

Semantic Parsing with Dual Learning

Ruisheng Cao*, Su Zhu*, Chen Liu, Jieyu Li and Kai Yu[†]

MoE Key Lab of Artificial Intelligence

SpeechLab, Department of Computer Science and Engineering

Shanghai Jiao Tong University, Shanghai, China

{211314, paul2204, chris-chen, oracion, kai.yu}@sjtu.edu.cn

Abstract

Semantic parsing converts natural language queries into structured logical forms. The paucity of annotated training samples is a fundamental challenge in this field. In this work, we develop a semantic parsing framework with the dual learning algorithm, which enables a semantic parser to make full use of data (labeled and even unlabeled) through a dual-learning game. This game between a primal model (semantic parsing) and a dual model (logical form to query) forces them to regularize each other, and can achieve feedback signals from some prior-knowledge. By utilizing the prior-knowledge of logical form structures, we propose a novel reward signal at the surface and semantic levels which tends to generate complete and reasonable logical forms. Experimental results show that our approach achieves new state-of-the-art performance on ATIS dataset and gets competitive performance on OVERNIGHT dataset.

1 Introduction

Semantic parsing is the task of mapping a natural language query into a logical form (Zelle and Mooney, 1996; Wong and Mooney, 2007; Zettlemoyer and Collins, 2007; Lu et al., 2008; Zettlemoyer and Collins, 2005). A logical form is one type of meaning representation understood by computers, which usually can be executed by an executor to obtain the answers.

The successful application of recurrent neural networks (RNN) in a variety of NLP tasks (Bahdanau et al., 2014; Sutskever et al., 2014; Vinyals et al., 2015) has provided strong impetus to treat semantic parsing as a sequence-to-sequence (Seq2seq) problem (Jia and Liang, 2016; Dong and Lapata, 2016). This approach generates

a logical form based on the input query in an end-to-end manner but still leaves two main issues: (1) lack of labeled data and (2) constrained decoding.

Firstly, semantic parsing relies on sufficient labeled data. However, data annotation of semantic parsing is a labor-intensive and time-consuming task. Especially, the logical form is unfriendly for human annotation.

Secondly, unlike natural language sentences, a logical form is strictly structured. For example, the lambda expression of “show flight from ci0 to ci1” is $(\lambda \$0 e (\text{and} (\text{from } \$0 \text{ ci0}) (\text{to } \$0 \text{ ci1}) (\text{flight } \$0)))$. If we make no constraint on decoding, the generated logical form may be invalid or incomplete at *surface* and *semantic* levels.

Surface The generated sequence should be structured as a complete logical form. For example, left and right parentheses should be matched to force the generated sequence to be a valid tree.

Semantic Although the generated sequence is a legal logical form at surface level, it may be meaningless or semantically ill-formed. For example, the predefined binary predicate `from` takes no more than two arguments. The first argument must represent a `flight` and the second argument should be a `city`.

To avoid producing incomplete or semantically ill-formed logical forms, the output space must be constrained.

In this paper, we introduce a semantic parsing framework (see Figure 1) by incorporating dual learning (He et al., 2016) to tackle the problems mentioned above. In this framework, we have a primal task (query to logical form) and a dual task (logical form to query). They can form a closed loop, and generate informative feedback signals to

*Ruisheng Cao and Su Zhu are co-first authors and contribute equally to this work.

[†]The corresponding author is Kai Yu.

train the primal and dual models even without supervision. In this loop, the primal and dual models restrict or regularize each other by generating intermediate output in one model and then checking it in the other. Actually, it can be viewed as a method of data augmentation. Thus it can leverage unlabeled data (queries or synthesized logical forms) in a more effective way which helps alleviate the problem of lack of annotated data.

In the dual learning framework, the primal and dual models are represented as two agents and they teach each other through a reinforcement learning process. To force the generated logical form complete and well-formed, we newly propose a *validity reward* by checking the output of the primal model at the surface and semantic levels.

We evaluate our approach on two standard datasets: ATIS and OVERNIGHT. The results show that our method can obtain significant improvements over strong baselines on both datasets with fully labeled data, and even outperforms state-of-the-art results on ATIS. With additional logical forms synthesized from rules or templates, our method is competitive with state-of-the-art systems on OVERNIGHT. Furthermore, our method is compatible with various semantic parsing models. We also conduct extensive experiments to further investigate our framework in semi-supervised settings, trying to figure out why it works.

The main contributions of this paper are summarized as follows:

- An innovative semantic parsing framework based on dual learning is introduced, which can fully exploit data (labeled or unlabeled) and incorporate various prior-knowledge as feedback signals. We are the first to introduce dual learning in semantic parsing to the best of our knowledge.
- We further propose a novel validity reward focusing on the surface and semantics of logical forms, which is a feedback signal indicating whether the generated logical form is well-formed. It involves the prior-knowledge about structures of logical forms predefined in a domain.
- We conduct extensive experiments on ATIS and OVERNIGHT benchmarks. The results show that our method achieves new state-of-the-art performance (test accuracy 89.1%)

on ATIS dataset and gets competitive performance on OVERNIGHT dataset.

2 Primal and Dual Tasks of Semantic Parsing

Before discussing the dual learning algorithm for semantic parsing, we first present the primal and dual tasks (as mentioned before) in detail. The primal and dual tasks are modeled on the attention-based Encoder-Decoder architectures (i.e. Seq2seq) which have been successfully applied in neural semantic parsing (Jia and Liang, 2016; Dong and Lapata, 2016). We also include copy mechanism (Gulcehre et al., 2016; See et al., 2017) to tackle unknown tokens.

2.1 Primal Task

The primal task is semantic parsing which converts queries into logical forms ($Q2LF$). Let $x = x_1 \cdots x_{|x|}$ denote the query, and $y = y_1 \cdots y_{|y|}$ denote the logical form. An *encoder* is exploited to encode the query x into vector representations, and a *decoder* learns to generate logical form y depending on the encoding vectors.

Encoder Each word x_i is mapped to a fixed-dimensional vector by a word embedding function $\psi(\cdot)$ and then fed into a bidirectional LSTM (Hochreiter and Schmidhuber, 1997). The hidden vectors are recursively computed at the i -th time step via:

$$\vec{\mathbf{h}}_i = \text{f}_{\text{LSTM}}(\psi(x_i), \vec{\mathbf{h}}_{i-1}), i = 1, \cdots, |x| \quad (1)$$

$$\overleftarrow{\mathbf{h}}_i = \text{f}_{\text{LSTM}}(\psi(x_i), \overleftarrow{\mathbf{h}}_{i+1}), i = |x|, \cdots, 1 \quad (2)$$

$$\mathbf{h}_i = [\vec{\mathbf{h}}_i; \overleftarrow{\mathbf{h}}_i] \quad (3)$$

where $[\cdot; \cdot]$ denotes vector concatenation, $\mathbf{h}_i \in \mathbb{R}^{2n}$, n is the number of hidden cells and f_{LSTM} is the LSTM function.

