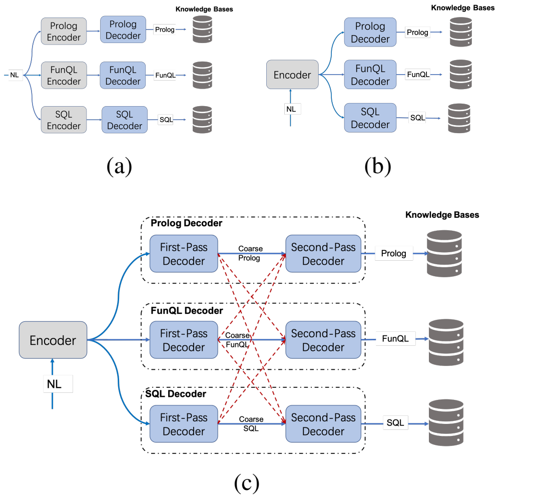
Multi-Target Semantic Parsing with Collaborative Deliberation Network

Xiang Li[1,2], **Fangyu Lei**[1,2], **Shizhu He**[1,2]*, **Kang Liu**[1,2], **Jun Zhao**[1,2],
[1]The Laboratory of Cognition and Decision Intelligence for Complex Systems,
Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences,
Beijing, China
{leifangyu2022, lixiang2022}@ia.ac.cn
{shizhu.he, jzhao, kliu}@nlpr.ia.ac.cn

Abstract

Semantic parsing aims at mapping natural language utterances into machine-interpretable meaning representations, facilitating user accesses to knowledge bases. However, knowledge in real-world scenarios is often duplicated in multiple storages and different representations. Although researchers have made great success by improving neural semantic parsers, existing works can only handle a specific kind of meaning representation, i.e., the single-target semantic parsing. In this paper, we introduce a multi-target semantic parsing model based on a collaborative deliberation network, which can not only decode multiple meaning representations simultaneously but also allow meaning representations to make use of information from each other while decoding. Experiments show that the proposed model improves the EM accuracy of four MRs averagely by 2.48% to 5.05% on three public datasets [1].



(a) (b)

(c)

Figure 1: Different implementations for multi-target semantic parsing (MulTSP). (a) Training one parser for each target. (b) A multi-task parser that independently decodes different targets with a shared encoder. (c) The collaborative deliberation network which exploits other coarse decoding outputs to enhance the decoding of any target.

1 Introduction

Knowledge bases (KBs) (Bollacker et al., 2008; Lehmann et al., 2015; Weikum et al., 2021) are fundamental facilities for many AI applications. The large amount of structural information they contain is mainly accessed by corresponding formal query language. A variety of query languages are adopted due to the diversity of implementations and domains, such as Prolog for expert systems, and the standardized SQL and SPARQL for relational databases and RDF stores respectively. Semantic parsing (Zelle and Mooney, 1996; Dahl et al., 1994) provides a friendly interface to people unfamiliar with these formal languages by transducing the natural language questions into the underlying query language, which has been widely used in abundant tasks, such as question answering system (Lan et al., 2021; Hu et al., 2018) and robot navigation (Walker et al., 2019). Recently, neural semantic parsing with the notable sequence-to-sequence (Seq2Seq) framework has made impressive progress (Jia and Liang, 2016; Yin and Neubig, 2017; Dong and Lapata, 2018; Peter Shaw and Altun, 2019).

However, recent parsers only focus on single-target semantic parsing designed for only one certain KB and its corresponding query language, whereas in real-world scenarios, the same fact is commonly duplicated and stored in many systems. For example, personnel and financial data are stored in both tabular and graph structures in some businesses, which calls for a multi-target semantic parsing (MulTSP) model with the capability of generating multiple meaning representations (MRs).

There are two approaches to adopt existing technologies for MulTSP. Specifically, the first is to

*Corresponding author

[1]Our implementation can be found on https://github.com/Xiang-Li-oss/MultiTarget

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023, pages 211–221
November 1–4, 2023. ©2023 Asian Federation of Natural Language Processing

train N seq2seq models (Dong and Lapata, 2016; Jia and Liang, 2016) for N target formal languages independently, which is far from an efficient way for MulTSP. And the second approach is to resort to the **multi-task learning** for all the target formal languages with different decoders, which only exchange information at the encoder. Figure 1(a) and Figure 1(b) illustrate their differences. As recent work has shown (Guo et al., 2020), there's a significant performance gap between different MRs, for example, the FunQL is more competitive than Lambda Calculus and Prolog. This experimental finding suggests the performance of decoding one MR might get improved with the help of other MRs by introducing collaborations among their decoders.

