Transferring Knowledge from Structure-aware Self-attention Language Model to Sequence-to-Sequence Semantic Parsing

Ran Ji and Jianmin Ji*
University of Science and Technology of China, Hefei, China
jiran@mail.ustc.edu.cn, jianmin@ustc.edu.cn

Abstract
Semantic parsing considers the task of mapping a natural language sentence into a target formal representation, where various sophisticated sequence-to-sequence (seq2seq) models have been applied with promising results. Generally, these target representations follow a syntax formalism that limits permitted forms. However, it is neither easy nor flexible to explicitly integrate this syntax formalism into a neural seq2seq model. In this paper, we present a structure-aware self-attention language model to capture structural information of target representations and propose a knowledge distillation based approach to incorporating the target language model into a seq2seq model, where grammar rules or sketches are not required in the training process. An ablation study shows that the proposed language model can notably improve the performance of the baseline model. The experiments show that our method achieves new state-of-the-art performance among neural approaches on four semantic parsing (ATIS, GEO) and Python code generation (Django, CoNaLa) tasks.

1 Introduction
Semantic parsing aims to map a natural language sentence into a machine executable formal representation, which has been considered as one of the prime challenges nowadays in natural language processing (NLP). These target formal representations can generally be divided into three categories (Kantham and Das, 2018), i.e., logical forms, like first order sentences or λ-calculus expressions (Zettlemoyer and Collins, 2005), programming language statements, like Python code or SQL programs, and graph-based forms, like labeled graphs in Abstract Meaning Representation (AMR) (Banarescu et al., 2013). In this paper, we focus on semantic parsing that yields logical forms.

Target logical forms often follow a syntax formalism that limits permitted formulas, which can be used to filter the output and improve the performance of semantic parsing. For example, in the pre-neural era, CCG based approaches (Kwiatkowski et al., 2013) achieved significant performance gains by introducing a linguistically motivated grammar induction scheme. Some neural semantic parsers (Yin and Neubig, 2018; Sun et al., 2020) first transduce the natural language utterance into an Abstract Syntax Tree (AST), then serve it as an intermediate meaning representation to incorporate with grammar rules for the target logical form. Semantic parsing can also be considered as a seq2seq transduction problem, where the decoder can leverage structural features of target representations. In particular, hierarchical tree decoders are applied in (Dong and Lapata, 2016; Alvarez-Melis and Jaakkola, 2017; Sun et al., 2019) to take into account the tree structure of the logical expression. Decoders constrained by a grammar model are applied in (Xiao et al., 2016; Yin and Neubig, 2017; Krishnamurthy et al., 2017; Dong and Lapata, 2018). The uncertainty-driving adaptive decoding is used to guide the decoder in (Zhang et al., 2019). Relatively sizeable monolingual corpus of the target programming language is used in (Norouzi et al., 2021) to improve performance.

Note that, manually specified grammar rules and sketches for target logical forms are required in most of these approaches, which limits their adaptabilities and scalabilities to a new semantic parsing task with updated target logical forms. In this paper, we consider using a structure-aware language model to capture formal patterns for target representations and incorporating the language model into seq2seq models for semantic parsing.

We first train the structure-aware language model on target logical forms to capture structural information. Then, we incorporate the language model to a seq2seq model for semantic parsing.

*The corresponding author.
Integrating a language model into a seq2seq model has been considered in automatic speech recognition (ASR) and neural machine translation (NMT). In particular, shallow fusion and deep fusion (Gulcehre et al., 2015) are two such approaches in NMT. Cold fusion (Sriram et al., 2018) is tested on ASR tasks. Bai et al. (2019) proposes a knowledge distillation based training approach to transferring knowledge from a language model to a seq2seq model for ASR. Here, we follow the knowledge distillation structure to integrate the language model to the baseline seq2seq model for semantic parsing.

We evaluate our approach on two semantic parsing datasets, ATIS (Dahl et al., 1994) and GEO (Zelle and Mooney, 1996) datasets, where target logical forms are λ-calculus expressions and two code generation tasks, Django (Oda et al., 2015) and CoNaLa (Yin et al., 2018), where target logical forms are Python code. We train the target language model based on target logical forms. The experimental results show that our approach achieves state-of-the-art performance among neural network based approaches on ATIS, GEO, Django and CoNaLa datasets.