Decoder Decoder is an unidirectional LSTM with the attention mechanism (Luong et al., 2015). The hidden vector at the t -th time step is computed by $\mathbf{s}_t = \text{f}_{\text{LSTM}}(\phi(y_{t-1}), \mathbf{s}_{t-1})$, where $\phi(\cdot)$ is the token embedding function for logical forms and $\mathbf{s}_t \in \mathbb{R}^n$. The hidden vector of the first time step is initialized as $\mathbf{s}_0 = \overleftarrow{\mathbf{h}}_1$. The attention weight for the current step t of the decoder, with the i -th step in the encoder is $a_i^t = \frac{\exp(u_i^t)}{\sum_{j=1}^{|x|} \exp(u_j^t)}$ and

$$u_i^t = \mathbf{v}^T \tanh(\mathbf{W}_1 \mathbf{h}_i + \mathbf{W}_2 \mathbf{s}_t + \mathbf{b}_a) \quad (4)$$

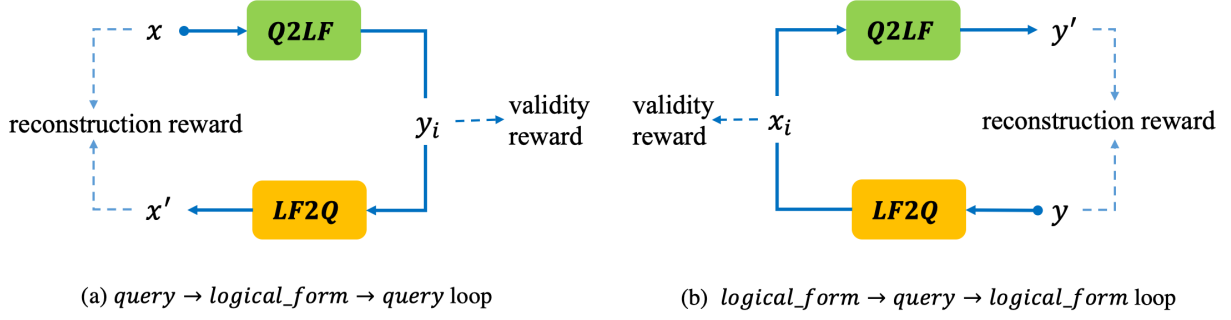


Figure 1: An overview of dual semantic parsing framework. The primal model ($Q2LF$) and dual model ($LF2Q$) can form a closed cycle. But there are two different directed loops, depending on whether they start from a query or logical form. Validity reward is used to estimate the quality of the middle generation output, and reconstruction reward is exploited to avoid information loss. The primal and dual models can be pre-trained and fine-tuned with labeled data to keep the models effective.

where $\mathbf{v}, \mathbf{b}_a \in \mathbb{R}^n$, and $\mathbf{W}_1 \in \mathbb{R}^{n \times 2n}$, $\mathbf{W}_2 \in \mathbb{R}^{n \times n}$ are parameters. Then we compute the vocabulary distribution $P_{gen}(y_t|y_{<t}, x)$ by

$$\mathbf{c}_t = \sum_{i=1}^{|x|} a_i^t \mathbf{h}_i \quad (5)$$

$$P_{gen}(y_t|y_{<t}, x) = \text{softmax}(\mathbf{W}_o[\mathbf{s}_t; \mathbf{c}_t] + \mathbf{b}_o) \quad (6)$$

where $\mathbf{W}_o \in \mathbb{R}^{|\mathcal{V}_y| \times 3n}$, $\mathbf{b}_o \in \mathbb{R}^{|\mathcal{V}_y|}$ and $|\mathcal{V}_y|$ is the output vocabulary size. Generation ends once an end-of-sequence token “EOS” is emitted.

Copy Mechanism We also include copy mechanism to improve model generalization following the implementation of See et al. (2017), a hybrid between Nallapati et al. (2016) and pointer network (Gulcehre et al., 2016). The predicted token is from either a fixed output vocabulary \mathcal{V}_y or raw input words x . We use sigmoid gate function σ to make a soft decision between generation and copy at each step t .

$$g_t = \sigma(\mathbf{v}_g^T[\mathbf{s}_t; \mathbf{c}_t; \phi(y_{t-1})] + b_g) \quad (7)$$

$$P(y_t|y_{<t}, x) = g_t P_{gen}(y_t|y_{<t}, x) + (1 - g_t) P_{copy}(y_t|y_{<t}, x) \quad (8)$$

where $g_t \in [0, 1]$ is the balance score, \mathbf{v}_g is a weight vector and b_g is a scalar bias. Distribution $P_{copy}(y_t|y_{<t}, x)$ will be described as follows.

Entity Mapping Although the copy mechanism can deal with unknown words, many raw words can not be directly copied to be part of a logical form. For example, kobe bryant is represented as `en.player.kobe_bryant` in OVERNIGHT (Wang et al., 2015). It is common that entities are identified by Uniform Resource Identifier (URI, Klyne and Carroll, 2006) in a

knowledge base. Thus, a mapping from raw words to URI is included after copying. Mathematically, P_{copy} in Eq.8 is calculated by:

$$P_{copy}(y_t = w|y_{<t}, x) = \sum_{i,j: KB(x_{i:j})=w} \sum_{k=i}^j a_k^t$$

where $i < j$, a_k^t is the attention weight of position k at decoding step t , $KB(\cdot)$ is a dictionary-like function mapping a specific noun phrase to the corresponding entity token in the vocabulary of logical forms.

2.2 Dual Model

The dual task ($LF2Q$) is an inverse of the primal task, which aims to generate a natural language query given a logical form. We can also exploit the attention-based Encoder-Decoder architecture (with copy mechanism or not) to build the dual model.

Reverse Entity Mapping Different with the primal task, we reversely map every possible KB entity y_t of a logical form to the corresponding noun phrase before query generation, $KB^{-1}(y_t)$ ¹. Since each KB entity may have multiple aliases in the real world, e.g. kobe bryant has a nickname the black mamba, we make different selections in two cases:

- For paired data, we select the noun phrase from $KB^{-1}(y_t)$, which exists in the query.
- For unpaired data, we randomly select one from $KB^{-1}(y_t)$.