To implement collaborations among decoders of different MRs, a straightforward way is to augment attention mechanism within tokens from all MRs at each decoding timestep. However, this is less likely to succeed due to two concerns: 1) The erroneous alignment problem, in which the tokens of different MRs could be hardly aligned by position and the attention mechanism could be even harmful to the current decoding step, for example, considering parsing *"How large is Texas ?"* to Lambda Calculus and FunQL in Table 1. At 2nd timestep, current token *'('* in FunQL is most likely to be useless for predicting *'size<lo,i>'* in Lambda Calculus. 2) Holistic information ignorance, in which future tokens from one MR are invisible for other MRs. For example, at the 2nd decoding step, *'size'* in FunQL is valuable yet invisible for *'size<lo,i>'* in Lambda Calculus.

To address the aforementioned problems, we present a multi-target semantic parsing model based on a collaborative deliberation network(CDNParser). As shown in Figure 1(c), each decoder in CDNParser performs two-stage decoding through its two child decoders. During the first decoding stage, each first-pass decoder generates a coarse target formal language sequence for its MR without teach forcing. After that, each second-pass decoder will predict final target sequences as well as collaborate with each other by sharing sequences of coarse target logical forms, which are almost complete target sequences for each MR. Through this design, our method meets the requirements of MulTSP and avoids the problem of erroneous alignments and holistic information ignorance.

To evaluate the proposed model in this paper, we have conducted extensive experiments on three public benchmarks. Compared with strong baselines, the experimental results demonstrate that the proposed CDNParser can 1) successfully meet the need for MulTSP, 2) for the EM (exact-match) accuracy of four MRs, CDNParser provides 2.52% average improvement on Geo dataset, 5.05% on Job dataset, and 2.48% on ATIS dataset.

To sum up, our contributions are as follows:

1. We propose to perform multi-target semantic parsing, which supports simultaneous decoding of multiple meaning representations. To our best knowledge, this is the first work for MulTSP task.

2. We propose a MulTSP model by designing a collaborative deliberation network (CDNParser) which allows the decoding process of each MR to utilize the assistance of other MRs.

3. We have conducted detailed experiments on several open datasets. And the experimental results demonstrate that CDNParser can not only decode multiple MRs simultaneously but also achieves better performance.

2 Related Work

2.1 Semantic Parsing Methods

Various semantic parsing models have been proposed over the past decades, which devotes to converting natural language utterances into logical forms that can be easily executed on a knowledge graph (Kwiatkowski et al., 2011; Andreas et al., 2013; Zhao and Huang, 2015). These systems typically learn lexicalized mapping rules and scoring models to construct a meaning representation for a given input.

Table 1: Examples of MRs for utterance "How large is Texas?" in the Geo domain.

MR	Logical Form
Prolog	answer(A, (size(B, A), const(B, stateid(texas))))
Lambda Calculus	(size:<lo,i> texas:s)
FunQL	answer(size(stateid('texas')))
SQL	select STATEalias0.AREA from ... where ... STATE_NAME = "texas"

With the development of neural networks, sequence-to-sequence models have been employed to semantic parsing with remarkable results. The neural-based parsing models avoid the requirement for feature engineering and explicit semantic composition rules (Dong and Lapata, 2016; Jia and Liang, 2016; Ling et al., 2016) in previous methods. Lots of ideas have been explored to improve the performance of these models, including data augmentation (Wu et al., 2021) and designing structural decoders (Dong and Lapata, 2018; Li et al., 2021). Although there have been some efforts to use share parameters for multiple languages or meaning representations, there hasn't been any attempts to study multi-target semantic parsing, which is the main focus of this paper.

2.2 Meaning Representations

We adopt four formal languages (that is Meaning Representations, MRs) in this paper: Prolog, Lambda Calculus, FunQL, and SQL. Since these MRs are explored extensively in semantic parsing task, making it easier to obtain labeled data in these MRs. Table 1 shows a logical form for each of the four MRs in the Geo domain.

Prolog is a logic programming language for deductive reasoning. Since its birth, it has been extended to many application areas, such as relational databases, mathematical logic, natural language understanding, etc. Prolog is based on first-order predicates logic, incorporating higher-order predicates to handle problems such as quantification and aggregation.

Lambda calculus is a formal system for describing operations, which can represent all first-order predicates, and can support some higher-level predicates. Lambda calculus expresses natural language through constants, quantifiers, logical connectives, Lambda operators, etc. These advantages promote the widespread use of Lambda calculus in semantic parsing tasks.

FunQL is a variable-free language that abstracts variables away and uses nested structures to represent compositionality. Unlike languages such as Prolog and Lambda calculus, predicates in FunQL take a set of entities as input and return another set of entities as output.

SQL is short for Structured Query Language and is a popular relational database query language. Given that SQL is domain-agnostic and has well-established execution engines, the subtask of semantic parsing, Text-to-SQL has also sparked a lot of attention.

2.3 Deliberation Network

Deliberation network (Xia et al., 2017) is inspired by human behavior. For example, to write a good document, we usually first create a complete draft and then polish it based on a global understanding of the whole draft. Deliberation network leverages the global information with both looking back and forward in sequence decoding through a deliberation process. (Li et al., 2019a) and (Xiong et al., 2019) apply deliberation network in dialogue system and machine translation and and adopt a new simplified training algorithm instead of the original Monte Carlo based algorithm. In our work, deliberation network acts as a key technology to implement collaborative multi-target decoding among different MRs.

