PAPER

Local-to-Global Structure-Aware Transformer for Question Answering over Structured Knowledge

Yingyao WANG[†a)], ***Nonmember***, **Han WANG**[††,†††b)], ***Student Member***, **Chaoqun DUAN**[†], ***and*** **Tiejun ZHAO**[†c)], ***Nonmembers***

SUMMARY Question-answering tasks over structured knowledge (i.e., tables and graphs) require the ability to encode structural information. Traditional pre-trained language models trained on linear-chain natural language cannot be directly applied to encode tables and graphs. The existing methods adopt the pre-trained models in such tasks by flattening structured knowledge into sequences. However, the serialization operation will lead to the loss of the structural information of knowledge. To better employ pre-trained transformers for structured knowledge representation, we propose a novel structure-aware transformer (SATrans) that injects the local-to-global structural information of the knowledge into the mask of the different self-attention layers. Specifically, in the lower self-attention layers, SATrans focus on the local structural information of each knowledge token to learn a more robust representation of it. In the upper self-attention layers, SATrans further injects the global information of the structured knowledge to integrate the information among knowledge tokens. In this way, the SATrans can effectively learn the semantic representation and structural information from the knowledge sequence and the attention mask, respectively. We evaluate SATrans on the table fact verification task and the knowledge base question-answering task. Furthermore, we explore two methods to combine symbolic and linguistic reasoning for these tasks to solve the problem that the pre-trained models lack symbolic reasoning ability. The experiment results reveal that the methods consistently outperform strong baselines on the two benchmarks.

key words: *knowledge representation, pretrained transformer, knowledge base question answering, table fact verification*

1. Introduction

With the development of deep learning and the progress of computational power, various pre-trained language models (PTLMs) are proposed [1]–[7] and widely applied to natural language processing (NLP) tasks. Benefiting from a massive pretraining dataset, PTLMs can learn more syntactic and semantic language features. However, as PTLMs are usually trained on a dataset that consists of sentences with a linear-chain structure, they cannot encode knowledge with a complex structure. Therefore, PTLMs cannot be directly applied to structured knowledge-based question-answering tasks which require understanding and reasoning over structured knowledge evidence, like tables and graphs.

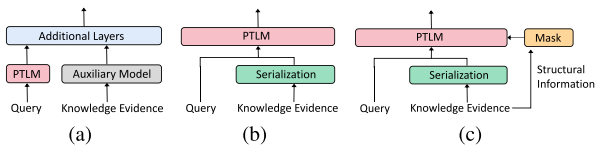


Fig. 1 Different methods to adopt PTLMs to structured knowledge-based question answering tasks.

Some studies have attempted to exploit the method to apply PTLMs to structured knowledge representation to mitigate this issue. The existing methods can be divided into three categories. As illustrated in Fig. 1(a), the first category leverages PTLMs and an auxiliary model to encode the natural language query and the structured knowledge, respectively, and introduces additional layers to model their relationship [8]. In these methods, encoding the query and knowledge is implemented between two adjacent steps, reducing the model efficiency because these operations can be compressed into one step. Moreover, these methods introduce more parameters that must be trained from scratch, increasing the difficulty of training the model.

Figure 1 (b) depicts the second method type, which leverages a unified PTLM to encode the query and its knowledge simultaneously. These methods [9] serialize knowledge with a complex structure into a sequence and adopt PTLMs to encode it like a sentence. However, the serialization operation destroys the structural information within the knowledge. For example, if a table is serialized into a sequence by connecting its row content, it is difficult for the PTLMs to recover the column alignments of different rows from the flattened word sequence. The same problem exists for the serialization of the knowledge graph. A common technique to flatten a knowledge graph is to connect its paths or triplets into a word sequence [10], causing the loss of the adjacency information within the graph.

Ideally, applying PTLMs to structured knowledge-based tasks should jointly encode the inputs of various structures with their structural information, such as the alignment information of table cells and the adjacency between graph nodes. As Fig. 1 (c) reveals, to overcome the drawback of the second category methods, the third category methods like [11] propose injecting the structural information from the knowledge into the mask of the self-attention layer. The

Manuscript received February 25, 2023.
Manuscript revised May 9, 2023.
Manuscript publicized June 27, 2023.

[†]The authors are with the Harbin Institute of Technology, Harbin, China, 150001.
[††]The author is with the Key Laboratory of Speech Acoustics and Content Understanding, Institute of Acoustics, Chinese Academy of Sciences, Beijing, China.
[†††]The author is with the University of Chinese Academy of Sciences, Beijing, China.
a) E-mail: yywang@hit-mtlab.net
b) E-mail: wanghan2018@mail.ioa.ac.cn
c) E-mail: tjzhao@hit.edu.cn

DOI: 10.1587/transinf.2023EDP7034

mask leverages the original structure of the knowledge to control the token representation fusion of the connected sequence. Through this mechanism, the method also succeeds in preserving structural information.

Applying the attention mask mechanism lets the PTLMs learn the structural information. However, existing methods usually use the fixed mask containing the global structural information during the whole knowledge representation process, even though the structure of knowledge is complex. Complex knowledge structures tend to be multi-grained. For example, tables contain cells, rows, and columns, while KG contains nodes, triples, and paths. In the case of simply using the fixed mask, the encoding process of each token is forced to consider the global structural information from the beginning. This out-of-focus structural information will introduce noise into the basic representation of the token. For example, when encoding a table cell, its row neighbors are valuable lexical features because cells in the same row usually describe the same fact. However, considering all alignment information of a cell will introduce the lexical information from column-granularity neighbors, but column neighbors usually have no semantic relevance. In fact, such global structural information contributes more to cross-granularity reasoning over the whole knowledge.

To address the above problem and better develop the PTLMs to represent structured knowledge, we propose a local-to-global structure-aware transformer (SATrans). It learns the structural information of knowledge by injecting local-to-global structural information into masks of different self-attention layers. Specifically, in the lower self-attention layers, SATrans focus on the local neighbor information of each knowledge token to learn a more robust representation of it. While in the upper self-attention layers, SATrans injects global structure information into the attention mask to integrate all token representations and perform cross-granularity reasoning over the knowledge.