In this paper, we show that the proposed language model can be used to capture structural features of target logical forms and the knowledge distillation structure can be used to transfer knowledge to a seq2seq model for semantic parsing, where manually specified grammar rules or sketches are no longer required. Notice that, this approach can be applied to various sophisticated seq2seq models, which results a more flexible and scalable method for neural semantic parsers to leverage structural features of target representations. The main contributions of the paper are summarized as follows:

- We propose a structure-aware self-attention language model to capture structural information of target logical forms.
- We propose a knowledge distillation structure to transfer knowledge from target language model to a seq2seq model, which suggests a more flexible and scalable method for neural semantic parsers to leverage structural features of target representations.
- We implement the approach on baseline seq2seq models, which achieves new state-of-the-art performance among neural semantic parsers on ATIS, GEO, Django and CoNaLa datasets.

2 Related Work

2.1 Neural Semantic Parsing

Neural semantic parsing has achieved promising results in recent years. In particular, AST based parsers (Yin and Neubig, 2018; Sun et al., 2020, 2019) first map a nature language sentence into an abstract syntax tree (AST), then parse the AST to the corresponding target logic form. On the other hand, seq2seq based semantic parsers often leverage structural features of natural language sentences or target representations to improve the performance. Specifically, a sequence-to-tree (seq2tree) model (Dong and Lapata, 2016) updates the decoder into a hierarchical LSTM tree, which helps the model to utilize the hierarchical structure of logical forms. A graph-to-sequence (graph2seq) model (Xu et al., 2018) updates the encoder into a graph encoder. Graph neural networks (GNNs) are also used in semantic parsing (Shaw et al., 2019) to incorporate information about relevant entities and their relations during the parsing. A sequence-to-action (seq2action) model (Chen et al., 2018) considers semantic parsing as an end-to-end semantic graph generation process. A coarse-to-fine (coarse2fine) model (Dong and Lapata, 2018) decomposes the decoding process into two stages. The first stage predicates a rough sketch of the meaning representation and the second stage fills in missing details conditioning on the natural language input and the sketch itself. The AdaNSP model (Zhang et al., 2019) proposes an adaptive decoding method to avoid intermediate representations in the parsing process, where the decoder is guided by the model uncertainty. TAE (Norouzi et al., 2021) exploit a relatively sizeable monolingual corpus of the target programming language to improve performance.

Notice that, manually specified grammar rules or sketches are required in most of these neural semantic parsing approaches to leverage structural features of natural language sentences or target representations. In this paper, we consider using the proposed target language model to capture these formal patterns and incorporating the language model into seq2seq models for semantic parsing.
2.2 Structural Language Models

In recent years, language models that capture structural information in natural language have been developed. (Shen et al., 2018) proposed a PRPN (Parsing-Reading-Predict Networks) model, which uses the syntactic structure information of natural language to better perform language modeling. The model is divided into three parts: parsing module, reading module and prediction module. The parsing module uses the convolutional neural network to predict the syntactic distance of two adjacent words, and obtains the syntactic tree of the sentence through the syntactic distance; the reading module uses the syntactic tree obtained by the parsing module to model the context; the prediction module predicts the next word. The PRPN model achieved good results at the time on both unsupervised syntactic analysis tasks and language model modeling.

(Shen et al., 2019) proposed the ON-LSTM (Ordered Neurons-LSTM) model, which gives LSTM neuron level information to model the hierarchical structure information of sentences. The author believes that the level of a word is related to its span in a sentence. The higher the level, the larger the span, so words with higher levels should be retained longer and are not easily updated. So the model proposes a new LSTM neuron: the ordered neuron, which enforces the order in which the neurons are updated. All lower-order neurons must be deleted before higher-order neurons can be deleted or updated, thus controlling the update frequency of neurons. The ON-LSTM model achieves good performance on four different tasks, language modeling, unsupervised parsing, target grammar evaluation, and logical reasoning.

(Wang et al., 2019) proposed the Tree Transformer model, which improved the Transformer to learn syntactic information in natural language. The Tree Transformer adds an additional constraint of "Constituent Attention" to the attention head of the Transformer’s encoder to enhance attention to natural language tree structures. The component constraint module judges whether two adjacent words can form a phrase, and if so, assigns more attention scores to these two words. Tree Transformer is designed for natural language parsing tasks and has achieved good results on unsupervised parsing tasks. In addition to the syntactic analysis task, the author also changed the Tree Transformer into a mask language model, and compared it with BERT on the corpus WSJ. Since the syntactic information can be learned in the Tree-Transformer, the effect of the language model is better than that of BERT.