3 Proposed Method

Given a dataset $D = \{(x, y^1, ..., y^N)\}$ where $x = \{x_1, x_2, ...x_{T_x}\}$ is the source natural language sequence, for example *"How large is Texas ?"*, and each $y^i = \{y^i_1, y^i_2, ...y^i_{T_{y^i}}\}$ is a target logical form sequence for one MR, such as FunQL sequence *"answer(size(stateid('texas')))"* or Prolog sequence *"answer(A,(size(B,A),const(B, stateid(texas))))"*. T_x. and T_{y^i} represent sequence length. Our model is aimed at learning a parser which transforms x into $(y^1, ..., y^N)$, simultaneously.

3.1 Model Overview

Figure 2 visualizes the overall architecture of our multi-target semantic parsing model. Our model is based on a collaborative deliberation network. A collaborative deliberation network consists of an encoder E and N (N=2 in this example) deliberation decoders $D^i (i = 1, 2, ..., N)$. In this example, two deliberation decoders are responsible for generating Lambda Calculus and FunQL, respectively. Each deliberation decoder is composed of a first-pass decoder D^i_1 and a second-pass decoder D^i_2, where collaboration happens. Briefly speaking, E is used to encode the source sequence into a sequence of vector representations. Each D^i_1 reads the encoder representations and generates a first-pass target sequence as a draft, which further participates in the collaboration mechanism and is provided as input to the second-pass decoders to

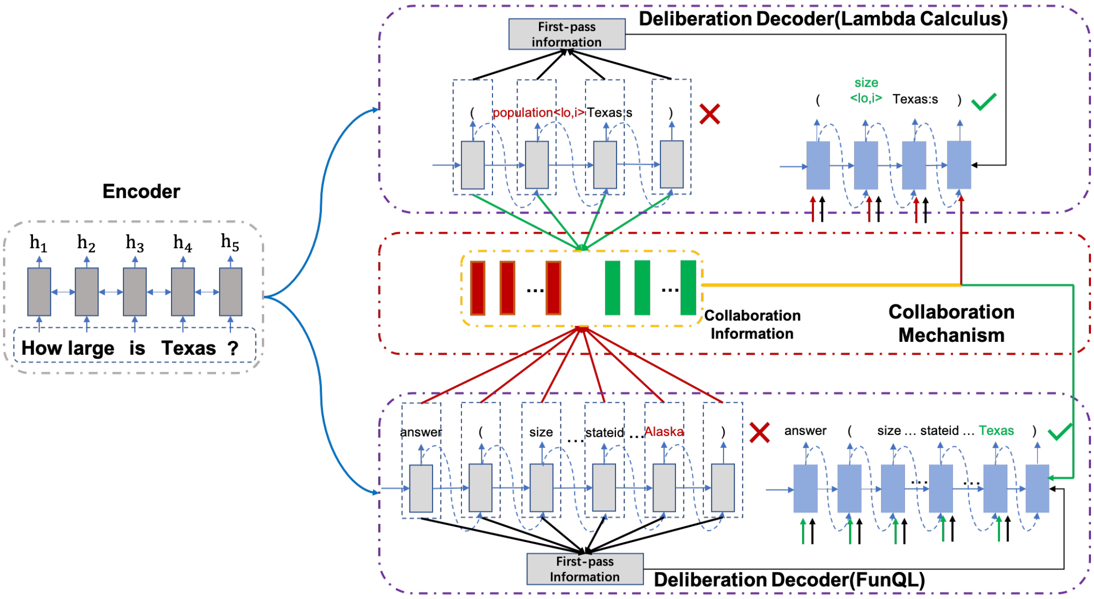


Figure 2: The overview of the collaborative deliberation network for MulTSP. For simplicity, we omit the encoder-decoder attention. The encoder on the left encodes natural language utterances before feeding them to two deliberation decoders. Each decoder performs two-phase decoding to generate Lambda Calculus and FunQL respectively. Moreover, during the second decoding phase, the two decoders are supposed to collaborate.

perform second pass decoding.

3.2 Encoder and First-Pass Decoder

When an source sequence x is fed into the encoder E, it is encoded into a hidden vector sequence $\{h_1, h_2, ..., h_{T_x}\}$ with the length of T_x. To be specific, $h_i = \text{LSTM}(x_i, h_{i-1})$, where x_i is the ith word embedding and h_0 is intialized to zero vector. For the sake of convenience, in the following part of this section, we will omit the superscript of y^i, D_1^i, and D_2^i and denote them as y, D_1, and D_2.

Each first-pass decoder D_1 will generate a sequence of hidden states $\hat{s}_j (\forall\, j \in [T_{\hat{y}}])$, and the first-pass target logical form sequence $\hat{y}_j (\forall\, j \in [T_{\hat{y}}])$. Next we explain the process in detail.