We evaluate SATrans on two structured knowledge-based question-answering tasks, Table Fact Verification (TFV) and Knowledge Base Question Answering (KBQA), to verify its ability of structured knowledge representation, respectively, on tables and graphs. The TFV task aims to classify whether a factoid statement is entailed or refuted by the given evidence table. The KBQA task aims to determine the answer to a question from a given knowledge base with a graphical structure. Both tasks involve a natural language query and structured knowledge evidence that exactly matches the requirement of verifying the ability of this method to model structured information. We conduct experiments on the TabFact [9] dataset for the TFV task and on the WebQSP [12] dataset for KBQA. The experimental results show that SATrans outperforms strong baselines on these benchmarks.

In addition to encoding structural information, structured knowledge-based question-answering tasks usually require the ability to perform symbolic reasoning. Specifically, in the TFV task, some statements require numerical operations over the table cells, such as counting, comparing, and calculating. In the KBQA task, the model must determine the core inference chain, representing the path from the question-related node to the correct answer node. The PTLMs cannot perform such symbolic reasoning. Thus, we further explore two methods to convert symbolic reasoning into linguistic reasoning for the two tasks to enhance the question-answering performance. The main contributions of this work are as follows:

1. We devise a SATrans to use PTLMs better to represent structured knowledge, such as tables and graphs, by injecting local-to-global structural information into the attention masks of different attention layers.
2. To fill the gap that the SATrans lack symbolic reasoning ability, we explore two methods to combine symbolic reasoning and semantic matching for the TFV and KBQA tasks.
3. We conduct extensive experiments on two structured knowledge-based tasks, TFV and KBQA. The results reveal that this method outperforms strong baselines on these benchmarks, confirming the effectiveness of the proposed method.

This journal paper is an extended version of our conference paper [13] of EMNLP. In this paper, we extend the application of the proposed SATrans from the TFV task to the KBQA task and conduct experiments on the widely used dataset WebQSP. The improvement on the KBQA further proves that our method has an advantage in encoding structured information in advance. The new content is described in Sect. 2.3 and Sect. 3.3 respectively. In addition, we add experiments and discussions in Sect. 3.2 to analyze the effect of different components in the model.

2. Methodology

In this work, we adopt the representative pre-trained model BERT [1] as the backbone of SATrans. The inputs of TFV comprise a statement and relevant table, and those of KBQA include a query and subgraph. To use the BERT to encode tables and subgraphs, we serialize them into sequences. After obtaining the serialized structured evidence, the tables and subgraphs are concatenated with the statement and query, and input into the BERT. Moreover, we propose SATrans, which uses a local-to-global structure-aware mask to preserve and integrate the structural information into the BERT.

In this section, we first introduce the framework of SATrans and then introduce the knowledge serialization and mask construction methods for TFV and KBQA tasks.

2.1 Structure-Aware Transformer

Figure 2 presents the architecture of the SATrans. The model consists of a serialization operation, an attention mask generator, and N-layer transformers. Formally, given a query sentence $\boldsymbol{P} = \{p_1, p_2, \ldots, p_{l_p}\}$ and the structured knowledge E, SATrans first serialize E into a sequence

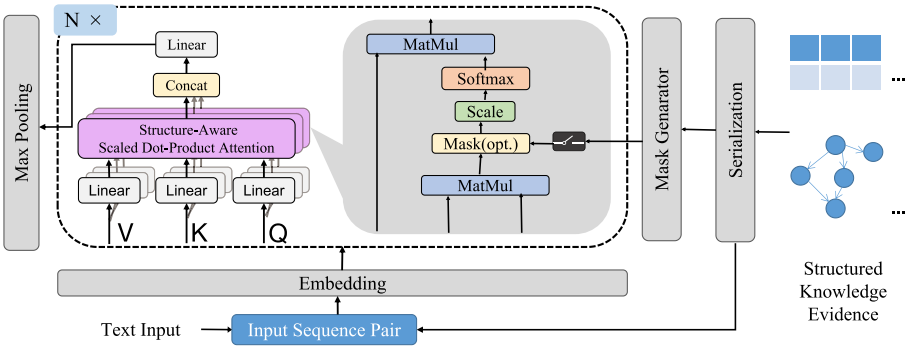


Fig. 2 Framework of the structured-aware transformer.

$\boldsymbol{Q} = \{q_1, q_2, \ldots, q_{l_q}\}$ Then, $\boldsymbol{P}$ and $\boldsymbol{Q}$ are concatenated as $\boldsymbol{S} = \{w_1, w_2, \ldots, w_L\}$, where $L = l_p + l_q$. The sequence $\boldsymbol{S}$ is input into the SATrans to learn the semantic representation. Specifically, $\boldsymbol{S}$ is first converted into vectors $\boldsymbol{H}^0 = \{\boldsymbol{h}_1^0, \boldsymbol{h}_2^0, \ldots, \boldsymbol{h}_L^0\}$ using an embedding layer. Afterward, $\boldsymbol{H}^0$ is input into the encoder of SATrans to learn the representation for each token.

For an input sequence containing the serialized knowledge, the local-to-global attention masks are constructed by the mask generator according to the knowledge structure. The mask is defined as $\boldsymbol{M} \in \mathbb{R}^{L \times L}$, each value $\boldsymbol{M}_{i,j}$ represents the adjacent relationship of tokens w_i and w_j in the knowledge structure. $w_i \sim w_j$ means that w_j is attended to when generating the representation of w_i, whereas $w_j \nsim w_i$ indicates the opposite. Each $\boldsymbol{M}_{i,j}$ is denoted as follows:

$$\boldsymbol{M}_{i,j} = \begin{cases} 0 & w_i \sim w_j \\ -\infty & w_i \nsim w_j \end{cases}. \tag{1}$$

As depicted in Fig. 2, after obtaining the mask $\boldsymbol{M}$, it is incorporated into the scaled dot-product attention. For example, in the n-th layer, given the output of the previous layer, $\boldsymbol{H}^{n-1} = \{\boldsymbol{h}_1^{n-1}, \boldsymbol{h}_2^{n-1}, \ldots, \boldsymbol{h}_L^{n-1}\}$, the mask is integrated into the m-th head as follows:

$$\boldsymbol{Q}_m, \boldsymbol{K}_m, \boldsymbol{V}_m = \boldsymbol{H}^{n-1}\boldsymbol{W}_m^Q, \boldsymbol{H}^{n-1}\boldsymbol{W}_m^K, \boldsymbol{H}^{n-1}\boldsymbol{W}_m^V, \tag{2}$$

$$\boldsymbol{A} = \text{softmax}(\frac{\boldsymbol{Q}_m {\boldsymbol{K}_m}^T + \boldsymbol{M}}{\sqrt{d_k}}), \tag{3}$$

$$\bar{\boldsymbol{H}}_m^n = \boldsymbol{A}\boldsymbol{V}_m. \tag{4}$$

where $\boldsymbol{W}_m$ is a trainable parameter.

