more complicated SQL parsing.

4.3 Ablation Studies

Table 4: Accuracy of the ablation studies on Spider development set by hardness levels.

Model	Easy	Medium	Hard	Extra Hard	All
SADGA	**82.3**	**67.3**	**54.0**	**42.8**	**64.7**
w/o *Local Graph Linking*	83.5(+1.2)	64.8(-2.5)	53.4(-0.6)	38.6(-4.2)	63.2(-1.5)
w/o Aggregation	83.5(+1.2)	62.1(-5.2)	55.2(+1.2)	42.2(-0.6)	62.9(-1.8)
Q-S Linking with the GGNN	82.3(-0)	63.7(-3.6)	51.1(-2.9)	45.2(+2.4)	63.1(-1.6)
w/o $\mathrm{GraphAggr}(\mathcal{G}_S, \mathcal{G}_Q)$	83.1(+0.8)	64.1(-3.2)	52.3(-1.7)	40.4(-2.4)	62.9(-1.8)
w/o $\mathrm{GraphAggr}(\mathcal{G}_Q, \mathcal{G}_S)$	79.0(-3.3)	63.7(-3.6)	50.0(-4.0)	41.6(-1.2)	61.5(-3.2)
w/o Relation Node	79.4(-2.9)	63.5(-3.8)	54.6(+0.6)	40.4(-2.4)	62.1(-2.6)
w/o Aggregation Gate (Equation 9)	81.9(-0.4)	60.1(-7.2)	54.6(+0.6)	40.4(-2.4)	61.2(-3.5)
w/o Global Pooling	82.7(+0.4)	64.3(-3.0)	54.0(-0)	41.6(-1.2)	63.5(-1.2)
w/o Relation in Aggregation	79.4(-2.9)	64.3(-3.0)	54.6(+0.6)	41.6(-1.2)	62.7(-2.0)
SADGA + BERT-base	**85.9**	**71.7**	**58.0**	**47.6**	**69.0**
w/o *Local Graph Linking*	85.5(-0.4)	69.5(-2.2)	54.0(-4.0)	42.8(-4.8)	66.4(-2.6)
w/o Aggregation	85.9(-0)	68.8(-2.9)	57.5(-0.5)	41.0(-6.6)	66.5(-2.5)

To validate the effectiveness of each components of SADGA, the ablation study is conducted on different parsing difficulty levels. And the model variants are as followed: **w/o Local Graph Linking** Discard the *Local Graph Linking* phase (i.e., Equation 7~10), which means $\boldsymbol{h}_{i,j}^k$ in Equation 11 is replaced by $\boldsymbol{h}_j^k$.

w/o Aggregation Remove the whole aggregation module to examine the effectiveness of our designed graph aggregation method.

In addition, other fine-grain ablation experiments are conducted on our raw SADGA. The ablation experimental results are presented in Table 4. According to the results on **All** level, our models, no matter the raw SADGA or the one with BERT-base enhancing, decrease by about 2.6% and 2.5% while discarding *Local Graph Linking* phase and the entire aggregation method, which indicates the positive contribution to SADGA. Especially on **Extra-Hard** level, discarding the *Local Graph Linking* and the aggregation respectively leads to a large decrease of accuracy, which indicates both components help SADGA parsing more complex SQL. Interestingly, on **Easy** level, the results indicate that both components have a negative influence on our raw model. This phenomenon is due to the fact that the **Easy level** samples do not require capturing the local structure of our dual graph while constructing the question-schema linking. However, the structure-aware ability is necessary for the complicated SQL on the **Extra-Hard** level.

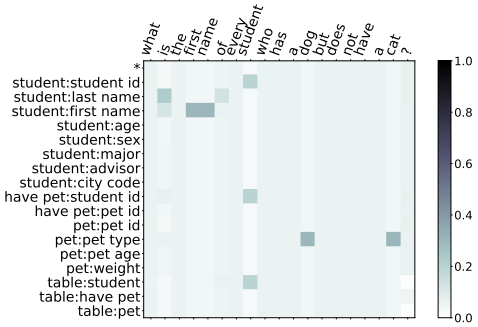

Figure 4: Alignment between the question words and tables/columns on the *Global Graph Linking* phase.