Identificial to the conventional encoder-decoder model, an additive attention is included in each D_1. At timestep t, D_1 first calculate contextual vector ctx_e as follows:

$$\alpha_i = \frac{\exp(v_e^T \tanh(W_0^e h_i + W_1^e \hat{s}_{t-1}))}{\sum_{k=1}^{T_x} \exp(v_e^T \tanh(W_0^e h_k + W_1^e \hat{s}_{t-1}))}$$

$$ctx_e = \sum_{i=1}^{T_x} \alpha_i h_i$$

where v_e^T, W_0^e, W_1^e are parameters.

After obtaining ctx_e, the hidden state $\hat{s}_i^t$ of first-pass decoder is got by:

$$\hat{s}_t = \text{LSTM}([\hat{y}_{t-1}; ctx_e], \hat{s}_{t-1})$$

After that, the concatenated vector $[\hat{s}_t, ctx_e, \hat{y}_{t-1}]$ will be fed into a feedforward network to predict the probability of $\hat{y}_t$ as:

$$p(\hat{y}_t | x, \hat{y}_{<t}) = \text{softmax}(\text{FFN}([\hat{s}_t, ctx_e, \hat{y}_{t-1}]))$$

Finally, $\hat{y}_t$ is predicted by:

$$\hat{y}_t = \text{argmax}\, p(\hat{y}_t | x, \hat{y}_{<t})$$

3.3 Second-Pass Decoder

The second-pass decoder is the key component of our approach since it contains the vital collaboration mechanism. Figure 2 exhibits the behavior of second-pass decoder.

Once first-pass decoder has generated the first-pass target logical form sequences $\hat{y}$ as a draft. It is fed into the second-pass decoder for further collaboration. Besides the sequence $\hat{y}$ and hidden state $\hat{s}$ produced by its corresponding first-pass decoder, the second-pass decoder also collaborates with other second-pass decoders, taking outputs from other first-pass decoders into consideration.

First, the information from source sequence is integrated by calculating contextual vector ctx'_e, the computation detail is similar to ctx_e in Section 3.2.

To synthesize information from the first-pass target logical form sequence, we compute the contextual vector ctx_c by:

$$\alpha_i = \frac{\exp(v_c^T \tanh(W_0^c[\hat{s}_i; \hat{y}_i] + W_1^c s_{t-1}))}{\sum\limits_{j=1}^{T_y} \exp(v_c^T \tanh(W_0^c[\hat{s}_j; \hat{y}_j] + W_1^c s_{t-1}))}$$

$$ctx_c = \sum_{i=1}^{T_y} \alpha_i [\hat{s}_i; \hat{y}_i]$$

where v_c^T, W_0^c, W_1^c are parameters.

Second-pass decoder calculates contextual vector ctx_o to collaborate with other deliberation decoders based on $\hat{y}^o$ and $\hat{s}^o$, where $\hat{y}^o = [y^j \; \forall j = 1, 2, ..., N \wedge j \neq i]$ represents the concatenation of first-pass target logical form sequences predicted by other deliberation decoders. Simmilar for $\hat{s}^o$ and ctx_o is defined as follow:

$$\alpha_i = \frac{\exp(v_o^T \tanh(W_0^o[\hat{s}_i^o; \hat{y}_i^o] + W_1^o s_{t-1}))}{\sum\limits_{j=1}^{T_y} \exp(v_o^T \tanh(W_0^o[\hat{s}_j^o; \hat{y}_j^o] + W_1^o s_{t-1}))}$$

$$ctx_o = \sum_{i=1}^{T_y} \alpha_i [\hat{s}_i^o; \hat{y}_i^o]$$

where v_o^T, W_0^o, W_1^o are parameters.

After obtaining all of the three contextual vectors, D_2 takes the previous hidden state s_{t-1} generated by itself, previously decoded token y_{t-1} to calculate s_t as:

$$s_t = \mathrm{LSTM}([y_{t-1}; ctx_e'; ctx_c], s_{t-1})$$

Finally, the concantenated vector:

$$a_t = [s_t; ctx_e'; ctx_c; ctx_o; y_{t-1}]$$

is fed to a feedforward network followed by a softmax layer to predict the probability of y_t. As can be seen from the above computation, the contextual vector ctx_o aggregates the global information extracted from the first-pass logical form sequence of other MRs, making the most use of assistance from other MRs.

3.4 Training

Since our model output multiple MRs, we minimize summed training for all MRs:

$$L = \sum_{i=1}^{N} L_i$$

each L_i represents a loss for a certain MR. Next, we explain how L_i is calculated in detail.