(Li et al., 2021) proposed a StructuralLM model to improve BERT to learn structural information in documents. StructuralLM treats each cell in the document as a semantic unit, and then makes the model’s training goal to classify the cell location to take full advantage of the cell and layout information. The pre-trained StructuralLM model achieved state-of-the-art results on three downstream tasks: form understanding, document visual question answering, and document image classification.

In this paper, we propose the structure-aware language model that use structure-aware self-attention to explicitly capture the structural information of the target forms.

2.3 Integrating Language Model into Seq2Seq Models

Integrating a language model into a seq2seq model has been considered in multiple NLP tasks, like automatic speech recognition (ASR) and neural machine translation (NMT). In particular, shallow fusion and deep fusion (Gulcehre et al., 2015) are proposed to integrate a language model into a seq2seq model. Both methods first train a language model and a translation model separately, then use the language model in the inference step. Specifically, shallow fusion performs a log-linear interpolation between the decoder and the language model to re-weight the translation model’s scores during the beam search. Deep fusion concatenates the language model and decoder’s hidden states next to each other, then uses the the hidden states to fine-tune the model. Cold fusion (Sriram et al., 2018) is tested on AST tasks. Cold fusion uses the logic outputs of the trained language model as features to train the translation model. Simple fusion (Stahlberg et al., 2018) uses the output of a trained language model together with the output of a translation model to train the translation model. Component fusion (Shan et al., 2019) first trains a source language model, later freezes the source language model and trains the translation model, then replaces the source language model with a target language model in the inference process.

The LST (Learning Spelling from Teachers) approach (Bai et al., 2019) proposes a knowledge distillation based training approach to transferring
knowledge from a language model to a seq2seq model for ASR. It first trains a recurrent neural network based language model (RNNLM) on large scale external text, then considers the RNNLM as the teacher to generate soft labels of speech transcriptions to train the decoder in the seq2seq model.

In this paper, we follow the knowledge distillation structure to transfer knowledge from target language model to the decoder of a baseline seq2seq model for semantic parsing. Different from LST, a new Transformer-based structure-aware language model is considered here, which can capture structural information of formal patterns for target representations. We show that the approach achieves new state-of-the-art performance on ATIS, GEO, Django and CoNaLa datasets.

3 Preliminaries

3.1 A Seq2Seq Model for Semantic Parsing

The training procedure of a baseline seq2seq model for semantic parsing is illustrated in Figure 1. The parsing model maps natural language sentences to target expression. The training procedure of a basic seq2seq parsing model is illustrated in Figure 1.

First, a natural language sentence is preprocessed into a sequence of word indexes $x = \{x_1, \ldots, x_m\}$ and the labeled logical form is preprocessed into a sequence of word indexes $y^* = \{y^*_1, \ldots, y^*_n\}$. Then, the encoder network produces the sequence $x = \{x_1, \ldots, x_m\}$ into a high level contextual representation $h = \{h_1, \ldots, h_m\}$. Later, the decoder network generates the target output $y = \{y_1, \ldots, y_n\}$ from $h$.

The training criterion is cross entropy:

$$L_{LM} = - \sum_{i=1}^{N} \sum_{k=1}^{V} \mathbf{1} \{y_t = k\} \log P_{LM}(y_t = k)$$ (2)

where $P_{PAR}$ is computed from Equation 1, $T$ is the length of the target sequence, $|V|$ is the size of the vocabulary, $\mathbf{1}$ is the indicator function.

3.2 Self-Attention

The multi-head self-attention module is a key component in Transformer (Vaswani et al., 2017). In particular, transformer’s sub-layers employ $h$ attention heads to perform self-attention. The results from each attention heads are concatenated and transformed to form the output of the sub-layer.