Equation (3) combines $\boldsymbol{M}$ and $\boldsymbol{Q}_m {\boldsymbol{K}_m}^T$ with an addition operation to constrain the attending objects of the query. Specifically, if $w_j \nsim w_i$, $\boldsymbol{A}_{i,j}$ is reset to zero after the softmax operation, $\boldsymbol{h}_j^{n-1}$ does not contribute to the representation of w_i (i.e. $\boldsymbol{h}_i^n$). Using the mask mechanism, we can precisely control the information updating on the serialized knowledge evidence and allow the information of each token to propagate following the original structure of the knowledge evidence. Mask $\boldsymbol{M}$ is generated under the guidance of the input structure, thus, the scaled dot-product attention with the mask mechanism is denoted as the structure-aware scaled dot-product attention. Finally, the output of the last layer is adpoted to produce the representation of the input sequence. Formally, given the output of the last layer $\boldsymbol{H}^N = \{\boldsymbol{h}_1^N, \boldsymbol{h}_2^N, \ldots, \boldsymbol{h}_L^N\}$, the final representation of the input sequence is obtained through a max-pooling operation:

$$\boldsymbol{V} = \texttt{maxpooling}(\boldsymbol{H}^N). \tag{5}$$

The TFV and KBQA are formulated as binary classification tasks in this work. Given a statement and table, or a question and candidate answer, the model assigns a label $L \in \{0, 1\}$, where $L = 1$ indicates that the table entails the statement or candidate answer is the correct answer to the question, and $L = 0$ indicates the opposite. The model outputs a score to indicate the probability that the concatenation is predicted to be 1 and uses the score to obtain the binary cross entropy to train the model:

$$\mathcal{L} = \texttt{BCELoss}(L, \texttt{Sigmoid}(\boldsymbol{V}\boldsymbol{W}_s + \boldsymbol{b}_s)). \tag{6}$$

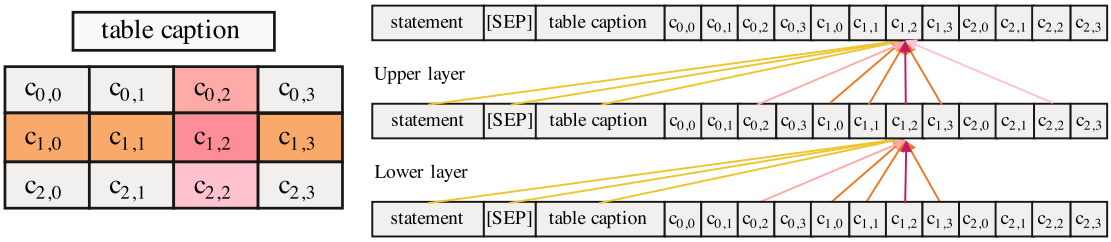
2.2 Table Fact Verification

2.2.1 Task Definition

Given a natural language statement $\boldsymbol{P} = \{p_1, p_2, \ldots, p_{l_p}\}$, TFV aims to classify whether the statement is entailed or refuted by table T. Table T consists of a caption t and cells $\{c_{i,j}; i \le m, j \le n\}$, where m and n are the numbers of rows and columns, respectively. Each $c_{i,j}$ could be different types, such as a word, number, phrase, or natural language sentence. The cells $\{c_{0,j}; j \le n\}$ of the table shown in Fig. 3 (a) indicate the column names which usually describe the field of the column cells. According to table T, a statement $\boldsymbol{P}$ is assigned with a verification label $L_s \in \{1, 0\}$. If $\boldsymbol{P}$ is entailed by T, its label $L_s = 1$, otherwise label $L_s = 0$. Each given statement is verified by matching its semantic representations and the corresponding table, and the statement with a higher matching score is entailed.

2.2.2 Table Serialization

In the TFV task, the input sequence is connected with the statement $\boldsymbol{P}$, the table caption t, and the flattened table T_f, where the table caption helps better understand the background of the statement. In T_f, all table cells are connected



(a) Example of table understanding. The colored row and column are crucial to understanding cell $c_{1,2}$.

(b) Example of local-to-global attention mask of cell $c_{1,2}$. Different colors of the arrows correspond to the different types of tokens. It demonstrates that the column-level attentions are enabled in the upper layers to support cross-row reasoning.

Fig. 3 Example of mask construction for KG.

into a token-level sequence. The table has two serialization ways: using the horizontal scan or the vertical scan. An example of horizontal scanning is presented in Fig. 3 (b). As illustrated in Fig. 3 (a), learning the table cell representation requires both horizontal and vertical views. If the table is flattened using a horizontal scan, the vertical alignment information is lost, and vice versa. For example, the column header $c_{0,2}$ is important to the encoding of $c_{1,2}$, but its signal could be perturbed by other cells in gray, because all $c_{0,*}$ and $c_{2,*}$ cells are far from $c_{1,2}$ in the flattened sequence and are processed equally. Therefore, we construct the attention masks and inject structure information to solve this.

2.2.3 Mask Construction for Tables

Figure 3 (b) sketches the attention masks of cell $c_{1,2}$. Specifically, in the lower layers, the cell representation considers the following four types of information: a) neighbor cells in the same row, b) the column header describing the attribute name, c) the table caption containing the table background, and d) the statement for verification. In the upper layers, column-level attention among cells is further enabled. Lower layers focus on capturing low-level lexical information, whereas upper layers are capable of simple cross-row reasoning. It is worth mentioning that information of the statement $\boldsymbol{P}$ and the table caption t is always visible for all cells of the serialized table.