In contrast to the original deliberation network (Gu et al., 2019), where they propose a complex training algorithm based on Monte Carlo Method, we follow the works of Xiong et al. (2019) and Li et al. (2019b), maximizing the summed likelihood of the two generated logical form sequences. And L_i is computed as:

$$\begin{aligned} L_i &= L_i^1 + L_i^2 \\ L_i^1 &= -\textstyle\sum_D \sum_{j=1}^{T_{y^i}} \log P(\hat{y}_j^i) \\ L_i^2 &= -\textstyle\sum_D \sum_{j=1}^{T_{y^i}} \log P(y_j^i) \end{aligned}$$

where $P(\hat{y}_j^i)$ is the probability predicted for jth token in ith MR by first-pass decoder and $P(y_j^i)$ is the probability predicted for jth token in ith MR by second-pass decoder.

4 Experiment

4.1 Datasets

We trained our model on the three datasets covering different domains: Geo, ATIS, and Job, each of which has been widely used in previous works and has been labeled with logical forms for four MRs including Prolog, Lambda Calculus, FunQL and SQL.

Geo focuses on querying a US geography database, originally containing 880 (utterance, logical form) pairs annotated by (Zelle and Mooney, 1996) through a Prolog-style MR. Latter, (Popescu et al., 2003) and (Rohit J. Kate and Mooney, 2005) proposed to use SQL and FunQL to represent the meanings, respectively. Furthermore, (Zettlemoyer and Collins, 2005) proposed to use Lambda Calculus and manually converted the Prolog logical forms to equivalent expressions in Lambda Calculus. By convention, we adopt the standard 600/280 training/test split.

ATIS is a standard semantic parsing dataset about flight booking, consisting of 5410 questions and their corresponding SQL queries. (Zettlemoyer and Collins, 2007) use Lambda Calculus to represent the meanings and automatically map SQL queries to equivalent Lambda Calculus. The standard split has 4480 training instances, 480 development instances and 450 test instances.

Job dataset contains 640 utterances and their corresponding Prolog queries to a dataset of job listings. Similar to Geo, (Zettlemoyer and Collins, 2005) proposed to represent the meanings with Lambda Calculus and manually converted Prolog logical forms to equivalent Lambda Calculus logical forms.

Since not all the four MRs introduced above are used in these three domains, following previous works, (Guo et al., 2020) has annotated the above three datasets with all the four MRs by semi-automatically translating logical forms in one MR into others.

4.2 Baselines

Our models are compared with two baselines:

1. Seq2Seq model: We formulate semantic parsing task as a neural machine translation task and employ the encoder-decoder architecture to solve it. To predict N MRs, we need N independent Seq2Seq models to generate logical forms for each MR. Each Seq2Seq model contains a one-layer BI-LSTM (Gers et al., 2000) encoder and a one-layer LSTM decoder.

2. Multi-task learning model: Applying multi-task learning to parse multiple MRs. It consists of a one-layer BI-LSTM encoder and multiple one-layer LSTM decoders. Each decoder is responsible for generating logical forms for one MR. The decoders only share the same encoder and have no communication with each other. The crucial distinction between this model and our model is whether to introduce the help of other MRs.

4.3 Implementations

We adopt AllenNLP (Gardner et al., 2018) and Pytorch (Paszke et al., 2019) framework to implement each approach. Due to the limited number of test data in each dataset, we run each approach five times and report the mean performance and standard deviation.

Many neural semantic parsing approaches adopt data anonymization to replace entities in utterances with placeholders. However, in this paper, we do not apply data anonymization to avoid bias.

For parameters, we tune the hyper-parameters of each approach on the development set or through cross-validation on the training set. We put the search space of hyper-parameters in Appendix A.

We initialize all parameters uniformly within the interval [-0.1, 0.1]. We use Adam algorithm to update parameters. Gradients are clipped to 5.0 to alleviate the gradient explosion problem. Early stop is used to determine the number of training epochs. We use greedy search to generate logical forms during inference.

4.4 Evaluation Metrics

Following previous work (Guo et al., 2020), we use EM (exact-match) accuracy as the evaluation metric, which is defined as the percentage of the samples that are correctly parsed to their gold standard meaning representations.

4.5 Experimental Results

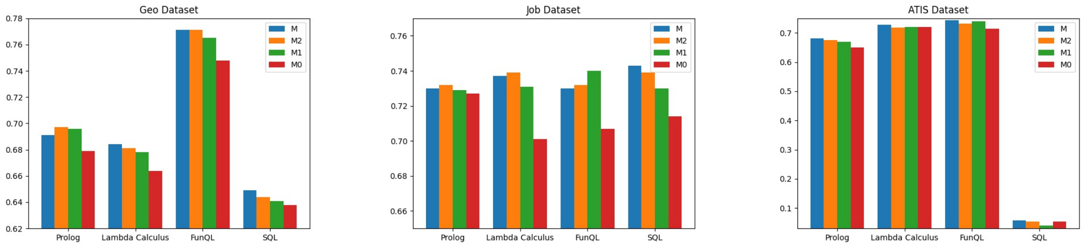
At first, we evaluate our model's performance on three standard benchmarks along with baseline models, in performing the MulTSP (multi-target semantic parsing) task. All approaches are required to predict four MRs, namely Prolog, Lambda Calculus, FunQL, and SQL.