¹ $KB^{-1}(\cdot)$ is the inverse operation of $KB(\cdot)$, which returns the set of all corresponding noun phrases given a KB entity.

3 Dual learning for Semantic Parsing

In this section, we present a semantic parsing framework with dual learning. We use one agent to represent the model of the primal task ($Q2LF$) and another agent to represent the model of the dual task ($LF2Q$), then design a two-agent game in a closed loop which can provide quality feedback to the primal and dual models even if only queries or logical forms are available. As the feedback reward may be non-differentiable, reinforcement learning algorithm (Sutton and Barto, 2018) based on policy gradient (Sutton et al., 2000) is applied for optimization.

Two agents, $Q2LF$ and $LF2Q$, participate in the collaborative game with two directed loops as illustrated in Figure 1. One loop `query`->`logical_form`->`query` starts from a query, generates possible logical forms by agent $Q2LF$ and tries to reconstruct the original query by $LF2Q$. The other loop `logical_form`->`query`->`logical_form` starts from the opposite side. Each agent will obtain quality feedback depending on reward functions defined in the directed loops.

3.1 Learning algorithm

Suppose we have fully labeled dataset $\mathcal{T} = \{\langle x, y \rangle\}$, unlabeled dataset \mathcal{Q} with only queries if available, and unlabeled dataset \mathcal{LF} with only logical forms if available. We firstly pre-train the primal model $Q2LF$ and the dual model $LF2Q$ on \mathcal{T} by maximum likelihood estimation (MLE). Let Θ_{Q2LF} and Θ_{LF2Q} denote all the parameters of $Q2LF$ and $LF2Q$ respectively. Our learning algorithm in each iteration consists of three parts:

3.1.1 Loop starts from a query

As shown in Figure 1 (a), we sample a query x from $\mathcal{Q} \cup \mathcal{T}$ randomly. Given x , $Q2LF$ model could generate k possible logical forms y_1, y_2, \dots, y_k via beam search (k is beam size). For each y_i , we can obtain a validity reward $R_q^{val}(y_i)$ (a scalar) computed by a specific reward function which will be discussed in Section 3.2.1. After feeding y_i into $LF2Q$, we finally get a reconstruction reward $R_q^{rec}(x, y_i)$ which forces the generated query as similar to x as possible and will be discussed in Section 3.2.2.

A hyper-parameter α is exploited to balance these two rewards in $r_i^q = \alpha R_q^{val}(y_i) + (1 - \alpha) R_q^{rec}(x, y_i)$, where $\alpha \in [0, 1]$.

By utilizing policy gradient (Sutton et al., 2000), the stochastic gradients of Θ_{Q2LF} and Θ_{LF2Q} are computed as:

$$\nabla_{\Theta_{Q2LF}} \hat{E}[r] = \frac{1}{k} \sum_{i=1}^k r_i^q \nabla_{\Theta_{Q2LF}} \log P(y_i|x; \Theta_{Q2LF})$$

$$\nabla_{\Theta_{LF2Q}} \hat{E}[r] = \frac{1 - \alpha}{k} \sum_{i=1}^k \nabla_{\Theta_{LF2Q}} \log P(x|y_i; \Theta_{LF2Q})$$

3.1.2 Loop starts from a logical form

As shown in Figure 1 (b), we sample a logical form y from $\mathcal{LF} \cup \mathcal{T}$ randomly. Given y , $LF2Q$ model generates k possible queries x_1, x_2, \dots, x_k via beam search. For each x_i , we can obtain a validity reward $R_{lf}^{val}(x_i)$ (a scalar) which will also be discussed in Section 3.2.1. After feeding x_i into $Q2LF$, we can also get a reconstruction reward $R_{lf}^{rec}(y, x_i)$, which forces the generated logical form as similar to y as possible and will be discussed in Section 3.2.2.

A hyper-parameter β is also exploited to balance these two rewards by $r_i^{lf} = \beta R_{lf}^{val}(x_i) + (1 - \beta) R_{lf}^{rec}(y, x_i)$, where $\beta \in [0, 1]$.

By utilizing policy gradient, the stochastic gradients of Θ_{Q2LF} and Θ_{LF2Q} are computed as:

$$\nabla_{\Theta_{LF2Q}} \hat{E}[r] = \frac{1}{k} \sum_{i=1}^k r_i^{lf} \nabla_{\Theta_{LF2Q}} \log P(x_i|y; \Theta_{LF2Q})$$

$$\nabla_{\Theta_{Q2LF}} \hat{E}[r] = \frac{1 - \beta}{k} \sum_{i=1}^k \nabla_{\Theta_{Q2LF}} \log P(y|x_i; \Theta_{Q2LF})$$

3.1.3 Supervisor guidance

The previous two stages are unsupervised learning processes, which need no labeled data. If there is no supervision for the primal and dual models after pre-training, these two models would be rotten especially when \mathcal{T} is limited.

To keep the learning process stable and prevent the models from crashes, we randomly select samples from \mathcal{T} to fine-tune the primal and dual models by maximum likelihood estimation (MLE). Details about the dual learning algorithm for semantic parsing are provided in Appendix A.

3.2 Reward design

As mentioned in Section 3.1, there are two types of reward functions in each loop: validity reward (R_q^{val} , R_{lf}^{val}) and reconstruction reward (R_q^{rec} , R_{lf}^{rec}). But each type of reward function may be different in different loops.

3.2.1 Validity reward

Validity reward is used to evaluate the quality of intermediate outputs in a loop (see Figure 1). In the loop starts from a query, it indicates whether the generated logical forms are well-formed at the surface and semantic levels. While in the loop starts from a logical form, it indicates how natural and fluent the intermediate queries are.

Loop starts from a query: We estimate the quality of the generated logical forms at two levels, i.e. surface and semantics. Firstly, we check whether the logical form is a complete tree without parentheses mismatching. As for semantics, we check whether the logical form is understandable without errors like type inconsistency. It can be formulated as

$$R_q^{val}(y) = \text{grammar_error_indicator}(y) \quad (9)$$

which returns 1 when y has no error at the surface and semantic levels, and returns 0 otherwise.

If there exists an executing program or search engine for logical form y , e.g. dataset OVERNIGHT (Wang et al., 2015), $\text{grammar_error_indicator}(\cdot)$ has been included.