2.2.4 Symbolic Reasoning on Tables

As mentioned, another preferred ability of SATrans is to perform symbolic reasoning, such as counting, comparing, and calculating. Pretrained transformers, such as BERT, are good at semantic-level understanding but not symbolic reasoning. To solve this problem to some extent, we explore enhancing the performance of counting verification by converting the counting problem into a semantic matching problem. Specifically, for every table column, the frequency of duplicate cell content is counted as a summary cell, leading to a summary row, which is appended to the table.

2.3 Knowledge Base Question Answering

2.3.1 Task Definition

Given a natural question $\boldsymbol{P}$=$\{p_1, p_2, \ldots, p_{l_p}\}$ and a KG G, KBQA aims to determine the answers $\mathcal{A} = \{a_1, \ldots, a_{l_a}\}$ to $\boldsymbol{P}$ from G. Typically, a knowledge base consists of a set of facts, and each fact is a triplet that includes a subject entity, relation, and object entity. Some triplets share the same entity, therefore, the knowledge base is always considered a graph. When solving the KBQA task, a set of topic entities $\mathcal{S} = \{s_1, \ldots, s_{l_s}\}$ is identified using the entity linking operation between the question and knowledge base. Then, a KB subgraph can be extracted around the topic entities. In this work, the subgraph extracted for each question is defined as "question subgraph" G_q. Specifically, G_q comprises n-hop paths around the topic entities $\mathcal{S}$, and all entities $\mathcal{O} = \{o_1, \ldots, o_{l_o}\}$ in it comprise the candidate answer set of the question. Besides the question subgraph, we also define a "candidate subgraph", $G_c(\bar{o})$, for each candidate answer, where $G_c(\bar{o}) \subset G_q$. The candidate subgraph of o_i consists of all paths between o_i and one of the topic entities $s_i \in \mathcal{S}$. In this work, the candidate subgraph is employed as the knowledge evidence to determine whether the corresponding candidate entity is a correct answer. Like the TFV task, the KBQA task is treated as a semantic matching problem between a question and its candidate subgraph.

2.3.2 Candidate Subgraph Serialization

In the KBQA task, the input of SATrans is the concatenation of question $\boldsymbol{P}$, and the serialized candidate subgraph $P_{\bar{o}} = \{\bar{s}, p_1, \ldots, p_n, \bar{o}\}$. When serializing a candidate subgraph, we first splice all the paths between the topic entity $\bar{s}$ and the corresponding candidate entity $\bar{o}$ into a word sequence. Then, $\bar{s}$ and $\bar{o}$ are placed at the head and tail of the sequence. Figure 5 reveals that given the topic entity "*Raphael*", the candidate entity "*Italy*" and the two paths between them, the four parts are concatenated to obtain the serialized candidate subgraph P_{Italy}. The serialization operation destroys the structural information in the candidate subgraph. The node "*nationality*" is directly connected to "*Raphael*" in the graph in Fig. 4. However, in the serialized

candidate subgraph, they are far from each other.

2.3.3 Mask Construction for Candidate Subgraphs

A candidate subgraph consists of a set of triplets. Thus, the tokens of $P_{\bar{o}}$ include three types: subject, relation, and object. Based on this, multiple triplets connect into paths consisting of the subgraph through shared entities. As Fig. 6 reveals, the lower-layer attention mask is constructed to ensure the representation learning of each token considers two aspects of information: a) the triplets to which it belongs and b) the question to be answered. Question $\boldsymbol{P}$ receives information from all tokens of $P_{\bar{o}}$. In the upper layers, path-level attention is enabled. This way, triplets information on the same reasoning path can be integrated, and the cross-path information will propagate through shared entities. Figure 6 presents representation learning with the masked self-attention of the relation token *place_of_birth* in P_{Italy}. In the lower layers, the entity '*place_of_birth*' can only view the triplet (*Raphael, place_of_birth, Urbino*), which makes the self-attention focus on the local lexical information of the one-hop neighbors of the entity. In the upper layers, '*place_of_birth*' can further view the path (*Raphael, place_of_birth, Urbino, location_contain, Italy*). In this way, the model can learn the multi-hop range reasoning of the entity. In general, under the control of the attention mask, the information on the serialized subgraph flows from triplet-level to path-level based on the graph structure.

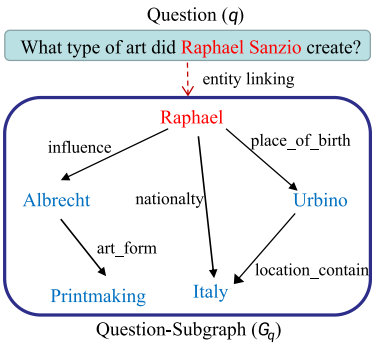


Fig. 4 An example of a given question q and its corresponding question subgraph G_q. In G_q, the entity 'Raphael' is the topic entity obtained by entity linking operation, all of the other entities in the subgraph are regarded as the candidate answers. Notably, the relation names shown in this figure is a simplified version of FreeBase relations.

2.3.4 Chain-Guide Training

WebQSP provides the core inference chain [12], which indicates crucial paths from the topic entity to the correct answers. The SATrans cannot directly determine the inference paths, thus, an auxiliary task is introduced to convert symbolic reasoning into linguistic reasoning. This task aims to predict whether a path in the candidate subgraph is the core inference chain and uses this information to boost the training of the SATrans. We denote this mechanism as chain-guide training. Specifically, given a candidate subgraph with n_p paths, the SATrans is adopted to learn the representation of the serialized sequence $\boldsymbol{H}^N$. Then, the average values of the representations corresponding to each path are denoted as semantic representations $\boldsymbol{H}_p = \{\boldsymbol{h}_{p_1}, \boldsymbol{h}_{p_2}, \ldots, \boldsymbol{h}_{p_{n_p}}\}$. Afterward, the representation of each path is input into a classifier to output a score to indicate the probability that the input path is the core inference chain:

$$s_{p_i} = \texttt{Sigmoid}(\boldsymbol{h}_{p_i}\boldsymbol{W}_p + \boldsymbol{b}_p). \tag{7}$$