Table 2 shows our results. By comparing various methods, we observe that 1) our model consistently outperforms all other candidates. On the one hand, our model achieves the best parsing accuracy among all methods, on the other hand, the performance of our model is more steady, making our method more reliable. For example, on Job dataset, CDNParser achieves 5.05% absolute improvement on average compared to the Seq2Seq model with a lower standard deviation. This justifies the effectiveness of our approach on MulTSP task. 2) Both our model and multi-task learning model can produce all four MRs. However, our model beats the multi-task learning model substantially, indicating that incorporating help from other MRs plays a crucial role in enhancing accuracy.

By comparing the performance of various MRs, we observe that 1) FunQL tends to outperform other MRs and our model's advancement on FunQL is relatively lesser as compared to the other three MRs. One possible reason is that FunQL is more compact than the other MRs due to its elimination of variables and quantifiers, making it easier to predict even without the aid of other MRs. 2) SQL appears to have the poorest performance yet tends to achieve the highest improvement. For example, in Geo dataset, the exact-match accuracy of FunQL

Table 2: Experimental results of our multi-target semantic parsing model and two baseline models. Seq2Seq* reports the results from (Guo et al., 2020). The results are averaged over 5 random seeds (mean±std). We highlight the best performance for every MR and our model's absolute improvement over Seq2Seq*. The poor SQL exact-match accuracy in ATIS is caused by the different distribution of SQL queries in test set.

Approach	**Prolog**	**Lambda Calculus**	**FunQL**	**SQL**
GEO				
Seq2Seq*	**70.0**±2.1	64.7±2.3	76.8±2.4	58.0±2.4
Seq2Seq	66.8±2.9	65.7±2.6	73.5±2.4	57.8±8.0
Multi-Task	67.9±0.6	66.4±1.0	74.8±2.0	63.8±1.8
CDNParser(ours)	69.1(-0.9)±0.7	**68.4**(+3.8)±1.7	**77.1**(+0.3)±1.3	**64.9**(+6.9)±0.8
JOB				
Seq2Seq*	68.1±2.3	65.8±1.3	71.4±2.4	68.5±3.0
Seq2Seq	67.8±2.3	65.2±1.5	71.6±2.6	67.9±3.4
Multi-Task	72.7±1.6	70.1±2.2	70.7±2.1	71.4±1.8
CDNParser(ours)	**73.0**(+4.9)±2.4	**73.7**(+7.9)±1.1	**73.0**(+1.6)±1.6	**74.3**(+5.8)±2.0
ATIS				
Seq2Seq*	65.9±1.5	68.7±1.8	70.8±0.6	5.6±0.3
Seq2Seq	65.1±1.6	69.0±1.6	70.0±0.8	5.6±0.3
Multi-Task	65.0±2.7	72.0±1.7	71.4±3.6	5.4±0.4
CDNParser(ours)	**68.2**(+2.3)±0.7	**72.7**(+4.0)±0.5	**74.3**(+3.5)±1.0	**5.7**(+0.1)±0.3



(a) Geo

(b) Job

(c) ATIS

Figure 3: Experimental results for ablation studies on three datasets. For each MR, four bars in different colors stand for four models with varying degrees of collaboration. As the collaboration weakens from left to right, the EM accuracy of each MR typically displays a declining trend.

is substantially higher than that of SQL in all approaches. However, SQL gets a remarkable 6.9% improvement from our model. This is understandable, SQL is a domain-general language, unlike domain-specific MRs (Prolog, Lambda Calculus, and FunQL), the domain knowledge can not be injected into SQL easily, making it a challenge for neural networks to generate SQL. Therefore, by obtaining assistance from the other three domain-specific MRs, SQL achieves the maximum promotion.

4.6 Ablation Study

We conduct ablation study to explore the effects of collaboration mechanism in our model. We assess the performance of several collaboration weakened models. We denote M as our original multi-target semantic parsing model, and M_2 as collaborative weakened model, in which each decoder is limited to randomly collaborate with other two out of three decoders. Similarly, each decoder in M_1 can only collaborate with one aonther random decoder and decoders in M_0 are not allowed to collaborate with each other, Figure 3 summarizes the experimental results.

From the results, we observe that: 1) As the collaboration weakens, the EM accuracy of each MR generally shows a downward trend. 2) When collaboration is completely deactivated, there is often a sudden drop in performance. This implies that the collaboration mechanism is a vital component in our method and the outstanding performance achieved by our model benefits a lot from the mutual assistance among MRs brought by collaboration mechanism.

4.7 Analysis

Our main experiment has shown that incorporating information from other MRs can improve semantic parsing performance. We call them auxiliary MRs. This subsection will dive deeper into this subject. We hypothesize that auxiliary MRs contribute differently and fewer MRs could even give better performance according to the results from Figure 3. This brings up an interesting question: *In the case of generating logical forms for a specific MR, how do we choose other auxiliary MRs?*

Table 3: Empirical Transformation Accuracy, row and column respectively represents the source and target MRs.