Otherwise, we should construct a grammar error indicator based on the ontology of the corresponding dataset. For example, a specification of ATIS can be extracted by clarifying all (1) entities paired with corresponding types, (2) unary/binary predicates with argument constraints (see Table 1). Accordingly, Algorithm 1 abstracts the procedure of checking the surface and semantics for a logical form candidate y based on the specification.

category	type	instances	
entity	me	lunch:me, snack:me	
	al	delta:al, usair:al	
	pd	morning:pd, late:pd	
category	instances	args0	
unary	_city	ci	
	_airport	ap	
	_oneway	flight	
category	instances	args0	args1
binary	_from	flight	ci
	_services	al	ci
	_during_day	flight	pd

Table 1: A truncated specification for ATIS.

Loop starts from a logical form: A language model (LM) is exploited to evaluate the quality of intermediate queries (Mikolov et al., 2010). We

Algorithm 1 Grammar error indicator on ATIS

Input: Logical form string y ; specification \mathcal{D}

Output: 1/0, whether y is valid

```

1: if  $\text{to\_lisp\_tree}(y)$  succeed then
2:    $\text{lispTree} \leftarrow \text{to\_lisp\_tree}(y)$ 
    $\triangleright$  using Depth-First-Search for  $\text{lispTree}$ 
3:   if  $\text{type\_consistent}(\text{lispTree}, \mathcal{D})$  then
4:     return 1
5:   end if
6: end if
7: return 0

```

apply length-normalization (Wu et al., 2016) to make a fair competition between short and long queries.

$$R_{lf}^{val}(x) = \log LM_q(x) / \text{Length}(x), \quad (10)$$

where $LM_q(\cdot)$ is a language model pre-trained on all the queries of $\mathcal{Q} \cup \mathcal{T}$ (referred in Section 3.1).

3.2.2 Reconstruction reward

Reconstruction reward is used to estimate how similar the output of one loop is compared with the input. We take log likelihood as reconstruction rewards for the loop starts from a query and the loop starts from a logical form. Thus,

$$R_q^{rec}(x, y_i) = \log P(x|y_i; \Theta_{LF2Q})$$

$$R_{lf}^{rec}(y, x_i) = \log P(y|x_i; \Theta_{Q2LF})$$

where y_i and x_i are intermediate outputs.

4 Experiment

In this section, we evaluate our framework on the ATIS and OVERNIGHT datasets.

4.1 Dataset

ATIS We use the preprocessed version provided by Dong and Lapata (2018), where natural language queries are lowercased and stemmed with NLTK (Loper and Bird, 2002), and entity mentions are replaced by numbered markers. We also leverage an external lexicon that maps word phrases (e.g., first class) to entities (e.g., first:cl) like what Jia and Liang (2016) did.

OVERNIGHT It contains natural language paraphrases paired with logical forms across eight domains. We follow the traditional 80%/20% train/valid splits as Wang et al. (2015) to choose the best model during training.

ATIS and OVERNIGHT never provide unlabeled queries. To test our method in semi-supervised learning, we keep a part of the training set as fully labeled data and leave the rest as unpaired queries and logical forms which simulate unlabeled data.

4.2 Synthesis of logical forms

Although there is no unlabeled query provided in most semantic parsing benchmarks, it should be easy to synthesize logical forms. Since a logical form is strictly structured and can be modified from the existing one or created from simple grammars, it is much cheaper than query collection. Our synthesized logical forms are public ².

4.2.1 Modification based on ontology

On ATIS, we randomly sample a logical form from the training set, and select one entity or predicate for replacement according to the specification in Table 1. If the new logical form after replacement is valid and never seen, it is added to the unsupervised set. 4592 new logical forms are created for ATIS. An example is shown in Figure 2.

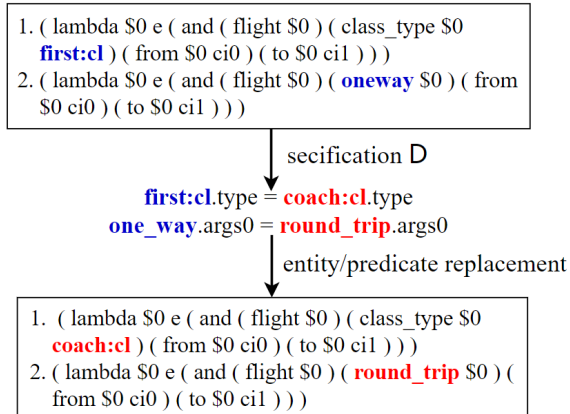


Figure 2: Synthesis of logical forms by replacement.

4.2.2 Generation based on grammar

Wang et al. (2015) proposed an underlying grammar to generate logical forms along with their corresponding canonical utterances on OVERNIGHT, which can be found in SEMPRES ³. We reorder the entity instances (e.g., ENTITYNP) of one type (e.g., TYPENP) in grammar files to generate new logical forms. We could include new entity instances if we want more unseen logical forms, but we didn't do that actually. Finally, we get about

²<https://github.com/RhythmCao/Synthesized-Logical-Forms>

³<https://github.com/percyliang/sempre>

500 new logical forms for each domain on average. More examples can be found in Appendix B.

4.3 Experimental settings

4.3.1 Base models

We use 200 hidden units and 100-dimensional word vectors for all encoders and decoders of $Q2LF$ and $LF2Q$ models. The LSTMs used are in single-layer. Word embeddings on query side are initialized by Glove6B (Pennington et al., 2014). Out-of-vocabulary words are replaced with a special token $\langle unk \rangle$. Other parameters are initialized by uniformly sampling within the interval $[-0.2, 0.2]$. The language model we used is also a single-layer LSTM with 200 hidden units and 100-dim word embedding layer.

4.3.2 Training and decoding

We individually pre-train $Q2LF/LF2Q$ models using only labeled data and language model LM_q using both labeled and unlabeled queries. The language model is fixed for calculating reward. The hyper-parameters α and β are selected according to validation set (0.5 is used), and beam size k is selected from $\{3, 5\}$. The batch size is selected from $\{10, 20\}$. We use optimizer Adam (Kingma and Ba, 2014) with learning rate 0.001 for all experiments. Finally, we evaluate the primal model ($Q2LF$, semantic parsing) and report test accuracy on each dataset.

4.4 Results and analysis

We perform a PSEUDO baseline following the setup in Sennrich et al. (2016) and Guo et al. (2018). The pre-trained $LF2Q$ or $Q2LF$ model is used to generate pseudo $\langle query, logical form \rangle$ pairs from unlabeled logical forms or unlabeled queries, which extends the training set. The pseudo-labeled data is used carefully with a discount factor (0.5) in loss function (Lee, 2013), when we train $Q2LF$ by supervised training.