The binary cross entropy is adopted as the objective function when training:

$$\bar{\mathcal{L}}_p = \frac{1}{n_p}\sum_{i=1}^{n_p} \texttt{BCELoss}(L_{p_i}, s_{p_i}), \tag{8}$$

where L_{p_i} denotes whether the path p_i is the core inference

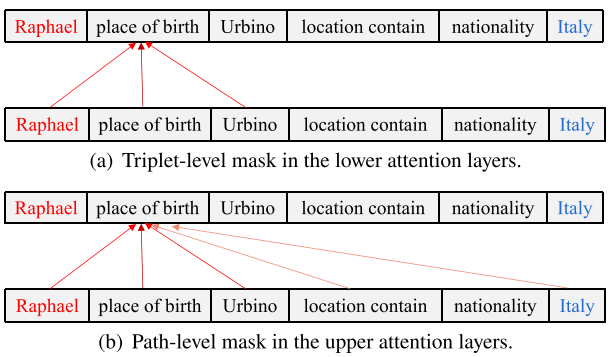


Fig. 6 Take the serialized subgraph of 'Italy' as the example, this figure gives the masked self-attention of tokens in 'place_of_birth'. In the lower layers, its visible range includes its triplet. Information of different triples in the same path is enabled in the upper layers. The question is always visible for each token, which is omitted in this figure.

Candidate entities (o)	Candidate-Subgraph ($G_o(o)$)	Serialized Subgraph (P_o)
Albrecht	p_1: Raphael → [influence] → Albrecht	Raphael influence Albrecht
Urbino	p_1: Raphael → [place_of_birth] → Urbino	Raphael place of birth Urbino
Printmaking	p_1: Raphael → [influence] → Albrecht → [art_form] → Printmaking	Raphael influence Albrecht art form Printmaking
Italy	p_1: Raphael → [place_of_birth] → Urbino → [location_contain] → Italy p_2: Raphael → [nationality] → Italy	Raphael place of birth Urbino location contain nationality Italy

Fig. 5 Candidate subgraph before and after serialization of each candidate entity in Fig. 4.

Table 1 The verification accuracy (%) of different models. † means the results are cited from publications. Results of *Table-BERT tuned** are obtained by tuning the learning rate from 5e-5 to 1e-5.

Model	Val	Test	Test(simple)	Test(complex)
LPA [9]†	65.1	65.3	78.7	58.5
Table-BERT [9]†	66.1	65.1	79.1	58.2
Table-BERT tuned*	68.38	68.30	82.35	61.48
BERT with cell position encoding	59.31	59.44	63.24	57.58
SATrans with Horizontal scan	72.96	72.82	85.44	66.62
- w/o attention mask	68.41	67.67	75.93	63.61
- w/o summary row	72.00	72.09	85.53	65.49
- w/o attention mask w/o summary row	66.84	66.01	74.37	61.90
SATrans with Vertical scan	**73.31**	**73.23**	**85.46**	**67.23**
- w/o our attention mask	64.21	64.27	68.77	62.06
- w/o summary row	71.71	71.59	84.70	65.15
- w/o summary row and w/o our attention mask	63.03	62.34	66.71	60.19
- all layers w/ fix global mask (row and column)	72.83	72.26	84.61	66.11
- all layers w/ fix local mask (only row)	72.02	71.82	83.45	66.10

chain. If it is, L_{p_i} equals 1, otherwise, it is 0.

Finally, we adopt the linear interpolation of the QA loss computed in Eq. (6) and core inference chain loss computed in Eq. (8) are employed as the loss function for the SATrans:

$$\bar{\mathcal{L}}_{final} = \alpha\mathcal{L} + (1-\alpha)\mathcal{L}_p, \tag{9}$$

where α is a coefficient to weight the importance of $\mathcal{L}$ and $\mathcal{L}_p$ during training.

3. Experiments

In this section, we mainly report our main experimental results to verify the effectiveness of SATrans. More comparative analysis and discussion are given in Sect. 4.

3.1 Implementation Details

3.1.1 Dataset

The TFV experiments are conducted using TabFact† [9], a large-scale TFV dataset. Its instances are split into 92238, 12792 and 12779 respectively for training, validation and testing. TabFact includes simple and complex statements. Simple statements only contain a single row, while complex statements involve higher-order semantics, and the statements require more ability on symbolic reasoning. The KBQA experiments are conducted on WebQSP [12], which has 3098 instances for training and 2032 instances for testing. WebQSP is built on Freebase††. Each question in WebQSP is further annotated in the core inference chain, defined as the path connecting a topic entity to a correct answer.

3.1.2 Settings

The SATrans weights are initialized using a BERT-based model with 12 self-attention layers. In this work, we select the first 6 layers as the lower layers and the last 6 layers as the upper layers. The experiments of layer division are reported in 3.2.4. In model training, all the training parameters are consistent with the baseline models of the two tasks. Specifically, for the TFV task, the model is fine-tuned with a learning rate of 2e-5 and a batch size of 10. The maximum sequence length is set to 256.

For the KBQA task, the entity linking results are from the S-MART [14] system, and the top entity is selected as the topic entity of the question. The two-hop range paths around the topic entity are retrieved as the question subgraph using the Personalized PageRank (PPR) method [15]. In the question subgraph, the top300 entities are selected as candidate answers by ranking the PPR scores. For model training, the batch size is set to 20, the maximum sequence length is set to 512, and the learning rate is set to 5e-5. Negative sampling with a 0.1 sample rate is adopted to address the data imbalance problem, which is selected by the grid search method in the 0.1-0.5 with 0.1 interval. For chain-guided training, the weight coefficient α in Eq. (9) is 0.5.

3.2 Results on TabFact

We use accuracy as the evaluation metric to compare the verification performance of SATrans and the baseline models. The experimental results on the TabFact are listed in Table 1. The performance of SATrans is reported using horizontal and vertical scans, and the only difference between these two settings is the scanning method in the table serialization process. The two settings use the same attention mask construction method, which uses row-granularity attention in the lower layers and adds column-granularity attention in the upper layers. As the results revealed, SATrans, with the vertical scan, achieves the best accuracy of 73.23% on the complete testing set and outperforms Table-BERT by 4.93%. The improvement on complex statements is even more significant, achieving a 5.75% improvement.