Given a sequence $x = (x_1, \ldots, x_n)$ as input, each attention head uses scaled dot-product attention to compute a new sequence $z = (z_1, \ldots, z_n)$ of the same length, i.e.,

$$z_i = \sum_{j=1}^{n} \alpha_{ij} (x_j W^V),$$ (3)

where $W^V$ is a matrix of parameters and $\alpha_{ij}$ are normalized by a softmax function, i.e.,

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{n} \exp(e_{ik})},$$ (4)

where $e_{ij}$ is computed using a compatibility function that compares two input elements, i.e.,

$$e_{ij} = \frac{(x_i W^Q)(x_j W^K)\top}{\sqrt{d_z}},$$ (5)

where $W^Q, W^K$ are parameters to be learned.

4 Method

In this section, we specify details of our method, i.e., using a knowledge distillation based structure to transfer knowledge from a structure-aware target language model to a seq2seq model. We first introduce the architecture of the new model. Then, we describe the proposed target language model. At last, we provide details of the method in the training process.

4.1 Model Overview

An overview of the new model’s architecture is shown in Figure 2. Note that, the new model is
generated from the basic seq2seq model in Figure 1 by introducing a knowledge distillation structure where the pretrained structure-aware language model serves as the teacher to guide the parsing model.

In specific, the structure-aware language model is pre-trained on target logical forms. The language model contains a structure-aware self-attention transformer encoder to explicitly capture the structural information. It is used to provide soft labels as prior knowledge to "teach" the parsing model in the training process, where the Kullback-Leibler divergence between estimated probabilities is intended to be minimized.

Notice that, there is no specific requirement for the seq2seq model in the architecture. Then, besides the basic seq2seq model, this knowledge distillation structure can be applied to other sophisticated seq2seq models to leverage structural features of target representations.

4.2 Target Language Model

Here we specify details of the proposed target language model, i.e., structure-aware self-attention language model. Architecture of the language model is shown in Figure 3.

Since the target logical forms can all be seen as bracket trees, they’re tree-structured. Self attention in Transformer learns how much attention to put on words in a sequence, but it ignores the syntactic information of trees. The siblings of tree nodes may have long distance in a sequence position, but they’re related closely. Therefore, we propose structure-aware self-attention to encode the depth information of sibling nodes into self-attention to capture this information.

Motivated by (Shaw et al., 2018), we extend the self-attention architecture to explicitly encode the relation between an element pair \((x_i, x_j)\) by modifying Equation (5) to

\[
e_{ij} = \frac{x_i W^Q (x_j W^K + a_{ij}^K) \top}{\sqrt{d_z}}.
\]

Different from (Shaw et al., 2018), we redefine the relation representations \(a_{ij}\).

We assume that the depth information is less useful when it is too deep. We define the maximum \(s\) as a constant \(k\):

\[
a_{ij}^K = \text{clip}(s(i,j), k)
\]

\[
\text{clip}(x, k) = \min(x, k)
\]

where \(s(i, j)\) is defined as follows:

\[
s(i, j) = \begin{cases} \text{dep}(i), & \text{father}(i) = \text{father}(j), \\ 0, & \text{otherwise}, \end{cases}
\]

where \(\text{dep}(i)\) is the depth of node \(i\) in a tree, \(\text{father}(i)\) means the father of node \(i\).

Figure 4 shows an example we chose in GEO dataset for demonstration.

We replace the original self-attention architecture of transformer encoder with our structure-aware self-attention. The encoder is bidirectional, so we add the subsequent mask (originally applied in the transformer decoder) to it to specify it as a language model. The subsequent mask creates a lower triangular matrix where the elements above the diagonal will be modified to zero and the elements below the diagonal will be set to whatever the input tensor is. Therefore, the prediction for position \(i\) will depend only on the known outputs at positions less than \(i\).

The generation of the language model is determined by:

\[
P_{LM}(y_t) = p(y_t \mid y_{<t}).
\]

In our experiments, the language model is trained based on λ-calculus expressions and python codes appeared in the training sets of the ATIS, GEO, Django and CoNaLa datasets respectively. The training objective of the language model is to minimize the cross-entropy with target expressions:

\[
L_{LM} = - \sum_{i=1}^{N} \sum_{k=1}^{|V|} 1 \{y_t = k\} \log P_{LM} (y_t = k)
\]

where \(N\) is the length of the target sequence, \(L_{LM}\) denote the training objective functions for the language model, \(P_{LM}\) is computed by Equation (9) respectively.