4.4.1 Main results

The results are illustrated in Table 2 and 3. ATT and ATTPTR represent that the primal/dual models are attention-based Seq2seq and attention-based Seq2seq with copy mechanism respectively. We train models with the dual learning algorithm if DUAL is included, otherwise we only train the primal model by supervised training. \mathcal{LF} refers to the synthesized logical forms. PSEUDO uses the

Method	Bas.	Blo.	Cal.	Hou.	Pub.	Rec.	Res.	Soc.	Avg.
SPO (Wang et al., 2015)	46.3	41.9	74.4	54.0	59.0	70.8	75.9	48.2	58.8
DSP-C (Xiao et al., 2016)	80.5	55.6	75.0	61.9	75.8	—	80.1	80.0	72.7
NO RECOMBINATION (Jia and Liang, 2016)	85.2	58.1	78.0	71.4	76.4	79.6	76.2	81.4	75.8
DATA RECOMB (Jia and Liang, 2016)*(+data)	87.5	60.2	81.0	72.5	78.3	81.0	79.5	79.6	77.5
CROSSDOMAIN (Su and Yan, 2017)	88.2	62.2	82.1	78.8	80.1	86.1	83.7	83.1	80.6
DOMAINENCODING (Herzig and Berant, 2017)	86.2	62.7	82.1	78.3	80.7	82.9	82.2	81.7	79.6
SEQ2ACTION (Chen et al., 2018a)	88.2	61.4	81.5	74.1	80.7	82.9	80.7	82.1	79.0
ATT	86.2	61.4	76.4	68.8	75.2	76.9	78.9	82.2	75.7
ATT + PSEUDO(\mathcal{LF})	87.2	60.9	73.2	71.4	75.8	80.1	79.2	82.0	76.2
ATT + DUAL	87.5	63.7	79.8	73.0	81.4	81.5	81.6	83.0	78.9
ATT + DUAL + \mathcal{LF}	88.0	65.2	80.7	76.7	80.7	82.4	84.0	83.8	80.2
ATTPTR	86.7	63.2	74.4	69.3	75.8	77.8	78.3	82.2	76.0
ATTPTR + PSEUDO(\mathcal{LF})	85.7	63.4	74.4	69.8	78.3	78.7	79.8	82.1	76.5
ATTPTR + DUAL	87.7	63.4	77.4	74.1	80.1	80.1	82.5	83.4	78.6
ATTPTR + DUAL + \mathcal{LF}	87.0	66.2	79.8	75.1	80.7	83.3	83.4	83.8	79.9

Table 2: Test accuracies on OVERNIGHT compared with previous systems.

Method	ATIS
ZC07 (Zettlemoyer and Collins, 2007)	84.6
FUBL (Kwiatkowski et al., 2011)	82.8
GUSP++ (Poon, 2013)	83.5
TISP (Zhao and Huang, 2015)	84.2
SEQ2TREE (Dong and Lapata, 2016)	84.6
ASN+SUPATT (Rabinovich et al., 2017)	85.9
TRANX (Yin and Neubig, 2018)	86.2
COARSE2FINE (Dong and Lapata, 2018)	87.7
ATT	84.4
ATT + PSEUDO(\mathcal{LF})	83.3
ATT + DUAL	86.4
ATT + DUAL + \mathcal{LF}	87.1
ATTPTR	85.7
ATTPTR + PSEUDO(\mathcal{LF})	86.2
ATTPTR + DUAL	88.6
ATTPTR + DUAL + \mathcal{LF}	89.1

Table 3: Test accuracies on ATIS compared with previous systems.

$LF2Q$ model and \mathcal{LF} to generate pseudo-labeled data. From the overall results, we can see that:

1) Even without the additional logical forms by synthesizing, the dual learning based semantic parser can outperform our baselines with supervised training, e.g., “ATT + DUAL” gets much better performances than “ATT + PSEUDO(\mathcal{LF})” in Table 2 and 3. We think the $Q2LF$ and $LF2Q$ models can teach each other in dual learning: one model sends informative signals to help regularize the other model. Actually, it can also be explained as a data augmentation procedure, e.g., $Q2LF$ can generate samples utilized by $LF2Q$ and vice versa. While the PSEUDO greatly depends on the quality of pseudo-samples even if a discount factor is considered.

2) By involving the synthesized logical forms \mathcal{LF} in the dual learning for each domain respectively, the performances are improved further. We

achieve state-of-the-art performance (89.1%)⁴ on ATIS as shown in Table 3. On OVERNIGHT dataset, we achieve a competitive performance on average (80.2%). The best average accuracy is from Su and Yan (2017), which benefits from cross-domain training. We believe our method could get more improvements with stronger primal models (e.g. with domain adaptation). Our method would be compatible with various models.

3) Copy mechanism can remarkably improve accuracy on ATIS, while not on OVERNIGHT. The average accuracy even decreases from 80.2% to 79.9% when using the copy mechanism. We argue that OVERNIGHT dataset contains a very small number of distinct entities that copy is not essential, and it contains less training samples than ATIS. This phenomenon also exists in Jia and Liang (2016).

4.4.2 Ablation study

Semi-supervised learning We keep a part of the training set as labeled data \mathcal{T} randomly and leave the rest as unpaired queries (\mathcal{Q}) and logical forms (\mathcal{LF}) to validate our method in a semi-supervised setting. The ratio of labeled data is 50%. PSEUDO here uses the $Q2LF$ model and \mathcal{Q} to generate pseudo-labeled data, as well as $LF2Q$ model and \mathcal{LF} . From Table 4, we can see that the dual learning method outperforms the PSEUDO baseline in two datasets dramatically. The dual learning method is more efficient to exploit unlabeled data. In general, both unpaired queries and logi-

⁴Although there is another published result that achieved better performance by using word class information from Wiktionary (Wang et al., 2014), it is unfair to compare it with our results and other previous systems which only exploit data resources of ATIS.