3.2.1 Effect of Attention Mask

After turning off the attention mask, the testing accuracy values are 67.67% and 64.27% for horizontal and vertical

†https://github.com/wenhuchen/Table-Fact-Checking
††https://developers.google.com/freebase

scans, respectively, a decrease of 5.15% and 8.96% compared to the complete SATrans. These gains are mainly from the ability of SATrans to select actual neighbor information for each cell in the flattened table. In other words, it weakens the influence of the pseudo-neighbors adjacent to the cell in the sequence but irrelevant to the cell in the structured knowledge. Moreover, the local-to-global mask helps the model learn the lexical knowledge and reasoning skills first, which further benefits the table representation ability and verification performance.

From the results, without the attention mask, flattening tables using horizontal scan achieves better performance than using vertical scan. The results are consistent with our intuition that row-level neighbors are more semantic related. Thus, row-level structural information is more critical. In the setting of only considering the column-level information, the gap is smaller when using SATrans, demonstrating its robustness towards different scan directions.

3.2.2 The Summary Row

Appending a summary row to the table brings 1% improvement to the verification accuracy. The improvement is stable in complex instances. This gain indicates that the pre-trained transformer lacks the ability of symbolic reasoning, although they have an advantage in semantic understanding. With the counting problem in scope, the experimental results reveal that converting the symbolic reasoning problem into semantic understanding by inputting symbolic reasoning results into SATrans is promising.

3.2.3 Case Study

We collected and analyzed instances fixed by SATrans compared to baselines. It is observed that a large portion (43/80) of them involve multiple table cells and require no logical reasoning. In addition, several instances (9/80) that require a simple count and comparison are fixed. The model was fixed (the other 38) and failed on some samples requiring complex symbolic reasoning, such as counting, intersecting, and comparison. The behavior is likely a random guess for both the SATrans and baselines. The results demonstrate that SATrans enhances the table representation and symbolic reasoning abilities, and the appended summary row benefits solving numerical reasoning problems to some extent.

3.2.4 Attention Layers Division

It is proved that designing different mask matrices for layers of SATrans improves performance. The results in Table 1 are under the setting, where the first and last 6 layers are regarded as the lower and upper layers, respectively. To select the best division of the lower and upper attention layers, we attempt various layer splits and report the verification results in Table 2. The number of lower layers is denoted as L_{low} successively set from 0 to 12 with 2-layer intervals, and the other $(12-L_{low})$ layers are the upper layers. Table 2 indicates that the model achieves the best accuracy on all the testing sets when $L_{low} = 6$, due to the model's ability to learn row and column information thoroughly for the representation of each cell.

Table 2 The accuracy (%) of different layer split.

L_{low}	Val	Test	Test(simple)	Test(complex)
0	72.02	71.82	83.45	66.10
2	72.71	72.72	84.53	66.91
4	72.70	72.95	85.08	66.99
6	**73.31**	**73.23**	**85.46**	**67.23**
8	72.43	72.34	84.39	66.41
10	71.13	71.92	82.95	66.51
12	72.83	72.26	84.61	66.11

3.3 Results on WebQSP

The F1 score is used as the metric of question-answering performance of the SATrans and baseline models. We compare the SATrans with a robust KBQA model, Graft-Net, and use the same entity linking results to compare the question-answering performance. Since Graft-Net uses two different encoders to encode the question and graphs, respectively, we replace its question encoder with BERT to demonstrate the gain from the PTMLs. The results reveal that the BERT encoder improves the performance of the baselines by 1.2% and 0.5% for the F1 scores. In contrast, our model achieves 65.7% F1 scores on the test set, outperforming the baseline models with or without BERT. These gains prove the effectiveness of our model better to apply the pre-trained model to the KBQA task.

3.3.1 Effect of the Attention Mask

Without the attention mask, the F1 score of SATrans drops from 65.7% to 62.6%. These decreases indicate that SATrans effectively represents graph-structured knowledge better and improves the end-to-end question-answering performance. To further analyze the effectiveness of the local-to-global attention mask, we compare it with only using masks in all attention layers, which store local and global information. The local mask makes the encoding process of each token consider its triplet-level information, while the global mask allows the token encoding process to consider the path-level information. As shown in Table 3, our local-to-global mask outperforms the baselines, proving that the proposed method improves graph representation learning.

The results also show that only considering triplet-level information achieves a better F1 score than path-level information. The reason is that fine-grained neighbors are more related to the target token, which can enhance the token representation. In this case, the path-level and graph-level structure information are all implicitly learned through the shared entities among the triplets. In contrast, considering a more coarser-grained neighbor for each token from the beginning will bring the noise to its representation learning, which leads to worse performance.

Table 3 The F1 score on WebQSP of different models.

Model	Test
STAGG [14]	52.5
MULTIQUE [16]	61.2
GRAFT-Net [17]	62.5
GRAFT-Net w/ BERT encoder†	63.7
SATrans	**65.7**
- w/o attention mask	62.6
- w/o chain-guide training	64.6
- all layers w/ fix global (path) mask	63.7
- all layers w/ fix local (triplet) mask	64.1

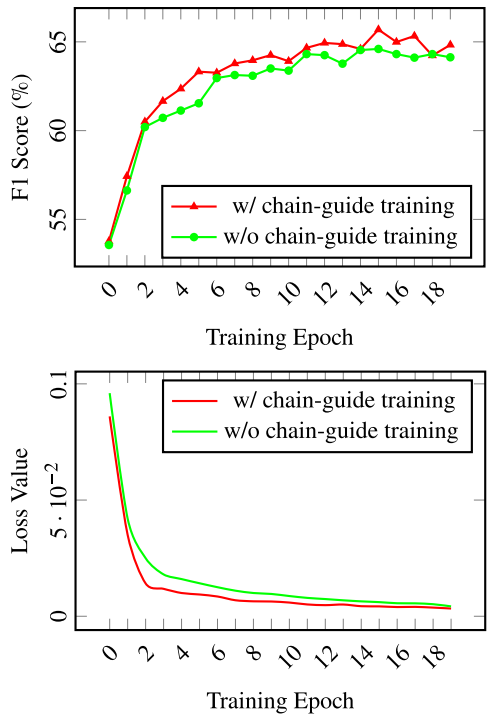


Fig. 7 Trends of F1 score and loss during the training process with or without the chain-guide training.