4.4 Case Study

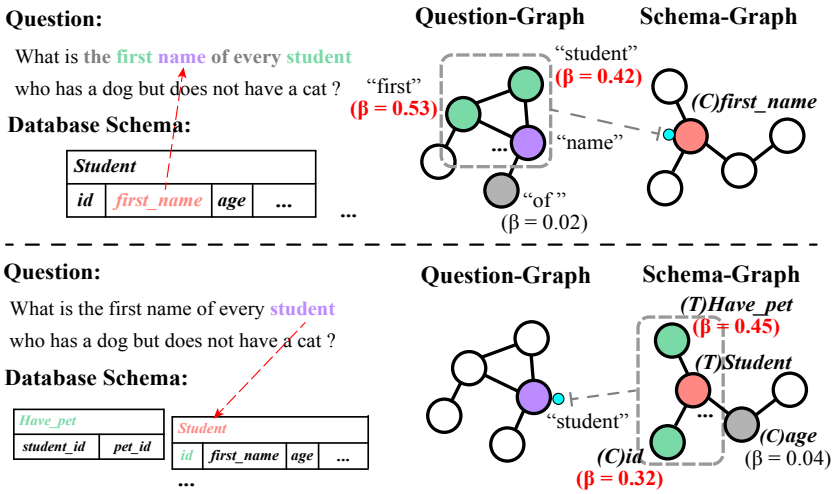

Figure 5: Analysis on the *Local Graph Linking* phase.

To further understand our method, in this section, we will conduct a detailed analysis of the case in which the question is "*What is the first name of every student who has a dog but does not have a cat?*".

Global Graph Linking Analysis We show the alignment figure between the question words and tables/columns on the *Global Graph Linking* phase when the question-graph acts as the query graph. As shown in Figure 4, we can obtain the interpretable result. For example, the question word "$student$" has a strong activation with the tables/columns related to the student, which helps better build the cross graph linking between the question and schema. Furthermore, we can observe that the column pet_type is successfully inferred by the word "dog" or "cat".

Local Graph Linking Analysis On the *Local Graph Linking* phase, we compute the attention between the query node and the neighbors of the key node, which allows the question words (tables/columns) to attend to the specific structure of the schema-graph (question-graph). In Figure 5, two examples about the neighbor attention on the *Local Graph Linking* phase are presented. As shown in the upper part of the Figure 5, the column $first_name$ of the table $Student$ attends to the neighbors of the word "$name$" in the question, where the word "$first$" and the word "$student$" obtain a high attention score, indicating that $first_name$ attends to the specific structure inside the dashed box.

Some tables/columns are difficult to be identified via matching-based alignment due to the fact that they do not attend explicitly in the question but are the critical entities, e.g., the table $have_pet$. Interestingly, as shown on the right side of Figure 5 shows, the table $Have_pet$ acquires a high attention weight when the question word "$student$" attends to the table $student$ and its neighbors. With the help of SADGA, the latent association between the table $Have_pet$ and "$student$" can be detected, which corresponds exactly to the semantics of the question.

5 Related Work

Cross-Domain Text-to-SQL Recent architectures proposed for the cross-domain Text-to-SQL show increasing complexity in both the encoder and the decoder. IRNet [9] encodes the question and schema separately via the LSTM with the string matching strategy and uses an AST-based decoder to decode an abstracted intermediate representation (IR). RAT-SQL [23] proposes a unified encoding mechanism to improve the joint representation of question and schema. Rubin and Berant [19] presents the first semi-autoregressive bottom-up semantic parser that enjoys logarithmic theoretical run-time and shows it is competitive with the commonly used autoregressive top-down parser.