	Prolog	Lambda Calculus	FunQL	SQL
Prolog	–	70.4%	82.3%	71.2%
Lambda Calculus	59.4%	–	73.3%	69.3%
FunQL	71.1%	72.2%	–	75.8%
SQL	54.0%	57.0%	60.6%	–

We hypothesize that an MR will be enhanced more by a homogeneous MR than a heterogeneous one. For example, the Prolog is intuitively closer to lambda calculus than SQL. A comprehensive measurement of the similarity is not trivial, and thus we resort to the approximated similarity based on empirical results. Specifically, we employ a Seq2Seq neural network to transform an MR into another one. The 4-by-4 transformation results form a matrix (Table 3). We propose a **Heuristic Selection Strategy** (HSS) based on this matrix: For any MR in demand as the target, the source MRs with higher transformation accuracy are more preferred. For example, taking Prolog as the target, the most similar MR is the FunQL (71.1%), followed by Lambda Calculus (59.4%) and SQL (54.0%). Note the similarity is not symmetric in this setting. When the amount of auxiliary MRs is restricted to 1 (or 2), HSS will choose only FunQL (or Lambda Calculus in addition).

To evaluate HSS, we respectively pick up 2 and 1 auxiliary MRs for each MR, there are 3 choices in both cases as marked by letters in parentheses in Table 4 (or Table 5). P, L, F, and S represent Prolog, Lambda Calculus, FunQL, and SQL is picked respectively. The "↑" indicates the higher EM accuracy than taking all the other 3 MRs as auxiliary MRs, while the "↓" indicates the lower EM accuracy.

Table 4: EM accuracy for each MR when picking up 2 auxiliary MRs.We highlight HSS selection for each MR.

Target MR	Composition		
GEO			
Prolog	**69.6(LF)**↑	68.8(LS)↓	68.0(FS)↓
Lambda	**69.9(PF)**↑	67.9(PS)↓	67.4(FS)↓
FunQL	**77.0(PL)**↓	75.7(PS)↓	75.5(LS)↓
SQL	65.0(PL)↑	**65.1(PF)**↑	64.0(LF)↑
JOB			
Prolog	**74.1(LF)**↑	73.3(LS)↑	73.6(FS)↑
Lambda	**74.8(PF)**↑	74.4(PS)↑	73.8(FS)↑
FunQL	**72.9(PL)**↓	73.8(PS)↑	74.4(LS)↑
SQL	73.2(PL)↓	**74.3(PF)**↑	73.5(LF)↓

Table 5: EM accuracy for each MR when picking up 1 auxiliary MR. We highlight HSS selection for each MR.

Target MR	Composition		
GEO			
Prolog	70.4(L)↑	**72.0(F)**↑	66.3(S)↓
Lambda	71.6(P)↑	**70.0(F)**↑	69.0(S)↑
FunQL	**78.0(P)**↑	77.3(L)↑	78.8(S)↑
SQL	64.9(P)↓	65.2(L)↑	**65.0(F)**↑
JOB			
Prolog	74.3(L)↑	**74.1(F)**↑	72.7(S)↓
Lambda	74.7(P)↑	**75.3(F)**↑	75.6(S)↑
FunQL	**71.4(P)**↓	73.7(L)↑	73.6(S)↑
SQL	73.5(P)↑	73.7(L)↑	**75.0(F)**↑

Table 4 and Table 5 shows that: 1) After choosing one or two auxiliary MRs, the EM accuracy for each MR may either rise or fall, confirming our assumption that auxiliary MRs do not contribute equally. 2) After picking one or two auxiliary MRs following our strategy, the EM accuracy for most MRs increases, proving that our strategy based on heuristic features is empirically reasonable and valuable in practice.

5 Conclusion

In this paper, we propose a multi-target semantic parsing model based on a collaborative deliberation network. On the one hand, our model is capable of predicting multiple MRs simultaneously, successfully completing the multi-target semantic parsing task, on the other hand, each MR in our model can leverage assistance information from other MRs thanks to the design of collaboration mechanism in our model. We believe that our method has good application prospects in real-world scenarios due to the diversity of knowledge forms.

Limitations

Our multi-target semantic parsing model based on collaborative deliberation network focues on specific domain, and is unable to handle multi-domain settings. In the future, we plan to extend our method to adapt multi-domain scenarios.

Acknowledgement

This work was supported by National Key R&D Program of China (No.2022ZD0118501) and the National Natural Science Foundation of China (No.62376270, No.U1936207, No.61976211). This work was supported by the Youth Innovation Promotion Association CAS, Yunnan Provincial Major Science and Technology Special Plan Projects (No.202202AD080004) and OPPO Research Fund.

References

Jacob Andreas, Andreas Vlachos, and Stephen Clark. 2013. Semantic parsing as machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 47–52.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pages 1247–1250.

Deborah A. Dahl, Madeleine Bates, Michael Brown, William Fisher, Kate Hunicke-Smith, David Pallett, Christine Pao, Alexander Rudnicky, and Elizabeth Shriberg. 1994. 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*.