Given a sequence of preprocessed logic form indexes \(y^* = \{y_0^*,\ldots,y_{n-1}^*\}\) obtained from a labeled logical form \((y_0^*\text{ is the start symbol}, y_n^*\text{ is the end symbol})\), the language model produce likelihoods of the target distribution as soft labels, i.e., it generates \(y^S = \{y_0^S,\ldots,y_n^S\}\).

4.3 Training

In the training process, we need to combine the loss from the seq2seq model, \(L_{PAR}\), and the loss from knowledge distillation, \(L_{KD}\).
In specific, to make the seq2seq model learn the knowledge from the language model, we put target sequences into the language model to get estimated probabilities, then we minimize the Kullback-Leibler (KL) divergence between output of the language model and output of the decoder. The loss from knowledge distillation is:

\[
L_{KD} = - \sum_{i=t}^{T} \sum_{k=1}^{|V|} KL(P_{PAR}(y_t = k), P_{LM}(y_t = k))
\]

where \( P_{LM} \) denotes the output of the language model computed by by Equation (9) and the function \( KL \) computes the KL divergence.

At last, the loss for the entire model is the combination:

\[
L = \eta L_{PAR} + (1 - \eta) L_{KD}
\]

where \( \eta \) is a coefficient between 0 and 1.

# Experiments

In order to evaluate the performance of our proposed model, we conduct the experiments detailed below.

## 5.1 Language Modeling

In this section, we evaluate our structure-aware language model on language modeling. We evaluate the performance on language modeling by measuring the perplexity (PPL) of target sentences. We use 31425 lambda statements and 51877 python statements collected from the github website as the language model dataset, and follows the ratio of 8:1:1 to devide the training set, validation set and the test set. It should be noted that the lambda statements and python statements in all test sets in the semantic parsing datasets are removed from the training dataset to prevent any impact.

We reproduce and test open-source structural language models in recent years, and compare them with proposed structure-aware language model, using perplexity as an evaluation indicator. In order to explore the contribution of the structure-aware self-attention module in the language model to the model, we also conduct an ablation experiment that removes the structure-aware self-attention.

Table 1 shows the result of our structure-aware language model and other structural language models on language modeling. Compared with other
We evaluate our approach on four semantic parsing and code generation benchmarks:

**ATIS** contains natural language questions of a flight dataset paired with a lambda calculus query. We follow the standard train-dev-test split of the datasets in (Zettlemoyer and Collins, 2007), which is 4434/491/448.

**GEO** contains natural language questions about US geography paired with Prolog database queries. We use the corresponding $\lambda$-calculus expressions with the same meaning as in (Kwiatkowski et al., 2011). We follow the standard train-dev-test split of the datasets in (Zettlemoyer and Collins, 2005), which is 600/0/280.

**Django** contains lines of Python source code extracted from the Django framework paired with an NL description. We follow the standard train-dev-test split of the datasets in (Oda et al., 2015), which is 16000/1000/1805.

**CoNaLa** contains manually annotated NL questions paired with python solution on STACKOVERFLOW. We follow the standard train-dev-test split of the datasets in (Yin et al., 2018), which is 2379/0/500.

5.3 Datasets

We first specify details of our implementation including the datasets, the hyperparameters, hardware, and software for training and testing networks. Then we present the experimental results, which show that our model achieves new state-of-the-art performance among various neural semantic parsers on all four datasets.

<table>
<thead>
<tr>
<th>Model</th>
<th>ATIS</th>
<th>GEO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>88.6</td>
<td>88.9</td>
</tr>
<tr>
<td>+ SLM KD fusion</td>
<td>90.4</td>
<td>91.1</td>
</tr>
<tr>
<td>- structure-aware</td>
<td>88.8</td>
<td>89.3</td>
</tr>
</tbody>
</table>

Table 2: Results on ATIS and GEO datasets.
### Table 3: Results on Django dataset.