3.3.2 The Chain-Guide Training

As the last row in Table 3 illustrates, without the chain-guide training, the F1 score achieved by SATrans on WebQSP declines by 1.1%. This phenomenon is because chain-guide training can improve the reasoning ability of SATrans. In addition, from the loss and F1 score curves shown in Fig. 7, the chain-guide training method can improve model performance and accelerate the convergence of the model training. The experiment results prove that learning to determine whether an inference chain exists in the candidate subgraph helps the model determine the correct answers.

4. Discussion with Related Work

4.1 Table Representation and TFV

Since TabFact was proposed [9], it has attracted much research attention to fact verification over structured knowledge. In this paper, we mainly follow two promising approaches studied on Tabfact. One is Table-BERT, which converts the table understanding task into a natural language inference task using the ability of PTLMs. The other one is Latent Program Algorithm (LPA), which solves this task as a logic program parsing problem due to its advantage in symbolic reasoning. Our work follows the same aspect of Table-BERT, and we devise a local-to-global structure-aware transformer to obtain better table representation. Besides, our work also benefits from [18] and [19]. These works enhance the symbolic reasoning ability of the PTLMs. We can directly adopt their enhanced PTLMs in our method.

More recently, TAPAS, an effective table parsing pretrained model, is proposed by [20]. It solves table-based tasks well and achieves a significant result by 81.0† accuracy on TabFact. Many works adopt TAPAS as their backbone model and further improve the verification performance on TabFact from various aspects [21], [22]. In the following, we mainly discuss the differences between TAPAS and SATrans and analyze the advantages of SATrans.

Firstly, an intuitive difference is that TAPAS requires expensive pretraining with large-scale tabular data, which makes TAPAS a dedicated model for table understanding. In contrast, SATrans does not require any additional data for pretraining. It can be flexibly applied as a plug-in to existing pre-trained transformers to help them understand structured knowledge. Besides, SATrans is not limited to table processing. In this paper, we have shown that it works equally well for representing graph structures. This ability enables it to be applied to understanding heterogeneous knowledge. For example, recent work [23] adopts our method into BART [24] for logic form generation and obtains 4.2 F1 score improvement on TAT-QA [25], a dataset of question-answering over tabular and textual data, that proves the above views.

Moreover, TAPAS captures tabular structure by extending BERT's architecture with additional position embeddings. To identify whether the table position encoding is better than our method, we conduct experiments where row and column positional embeddings are added to the original positional embeddings of BERT to identify the table alignment information. The experimental results are listed in the fourth row of Table 1. BERT with cell position encoding achieves 59.8% accuracy, while the performance of baseline BERT is 68.3%. The results indicate that BERT is perturbed by the additional table positional embeddings, and the model did not converge well. Though the table position information is appended to the inputs, the following transformer layers are not ready to accept and propagate the signal without pretraining. It is demonstrated that simply providing positional information without pretraining is not sufficient for Transformer to encode tables.

†The result is from [21].

4.2 Graph Representation and KBQA

Intuitively, methods based on the GNN [8], [26], [27] are widely used to encode knowledge graphs that are also "structure-aware". Specifically, GNN is designed to encode graph-structured data, which can aggregate neighbor information for each node. However, compared with transformers, the GNN-based models cannot capture long-distance information at the sequence level, so they can not be pretrained using a large corpus. Therefore, the structure-aware transformer has more advantage of learning semantic information than the GNN-based model. Recently, leveraging PTLMs to encode structured data has attracted more and more research attention.

A representative work, K-BERT [28], explores structured knowledge representation and first injects the structural information into the attention mask. Specifically, K-BERT adopts a fixed attention mask stored global structure in all self-attention layers. In contrast, our SATrans leverage the local-to-global mask to make the model learn the local structure in the lower attention layers and integrate the global information in the upper global layers. Our experiments have proved that our designation is effective.

For KBQA, we mainly compare our SATrans with GRAFT-Net, since both the models are neural network (NN)-based, which encodes the question and the KG graphs into vectors and measures their semantic similarities to select the answer. There are also many semantic parsing (SP)-based baselines [29], [30], which derive answers from KG by generating query graphs or executable logic forms. They achieve better performance on WebQSP by a 74.0 F1 score. Despite their performance on QA tasks, SP-based methods are heavily dependent on expensive work for logic form annotations. In contrast, our SATrans, designed to enhance PTLMs, achieve comparable performance with those refined-designed models specifically for QA. We think that the two SP-based approaches and our model pay attention to different perspectives for solving the KBQA problems, and it's probably to combine them to achieve better performance.

5. Conclusion

This work proposes a local-to-global structure-aware transformer to better apply the pretrained transformer in question-answering tasks over structured knowledge. The SATrans injects the structural knowledge information into the attention mask of the self-attention layers and allows the lower and upper attention layers to focus on local and global information for each token. Furthermore, to fill the gap that the SATrans lack symbolic reasoning ability, we explored two methods to combine symbolic reasoning and linguist reasoning. Extensive experiments were conducted on two structured knowledge-based question-answering tasks (i.e., TFV and KBQA) to evaluate the proposed method. The results revealed that the proposed method outperforms strong baselines on these two benchmarks.

Acknowledgments

The work of Tiejun Zhao was supported by the National Key Research and Development Program of China under Grant 2018YFC0830700.