Method	Bas.	Blo.	Cal.	Hou.	Pub.	Rec.	Res.	Soc.	Avg.	ATIS
ATT	80.1	55.4	61.9	53.4	60.2	64.4	71.1	76.8	65.4	78.6
ATT+PSEUDO(\mathcal{Q})	80.1	59.4	60.1	52.9	59.6	66.2	73.8	79.0	66.4	78.8
ATT+PSEUDO(\mathcal{LF})	83.9	60.4	64.3	54.5	58.4	69.0	70.5	77.3	67.3	77.9
ATT+PSEUDO($\mathcal{LF}+\mathcal{Q}$)	80.6	59.1	61.9	57.1	62.7	65.3	73.2	79.8	67.5	78.3
ATT+DUAL	82.6	60.2	72.0	58.7	66.5	73.6	74.1	79.3	70.9	79.5
ATT+DUAL+ \mathcal{Q}	83.9	60.7	70.2	60.8	69.6	71.3	76.2	79.8	71.6	79.7
ATT+DUAL+ \mathcal{LF}	83.4	61.4	71.4	59.3	70.2	73.1	75.3	78.6	71.6	80.4
ATT+DUAL+ $\mathcal{LF}+\mathcal{Q}$	85.4	62.9	73.2	59.3	72.0	75.5	75.6	79.5	72.9	81.7
ATT+PTR	81.1	58.1	63.1	48.7	55.3	69.4	68.4	77.4	65.2	84.8
ATT+PTR+PSEUDO(\mathcal{Q})	82.1	59.6	61.3	47.6	57.1	72.2	69.9	78.4	66.0	85.0
ATT+PTR+PSEUDO(\mathcal{LF})	82.4	59.1	62.5	54.5	63.4	71.3	69.6	77.6	67.5	86.2
ATT+PTR+PSEUDO($\mathcal{LF}+\mathcal{Q}$)	81.3	59.4	65.5	49.2	58.4	72.7	72.0	78.6	67.1	85.0
ATT+PTR+DUAL	81.8	60.2	68.5	57.1	65.2	72.2	74.1	79.0	69.8	86.2
ATT+PTR+DUAL+ \mathcal{Q}	81.6	60.7	69.6	61.4	68.9	74.1	79.8	80.1	72.0	86.6
ATT+PTR+DUAL+ \mathcal{LF}	82.6	62.2	68.5	62.4	69.6	73.1	77.4	79.4	71.9	87.3
ATT+PTR+DUAL+ $\mathcal{LF}+\mathcal{Q}$	83.6	62.2	72.6	61.9	71.4	75.0	76.5	80.4	73.0	86.8

Table 4: Semi-supervised learning experiments. We keep 50% of the training set as labeled data randomly, and leave the rest as unpaired queries(\mathcal{Q}) and logical forms(\mathcal{LF}) to simulate unsupervised dataset.

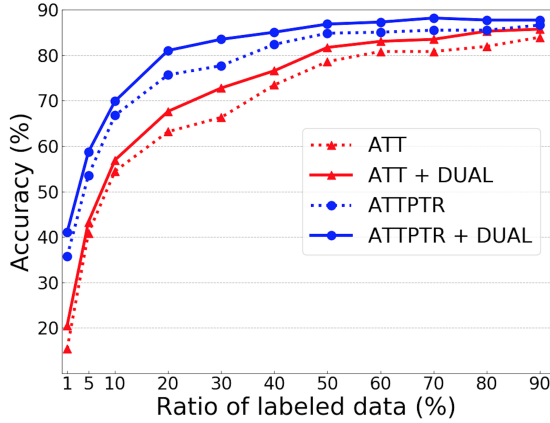


Figure 3: Test accuracies on ATIS. It varies the ratio of labeled data, and keeps the rest as unlabeled data.

cal forms could boost the performance of semantic parsers with dual learning.

Different ratios To investigate the efficiency of our method in semi-supervised learning, we vary the ratio of labeled data kept on ATIS from 1% to 90%. In Figure 3, we can see that dual learning strategy enhances semantic parsing over all proportions. The prominent gap happens when the ratio is between 0.2 and 0.4. Generally, the more unlabeled data we have, the more remarkable the leap is. However, if the labeled data is really limited, less supervision can be exploited to keep the primal and dual models reasonable. For example, when the ratio of labeled data is from only 1% to 10%, the improvement is not that obvious.

Does more unlabeled data give better result? We also fix the ratio of labeled data as 30%, and change the ratio of unlabeled samples to the rest data on ATIS, as illustrated in Figure 4. Even

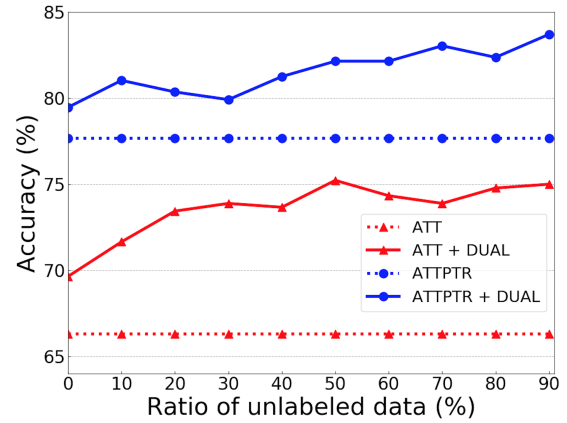


Figure 4: Test accuracies on ATIS. It fixes the ratio of labeled data as 30%, and varies the ratio of unlabeled samples to the rest data.

without unlabeled data (i.e. the ratio of unlabeled data is zero), the dual learning based semantic parser can outperform our baselines. However, the performance of our method doesn’t improve constantly, when the amount of unlabeled data is increased. We think the power of the primal and dual models is constrained by the limited amount of labeled data. When some complex queries or logical forms come, the two models may converge to an equilibrium where the intermediate value loses some implicit semantic information, but the rewards are high.

Choice for validity reward We conduct another experiment by changing validity reward in Eq.9 with length-normalized LM score (i.e. language model of logical forms) like Eq.10. Results (Table 5) show that “hard” surface/semantic check is more suitable than “soft” probability of logical

Method	Validity	ATIS	OVERNIGHT
ATT	LM_{lf}	80.6	71.5
+ DUAL	grammar check	81.7	72.9
ATTPTR	LM_{lf}	86.2	71.4
+ DUAL	grammar check	86.8	73.0

Table 5: Test accuracies on ATIS and OVERNIGHT in semi-supervised learning setting (the ratio of labeled data is 50%). On OVERNIGHT, we average across all eight domains. LM_{lf} means using a logical form language model for validity reward, while “grammar check” means using the surface and semantic check.

form LM. We think that simple language models may suffer from long-dependency and data imbalance issues, and it is hard to capture inner structures of logical forms from a sequential model.