Graph neural networks (GNNs) have been widely applied in various NLP tasks, and the cross-domain Text-to-SQL is no exception. Bogin et al. [3] proposes to encode the schema as a graph and utilizes it to guide decoding. Global-GNN [2] applies a GNN to softly select a subset of tables/columns for the output query, conditioned on the question. ShadowGNN [5] presents a graph project neural network to abstract the representation of the question and schema with a simple attention way.

Differently, SADGA not only adapts a unified dual-graph framework for both the question and schema but also devises a structure-aware graph aggregation mechanism to sufficiently utilize the global and local structure information of the dual-graph on the question-schema linking.

Graph Aggregation The global-local graph aggregation network [33] is proposed recently to aggregate the information between graphs. Nevertheless, it does not consider the structural information signifies that nodes in the graph are a series of entities without the graph structure. Differently, our approach considers the structural information that contributes to implementing a better structure-aware graph aggregation algorithm and introduces prior knowledge relations among nodes.

Pre-training Models Inspired by the success of pretrained language models, some recent work has tried to apply similar pretraining objectives to text-table data. More recently, TAPAS [10] and TaBERT [27] leverage the semi-structured table data to enhance the representation ability of language models. For Text-to-SQL, the pretrained model GraPPa [30] and STRUG [7] apply synchronous context-free grammar to generate synthetic data and utilized existing high-quality data-to-text dataset for pre-training, respectively. Moreover, [21] explore the direction of utilizing the generators to enhance the joint utterances and structured schema encoding ability of the pretrained models.

6 Conclusions

In this paper, we propose a *Structure-Aware Dual Graph Aggregation Network* (SADGA) network for cross-domain Text-to-SQL task. SADGA not only introduces a unified dual-graph encoding for both natural language question and database schema, but also devises a structure-aware aggregation mechanism of SADGA to takes full advantage of the global and local structure information of the dual-graph in the question-schema linking. Experimental results show that our proposal achieves the 3rd on the challenging Text-to-SQL benchmark Spider. This study shows that both the dual-graph encoding and structure-aware dual graph aggregation method are able to improve the generalization ability of the cross-domain alignment tasks. As future work, we will extend SADGA to other heterogeneous graph tasks.

References

[1] Daniel Beck, Gholamreza Haffari, and Trevor Cohn. Graph-to-sequence learning using gated graph neural networks. *arXiv preprint arXiv:1806.09835*, 2018.

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

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

[4] Ruichu Cai, Boyan Xu, Xiaoyan Yang, Zhenjie Zhang, Zijian Li, and Zhihao Liang. An encoder-decoder framework translating natural language to database queries. *arXiv preprint arXiv:1711.06061*, 2017.

[5] Zhi Chen, Lu Chen, Yanbin Zhao, Ruisheng Cao, Zihan Xu, Su Zhu, and Kai Yu. Shadowgnn: Graph projection neural network for text-to-sql parser. *arXiv preprint arXiv:2104.04689*, 2021.

[6] Deborah A Dahl, Madeleine Bates, Michael K Brown, William M Fisher, Kate Hunicke-Smith, David S Pallett, Christine Pao, Alexander Rudnicky, and Elizabeth Shriberg. Expanding the scope of the atis task: The atis-3 corpus. In *HUMAN LANGUAGE TECHNOLOGY: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*, 1994.

[7] Xiang Deng, Ahmed Hassan Awadallah, Christopher Meek, Oleksandr Polozov, Huan Sun, and Matthew Richardson. Structure-grounded pretraining for text-to-sql. *arXiv preprint arXiv:2010.12773*, 2021.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[9] Jiaqi Guo, 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.

[10] Jonathan Herzig, Paweł Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Martin Eisenschlos. Tapas: Weakly supervised table parsing via pre-training. *arXiv preprint arXiv:2004.02349*, 2020.

[11] Wonseok Hwang, Jinyeong Yim, Seunghyun Park, and Minjoon Seo. A comprehensive exploration on wikisql with table-aware word contextualization. *arXiv preprint arXiv:1902.01069*, 2019.