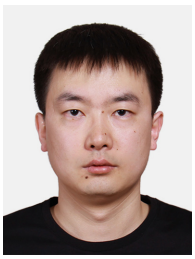
References

[1] J. Devlin, M.W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, 2018.
[2] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R.R. Salakhutdinov, and Q.V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," Advances in neural information processing systems, pp.5753–5763, 2019.
[3] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," arXiv preprint arXiv:1909.11942, 2019.
[4] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," arXiv preprint arXiv:1907.11692, 2019.
[5] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, "Ernie: Enhanced language representation with informative entities," arXiv preprint arXiv:1905.07129, 2019.
[6] K. Clark, M.T. Luong, Q.V. Le, and C.D. Manning, "Electra: Pre-training text encoders as discriminators rather than generators," arXiv preprint arXiv:2003.10555, 2020.
[7] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P.J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," arXiv preprint arXiv:1910.10683, 2019.
[8] F. Scarselli, M. Gori, A.C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," IEEE Transactions on Neural Networks, vol.20, no.1, pp.61–80, 2008.
[9] C. Wenhu and W. Hongmin, "Tabfact: A large-scale dataset for table-based fact verification," International Conference on Learning Representations (ICLR), Addis Ababa, Ethiopia, April 2020.
[10] Q. Chen, F. Ji, H. Chen, and Y. Zhang, "Improving commonsense question answering by graph-based iterative retrieval over multiple knowledge sources," arXiv preprint arXiv:2011.02705, 2020.
[11] W. Liu, P. Zhou, Z. Zhao, Z. Wang, Q. Ju, H. Deng, and P. Wang, "K-bert: Enabling language representation with knowledge graph," arXiv preprint arXiv:1909.07606, 2019.
[12] W.-t. Yih, M. Richardson, C. Meek, M.-W. Chang, and J. Suh, "The value of semantic parse labeling for knowledge base question answering," Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp.201–206, 2016.
[13] H. Zhang, Y. Wang, S. Wang, X. Cao, F. Zhang, and Z. Wang, "Table fact verification with structure-aware transformer," Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, pp.1624–1629, Association for Computational Linguistics, Nov. 2020.
[14] W.-t. Yih, M.-W. Chang, X. He, and J. Gao, "Semantic parsing via staged query graph generation: Question answering with knowledge base," Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Beijing, China, pp.1321–1331, Association for Computational Linguistics, July 2015.
[15] T.H. Haveliwala, "Topic-sensitive pagerank," Proceedings of the 11th international conference on World Wide Web, pp.517–526, 2002.

[16] N. Bhutani, X. Zheng, K. Qian, Y. Li, and H. Jagadish, "Answering complex questions by combining information from curated and extracted knowledge bases," Proceedings of the First Workshop on Natural Language Interfaces, Online, pp.1–10, Association for Computational Linguistics, July 2020.
[17] H. Sun, B. Dhingra, M. Zaheer, K. Mazaitis, R. Salakhutdinov, and W.W. Cohen, "Open domain question answering using early fusion of knowledge bases and text," arXiv preprint arXiv:1809.00782, 2018.
[18] M. Geva, A. Gupta, and J. Berant, "Injecting numerical reasoning skills into language models," Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp.946–958, 2020.
[19] A. Asai and H. Hajishirzi, "Logic-guided data augmentation and regularization for consistent question answering," Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp.5642–5650, 2020.
[20] J. Herzig, P.K. Nowak, T. Müller, F. Piccinno, and J. Eisenschlos, "TaPas: Weakly supervised table parsing via pre-training," Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, pp.4320–4333, Association for Computational Linguistics, July 2020.
[21] F. Wang, K. Sun, J. Pujara, P. Szekely, and M. Chen, "Table-based fact verification with salience-aware learning," arXiv preprint arXiv:2109.04053, 2021.
[22] J. Eisenschlos, S. Krichene, and T. Müller, "Understanding tables with intermediate pre-training," Findings of the Association for Computational Linguistics: EMNLP 2020, Online, pp.281–296, Association for Computational Linguistics, Nov. 2020.
[23] Y. Zhou, J. Bao, C. Duan, Y. Wu, X. He, and T. Zhao, "Unirpg: Unified discrete reasoning over table and text as program generation," arXiv, 2022.
[24] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, pp.7871–7880, Association for Computational Linguistics, July 2020.
[25] F. Zhu, W. Lei, Y. Huang, C. Wang, S. Zhang, J. Lv, F. Feng, and T.S. Chua, "Tat-qa: A question answering benchmark on a hybrid of tabular and textual content in finance," arXiv preprint arXiv:2105.07624, 2021.
[26] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," arXiv preprint arXiv:1710.10903, 2017.
[27] S. Yun, M. Jeong, R. Kim, J. Kang, and H.J. Kim, "Graph transformer networks," Advances in neural information processing systems, vol.32, 2019.
[28] L. Weijie and Peng, "K-BERT: Enabling language representation with knowledge graph," Proceedings of AAAI 2020, 2020.
[29] Y. Lan and J. Jiang, "Query graph generation for answering multi-hop complex questions from knowledge bases," Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp.969–974, Association for Computational Linguistics, 2020.
[30] X. Huang, J.-J. Kim, and B. Zou, "Unseen entity handling in complex question answering over knowledge base via language generation," Findings of the Association for Computational Linguistics: EMNLP 2021, Punta Cana, Dominican Republic, pp.547–557, Association for Computational Linguistics, Nov. 2021.

Yingyao Wang is currently working toward a Ph.D. degree in the School of Computer Science and Technology, Harbin Institute of Technology, China. Her research interests include relation extraction and question-answering. She has published many international conference papers in her research area, such as COLING, EMNLP.

Han Wang received the B.E. degree in Communication Engineering from Minzu University of China in 2016. He is currently a Ph.D. student at the Institute of Acoustics, Chinese Academy of Sciences and University of Chinese Academy of Sciences, Beijing, China. His research interests include incremental/lifelong/continual learning, dialogue system, and text classification.



Chaoqun Duan received the B.S. degree in computer science from Shanxi University, China in 2013 and the M.S. degree in computer science from Harbin Institute of Technology, China in 2015. He received the Ph.D Degree in Harbin Institute of Technology in 2022. His research interests include machine translation and natural language processing.

Tiejun Zhao received his Ph.D. Degree in 1992 from the Department of Computer Science, Harbin Institute of Technology, Harbin, China. He is a full professor of Department of Computer Science, and the director of Machine Intelligence&Translation (MITLab) from Harbin Institute of Technology. His research interests include machine translation, question answering.