<table>
<thead>
<tr>
<th>Model</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coarse2fine(Dong and Lapata, 2018)</td>
<td>74.1</td>
</tr>
<tr>
<td>TranX (Yin and Neubig, 2018)</td>
<td>73.7</td>
</tr>
<tr>
<td>TranX2 (Yin and Neubig, 2019)</td>
<td>77.3±0.4</td>
</tr>
<tr>
<td>TranX2+BERT</td>
<td>79.7±0.42</td>
</tr>
<tr>
<td>Reranker (Yin and Neubig, 2019)</td>
<td>80.2±0.4</td>
</tr>
<tr>
<td>TAE (Norouzi et al., 2021)</td>
<td>81.03±0.14</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ours</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>81.03</td>
</tr>
<tr>
<td>+ SLM KD fusion</td>
<td>81.83</td>
</tr>
<tr>
<td>- structure-aware</td>
<td>81.16</td>
</tr>
</tbody>
</table>

### Table 4: Results on CoNaLa dataset.

<table>
<thead>
<tr>
<th>Model</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>TranX (Yin and Neubig, 2018)</td>
<td>24.3</td>
</tr>
<tr>
<td>Reranker (Yin and Neubig, 2019)</td>
<td>30.11±0.6</td>
</tr>
<tr>
<td>EK (Xu et al., 2020)</td>
<td>27.20</td>
</tr>
<tr>
<td>EK+100k (Xu et al., 2020)</td>
<td>28.14</td>
</tr>
<tr>
<td>EK+100K+API (Xu et al., 2020)</td>
<td>32.26</td>
</tr>
<tr>
<td>TAE (Norouzi et al., 2021)</td>
<td>32.57±0.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ours</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>32.57</td>
</tr>
<tr>
<td>+ SLM KD fusion</td>
<td>33.10</td>
</tr>
<tr>
<td>- structure-aware</td>
<td>32.62</td>
</tr>
</tbody>
</table>

5.4 Implementation Details

We use AdaNSP (Zhang et al., 2019), a competitive seq2seq semantic parsing model built on AllenNLP (Wallace et al., 2019), as our base model for two semantic parsing tasks. The model uses adaptive decoding method that guide the decoder by model uncertainty and automatically uses deeper computations when necessary. The AdaNSP model is not the state-of-the-art model now, but it is based on seq2seq architecture and open-sourced so it is easy to implement our method. We adapt the same hyperparameters as in (Zhang et al., 2019). We use TAE (Norouzi et al., 2021), a seq2seq code generation model as our base model for two code generation tasks. The model exploit a relatively sizeable monolingual corpus of the target programming language to a transformer-based seq2seq model and reach a superior performance.

We trained our model with the hyperparameters listed in Table 5, which was chosen based on the performance of the model on the validation set for ATIS, Django and on the randomly selected training set for GEO, CoNaLa, where the validation set is not provided. For structures of the language model, we set the number of layers 3, positional feed forward dimensions 512, and attention heads 8. We trained the parsing model with the original settings of the baseline system. We trained the language model for 100 epochs respectively, and the entire model for 200 epochs on an Nvidia GeForce RTX 3090 GPU, which takes around 5 hours.

5.5 Evaluation

We use logical form accuracy as the evaluation metric for ATIS and GEO datasets, which is computed with pared trees of the predictions and gold logical forms. The order of the children can be changed within conjunction nodes. We use STree parser code from (Dong and Lapata, 2018) to parse the target lambda expressions and predictions into bracket trees and compare them. We use exact match accuracy as the evaluation metric for Django dataset and corpus-level BLEU for CoNaLa.

5.6 Results

We compare our method with state-of-the-art semantic parsers on ATIS, GEO, Django and CoNaLa datasets. Table 2-4 show the results of our model and existing semantic parsers on four datasets. Our model achieves the state-of-the-art performance on four datasets.

We also performed an ablation study by removing the proposed structure-aware self-attention. In specific, we use an original transformer encoder as the language model and integrate it into the parsing model by knowledge distillation. The results show that the model using the structure-aware language model outperforms the one using only original language model.

6 Conclusion

In this paper, we present a structure-aware self-attention language model to capture structural information of target representations and propose a knowledge distillation based approach to incorporating the target language model into a seq2seq model. We show that using knowledge distillation from a target language model provides a flexible and scalable way for neural semantic parsers to leverage structural features of target representations. Our method achieves strong results.
and doesn’t need any manually designed rules or sketches.

For future direction, we are interested in exploring other datasets to verify the model’s ability for structural data. We will also attempt to integrate grammar rules to this model to have a better performance on semantic parsing tasks.

Acknowledgements


References


Chenliang Li, Bin Bi, Ming Yan, Wei Wang, Songfang Huang, Fei Huang, and Luo Si. 2021. Structurallm: Structural pre-training for form understanding. ArXiv, abs/2105.11210.