[12] Amol Kelkar, Rohan Relan, Vaishali Bhardwaj, Saurabh Vaichal, Chandra Khatri, and Peter Relan. Bertrand-dr: Improving text-to-sql using a discriminative re-ranker. *arXiv preprint arXiv:2002.00557*, 2020.

[13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[14] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015.

[15] Xi Victoria Lin, Richard Socher, and Caiming Xiong. Bridging textual and tabular data for cross-domain text-to-sql semantic parsing. *arXiv preprint arXiv:2012.12627*, 2020.

[16] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014.

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

[18] Patti Price. Evaluation of spoken language systems: The atis domain. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*, 1990.

[19] Ohad Rubin and Jonathan Berant. Smbop: Semi-autoregressive bottom-up semantic parsing. *arXiv preprint arXiv:2010.12412*, 2021.

[20] Torsten Scholak, Raymond Li, Dzmitry Bahdanau, Harm de Vries, and Chris Pal. Duorat: Towards simpler text-to-sql models. *arXiv preprint arXiv:2010.11119*, 2021.

[21] Peng Shi, Patrick Ng, Zhiguo Wang, Henghui Zhu, Alexander Hanbo Li, Jun Wang, Cicero Nogueira dos Santos, and Bing Xiang. Learning contextual representations for semantic parsing with generation-augmented pre-training. *arXiv preprint arXiv:2012.10309*, 2021.

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[23] Bailin Wang, Richard Shin, Xiaodong Liu, Oleksandr Polozov, and Matthew Richardson. Rat-sql: Relation-aware schema encoding and linking for text-to-sql parsers. *arXiv preprint arXiv:1911.04942*, 2020.

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

[25] Navid Yaghmazadeh, Yuepeng Wang, Isil Dillig, and Thomas Dillig. Sqlizer: query synthesis from natural language. *Proceedings of the ACM on Programming Languages*, 1(OOPSLA): 1–26, 2017.

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

[27] Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. Tabert: Pretraining for joint understanding of textual and tabular data. *arXiv preprint arXiv:2005.08314*, 2020.

[28] Tao Yu, Zifan Li, Zilin Zhang, Rui Zhang, and Dragomir Radev. Typesql: Knowledge-based type-aware neural text-to-sql generation. *arXiv preprint arXiv:1804.09769*, 2018.

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

[30] Tao Yu, Chien-Sheng Wu, Xi Victoria Lin, Bailin Wang, Yi Chern Tan, Xinyi Yang, Dragomir Radev, Richard Socher, and Caiming Xiong. Grappa: Grammar-augmented pre-training for table semantic parsing. *arXiv preprint arXiv:2009.13845*, 2021.

[31] John M Zelle and Raymond J Mooney. Learning to parse database queries using inductive logic programming. In *Proceedings of the national conference on artificial intelligence*, pages 1050–1055, 1996.

[32] Rui Zhang, 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.

[33] Shengyu Zhang, Ziqi Tan, Zhou Zhao, Jin Yu, Kun Kuang, Tan Jiang, Jingren Zhou, Hongxia Yang, and Fei Wu. Comprehensive information integration modeling framework for video titling. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2744–2754, 2020.

[34] Victor Zhong, Caiming Xiong, and Richard Socher. Seq2sql: Generating structured queries from natural language using reinforcement learning. *CoRR*, abs/1709.00103, 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No] Our paper focus on a challenging language generation problem without any potential negative societal impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

   (a) If your work uses existing assets, did you cite the creators? [Yes]

   (b) Did you mention the license of the assets? [Yes]

   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]

   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] **Spider** uses the CC BY-SA 4.0 license, and we do not discuss it.

   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] **Spider** is collected in 200 public databases, which do not contain personally identifiable information or offensive content.

5. If you used crowdsourcing or conducted research with human subjects...

   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [No]

   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [No]

   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [No]