5 Related Work

Lack of data A semantic parser can be trained from labeled logical forms or weakly supervised samples (Krishnamurthy and Mitchell, 2012; Berant et al., 2013; Liang et al., 2017; Goldman et al., 2018). Yih et al. (2016) demonstrate logical forms can be collected efficiently and more useful than merely answers to queries. Wang et al. (2015) construct a semantic parsing dataset starting from grammar rules to crowdsourcing paraphrase. Jia and Liang (2016) induces synchronous context-free grammar (SCFG) and creates new “recombinant” examples accordingly. Su and Yan (2017) use multiple source domains to reduce the cost of collecting data for the target domain. Guo et al. (2018) pre-train a question generation model to produce pseudo-labeled data as a supplement. In this paper, we introduce the dual learning to make full use of data (both labeled and unlabeled). Yin et al. (2018) introduce a variational auto-encoding model for semi-supervised semantic parsing. Beyond semantic parsing, the semi-supervised and adaptive learnings are also typical in natural language understanding (Tur et al., 2005; Bapna et al., 2017; Zhu et al., 2014, 2018; Zhu and Yu, 2018).

Constrained decoding To avoid invalid parses, additional restrictions must be considered in the decoding. Dong and Lapata (2016) propose SEQ2TREE method to ensure the matching of parentheses, which can generate syntactically valid output. Cheng et al. (2017) and Dong and Lapata (2018) both try to decode in two steps, from a coarse rough sketch to a finer structure hierarchically. Krishnamurthy et al. (2017) define a

grammar of production rules such that only well-typed logical forms can be generated. Yin and Neubig (2017) and Chen et al. (2018a) both transform the generation of logical forms into query graph construction. Zhao et al. (2019) propose a hierarchical parsing model following the structure of semantic representations, which is predefined by domain developers. We introduce a validity reward at the surface and semantic levels in the dual learning algorithm as a constraint signal.

Dual learning Dual learning framework is first proposed to improve neural machine translation (NMT) (He et al., 2016). Actually, the primal and dual tasks are symmetric in NMT, while not in semantic parsing. The idea of dual learning has been applied in various tasks (Xia et al., 2017), such as Question Answering/Generation (Tang et al., 2017, 2018), Image-to-Image Translation (Yi et al., 2017) and Open-domain Information Extraction/Narration (Sun et al., 2018). We are the first to introduce dual learning in semantic parsing to the best of our knowledge.

6 Conclusions and Future Work

In this paper, we develop a semantic parsing framework based on dual learning algorithm, which enables a semantic parser to fully utilize labeled and even unlabeled data through a dual-learning game between the primal and dual models. We also propose a novel reward function at the surface and semantic levels by utilizing the prior-knowledge of logical form structures. Thus, the primal model tends to generate complete and reasonable semantic representation. Experimental results show that semantic parsing based on dual learning improves performance across datasets.

In the future, we want to incorporate this framework with much refined primal and dual models, and design more informative reward signals to make the training more efficient. It would be appealing to apply graph neural networks (Chen et al., 2018b, 2019) to model structured logical forms.

Acknowledgments

This work has been supported by the National Key Research and Development Program of China (Grant No.2017YFB1002102) and the China NSFC projects (No. 61573241).

References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Ankur Bapna, Gökhan Tür, Dilek Hakkani-Tür, and Larry P. Heck. 2017. Towards zero-shot frame semantic parsing for domain scaling. In *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association*, pages 2476–2480.
- Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544.
- Bo Chen, Le Sun, and Xianpei Han. 2018a. Sequence-to-action: End-to-end semantic graph generation for semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 766–777.
- Lu Chen, Zhi Chen, Bowen Tan, Sishan Long, Milica Gasic, and Kai Yu. 2019. Agentgraph: Towards universal dialogue management with structured deep reinforcement learning. *arXiv preprint arXiv:1905.11259*.
- Lu Chen, Bowen Tan, Sishan Long, and Kai Yu. 2018b. Structured dialogue policy with graph neural networks. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1257–1268, Santa Fe, New Mexico, USA.
- Jianpeng Cheng, Siva Reddy, Vijay Saraswat, and Mirella Lapata. 2017. Learning structured natural language representations for semantic parsing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 44–55.
- 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.
- 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.
- Omer Goldman, Veronica Latcinnik, Ehud Nave, Amir Globerson, and Jonathan Berant. 2018. Weakly supervised semantic parsing with abstract examples. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1809–1819.
- Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 140–149.
- Daya Guo, Yibo Sun, Duyu Tang, Nan Duan, Jian Yin, Hong Chi, James Cao, Peng Chen, and Ming Zhou. 2018. Question generation from sql queries improves neural semantic parsing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1597–1607.
- Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. 2016. Dual learning for machine translation. In *Advances in Neural Information Processing Systems*, pages 820–828.
- Jonathan Herzig and Jonathan Berant. 2017. Neural semantic parsing over multiple knowledge-bases. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 623–628.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- 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 (Volume 1: Long Papers)*, pages 12–22.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Graham Klyne and Jeremy J Carroll. 2006. Resource description framework (rdf): Concepts and abstract syntax.
- Jayant Krishnamurthy, Pradeep Dasigi, and Matt Gardner. 2017. Neural semantic parsing with type constraints for semi-structured tables. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1516–1526.
- Jayant Krishnamurthy and Tom M Mitchell. 2012. Weakly supervised training of semantic parsers. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 754–765.
- Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2011. Lexical generalization in ccg grammar induction for semantic parsing. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1512–1523.
- Dong-Hyun Lee. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*.

- Chen Liang, Jonathan Berant, Quoc Le, Kenneth D. Forbus, and Ni Lao. 2017. Neural symbolic machines: Learning semantic parsers on freebase with weak supervision. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23–33.
- Edward Loper and Steven Bird. 2002. NLTK: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*.
- Wei Lu, Hwee Tou Ng, Wee Sun Lee, and Luke S Zettlemoyer. 2008. A generative model for parsing natural language to meaning representations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 783–792.
- Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.
- Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*.
- Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.
- Hoifung Poon. 2013. Grounded unsupervised semantic parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 933–943.
- Maxim Rabinovich, Mitchell Stern, and Dan Klein. 2017. Abstract syntax networks for code generation and semantic parsing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1139–1149.
- Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96.
- Yu Su and Xifeng Yan. 2017. Cross-domain semantic parsing via paraphrasing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1235–1246.
- Mingming Sun, Xu Li, and Ping Li. 2018. Logician and orator: Learning from the duality between language and knowledge in open domain. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2119–2130.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.
- Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.
- Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. 2000. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063.
- Duyu Tang, Nan Duan, Tao Qin, Zhao Yan, and Ming Zhou. 2017. Question answering and question generation as dual tasks. *arXiv preprint arXiv:1706.02027*.
- Duyu Tang, Nan Duan, Zhao Yan, Zhirui Zhang, Yibo Sun, Shujie Liu, Yuanhua Lv, and Ming Zhou. 2018. Learning to collaborate for question answering and asking. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1564–1574.
- Gokhan Tur, Dilek Hakkani-Tür, and Robert E. Schapire. 2005. Combining active and semi-supervised learning for spoken language understanding. *Speech Communication*, 45(2):171–186.
- Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In *Advances in neural information processing systems*, pages 2773–2781.
- Adrienne Wang, Tom Kwiatkowski, and Luke Zettlemoyer. 2014. Morpho-syntactic lexical generalization for CCG semantic parsing. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1284–1295, Doha, Qatar. Association for Computational Linguistics.