Li Dong and Mirella Lapata. 2016. Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33–43, Berlin, Germany. Association for Computational Linguistics.

Li Dong and Mirella Lapata. 2018. Coarse-to-fine decoding for neural semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 731–742, Melbourne, Australia. Association for Computational Linguistics.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. 2000. Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471.

Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. *Advances in Neural Information Processing Systems*, 32.

Jiaqi Guo, Qian Liu, Jian-Guang Lou, Zhenwen Li, Xueqing Liu, Tao Xie, and Ting Liu. 2020. Benchmarking meaning representations in neural semantic parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1520–1540, Online. Association for Computational Linguistics.

Sen Hu, Lei Zou, and Xinbo Zhang. 2018. A state-transition framework to answer complex questions over knowledge base. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2098–2108, Brussels, Belgium. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2016. Data recombination for neural semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 12–22.

Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2011. Lexical generalization in ccg grammar induction for semantic parsing. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1512–1523.

Yunshi Lan, Gaole He, Jinhao Jiang, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. A survey on complex knowledge base question answering: Methods, challenges and solutions. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195.

Yuntao Li, Bei Chen, Qian Liu, Yan Gao, Jian-Guang Lou, Yan Zhang, and Dongmei Zhang. 2021. Keep the structure: A latent shift-reduce parser for semantic parsing. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3864–3870. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Zekang Li, Cheng Niu, Fandong Meng, Yang Feng, Qian Li, and Jie Zhou. 2019a. Incremental transformer with deliberation decoder for document grounded conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 12–21, Florence, Italy. Association for Computational Linguistics.

Zekang Li, Cheng Niu, Fandong Meng, Yang Feng, Qian Li, and Jie Zhou. 2019b. Incremental transformer with deliberation decoder for document grounded conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 12–21.

Wang Ling, Phil Blunsom, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, Fumin Wang, and Andrew Senior. 2016. Latent predictor networks for code generation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 599–609.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Angelica Chen Francesco Piccinno Peter Shaw, Philip Massey and Yasemin Altun. 2019. Generating logical forms from graph representations of text and entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 95–106.

Ana-Maria Popescu, Oren Etzioni, and Henry Kautz. 2003. Towards a theory of natural language interfaces to databases. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pages 149–157.

Yuk Wah Wong Rohit J. Kate and Raymond J. Mooney. 2005. Learning to transform natural to formal languages. pages 1062–1068.

Nick Walker, Yu-Tang Peng, and Maya Cakmak. 2019. Neural semantic parsing with anonymization for command understanding in general-purpose service robots. In *Robot World Cup*, pages 337–350. Springer.

Gerhard Weikum, Luna Dong, Simon Razniewski, and Fabian M. Suchanek. 2021. Machine knowledge: Creation and curation of comprehensive knowledge bases. *Found. Trends Databases*, 10:108–490.

Kun Wu, Lijie Wang, Zhenghua Li, Ao Zhang, Xinyan Xiao, Hua Wu, Min Zhang, and Haifeng Wang. 2021. Data augmentation with hierarchical sql-to-question generation for cross-domain text-to-sql parsing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8974–8983.

Yingce Xia, Fei Tian, Lijun Wu, Jianxin Lin, Tao Qin, Nenghai Yu, and Tie-Yan Liu. 2017. Deliberation networks: Sequence generation beyond one-pass decoding. *Advances in neural information processing systems*, 30.

Hao Xiong, Zhongjun He, Hua Wu, and Haifeng Wang. 2019. Modeling coherence for discourse neural machine translation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7338–7345.

Pengcheng Yin and Graham Neubig. 2017. A syntactic neural model for general-purpose code generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 440–450.

John M. Zelle and Raymond J. Mooney. 1996. Learning to parse database queries using inductive logic programming. In *Proceedings of the 13th National Conference on Artificial Intelligence*, pages 1050–1055.

Luke Zettlemoyer and Michael Collins. 2007. Online learning of relaxed ccg grammars for parsing to logical form. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 678–687.

Luke S. Zettlemoyer and Michael Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, page 658–666.

Kai Zhao and Liang Huang. 2015. Type-driven incremental semantic parsing with polymorphism. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1416–1421.

A Appedix

A.1 Preprocess

According to (Jia and Liang, 2016), we preprocess the Prolog logical forms to the De Brujin index notation. We standardize the variable naming of Lambda Calculus based on their occurrence order in logical forms.

A.2 Hyper-Parameters

For our model, a one-layer bi-directional LSTM is chosen for encoder and a one-layer LSTM is selected as decoder. The embedding dimension of both source and target languages ranges over {100, 200}. The decoder keeps the same hidden dimension as encoder, ranging over {32, 64}. As for the attention model, we apply additive attention. We employ dropout at training time to alleviate over-fitting with rate ranging over {0.1, 0.2, 0.3}. We select batch size from {16, 32, 64}, and select learning rate from {0.005, 0.01}.