- Yushi Wang, Jonathan Berant, and Percy Liang. 2015. Building a semantic parser overnight. In *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)*, pages 1332–1342.
- Yuk Wah Wong and Raymond Mooney. 2007. Learning synchronous grammars for semantic parsing with lambda calculus. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 960–967.
- Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google’s neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.
- Yingce Xia, Tao Qin, Wei Chen, Jiang Bian, Nenghai Yu, and Tie-Yan Liu. 2017. Dual supervised learning. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3789–3798.
- Chunyang Xiao, Marc Dymetman, and Claire Gardent. 2016. Sequence-based structured prediction for semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1341–1350.
- Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. 2017. Dualgan: Unsupervised dual learning for image-to-image translation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2849–2857.
- Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–206.
- 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 (Volume 1: Long Papers)*, pages 440–450.
- Pengcheng Yin and Graham Neubig. 2018. Tranx: A transition-based neural abstract syntax parser for semantic parsing and code generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 7–12.
- Pengcheng Yin, Chunting Zhou, Junxian He, and Graham Neubig. 2018. StructVAE: Tree-structured latent variable models for semi-supervised semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 754–765, Melbourne, Australia. Association for Computational Linguistics.
- John M Zelle and Raymond J Mooney. 1996. Learning to parse database queries using inductive logic programming. In *Proceedings of the 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 (EMNLP-CoNLL)*.
- Luke S. Zettlemoyer and Michael Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorical grammars. In *UAI*, pages 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.
- Zijian Zhao, Su Zhu, and Kai Yu. 2019. A hierarchical decoding model for spoken language understanding from unaligned data. In *ICASSP 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7305–7309. IEEE.
- Su Zhu, Lu Chen, Kai Sun, Da Zheng, and Kai Yu. 2014. Semantic parser enhancement for dialogue domain extension with little data. In *Spoken Language Technology Workshop (SLT), 2014 IEEE*, pages 336–341. IEEE.
- Su Zhu, Ouyu Lan, and Kai Yu. 2018. Robust spoken language understanding with unsupervised asr-error adaptation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2018*, pages 6179–6183. IEEE.
- Su Zhu and Kai Yu. 2018. Concept transfer learning for adaptive language understanding. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 391–399. Association for Computational Linguistics.

A Detailed algorithm

Algorithm 2 Dual Learning Framework for Semantic Parsing

Input:

- 1: Supervised dataset $\mathcal{T} = \{\langle x, y \rangle\}$; Unsupervised dataset for queries \mathcal{Q} and logical forms \mathcal{LF} ;
- 2: Pre-trained models on \mathcal{T} : $Q2LF$ model $P(y|x; \Theta_{Q2LF})$, $LF2Q$ model $P(x|y; \Theta_{LF2Q})$;
- 3: Pre-trained model on \mathcal{Q} and queries of \mathcal{T} : Language Model for queries LM_q ;
- 4: Indicator performs surface and semantic check for a logical form: $\text{grammar_error_indicator}(\cdot)$;
- 5: Beam search size k , hyper parameters α and β , learning rate η_1 for $Q2LF$ and η_2 for $LF2Q$;

Output: Parameters Θ_{Q2LF} of $Q2LF$ model

6: **repeat**

7: ▷ Reinforcement learning process uses unlabeled data, also reuses labeled data

- | | |
|---|---|
| <p>8: Sample a query x from $\mathcal{Q} \cup \mathcal{T}$;</p> <p>9: $Q2LF$ model generates k logical forms y_1, y_2, \dots, y_k via beam search;</p> | <p>18: Sample a logical form y from $\mathcal{LF} \cup \mathcal{T}$;</p> <p>19: $LF2Q$ model generates k queries x_1, x_2, \dots, x_k via beam search;</p> |
|---|---|

- | | |
|---|--|
| <p>10: for each possible logical form y_i do</p> <p>11: Obtain validity reward for y_i as</p> | <p>20: for each possible query x_i do</p> <p>21: Obtain validity reward for x_i as</p> |
|---|--|

$R_q^{val}(y_i) = \text{grammar_error_indicator}(y_i)$	$R_{lf}^{val}(x_i) = \log LM_q(x_i) / \text{Length}(x_i)$
--	---

- | | |
|--|---|
| <p>12: Get reconstruction reward for y_i as</p> <p style="margin-left: 40px;">$R_q^{rec}(x, y_i) = \log P(x y_i; \Theta_{LF2Q})$</p> | <p>22: Get reconstruction reward for x_i as</p> <p style="margin-left: 40px;">$R_{lf}^{rec}(y, x_i) = \log P(y x_i; \Theta_{Q2LF})$</p> |
|--|---|

- | | |
|---|--|
| <p>13: Compute total reward for y_i as</p> <p style="margin-left: 40px;">$r_i^q = \alpha R_q^{val}(y_i) + (1 - \alpha) R_q^{rec}(x, y_i)$</p> | <p>23: Compute total reward for x_i as</p> <p style="margin-left: 40px;">$r_i^{lf} = \beta R_{lf}^{val}(x_i) + (1 - \beta) R_{lf}^{rec}(y, x_i)$</p> |
|---|--|

- | | |
|---|---|
| <p>14: end for</p> <p>15: Compute stochastic gradient of Θ_{Q2LF}:</p> | <p>24: end for</p> <p>25: Compute stochastic gradient of Θ_{LF2Q}:</p> |
|---|---|

$\nabla_{\Theta_{Q2LF}} \hat{E}[r] = \frac{1}{k} \sum_{i=1}^k r_i^q \nabla_{\Theta_{Q2LF}} \log P(y_i x; \Theta_{Q2LF})$	$\nabla_{\Theta_{LF2Q}} \hat{E}[r] = \frac{1}{k} \sum_{i=1}^k r_i^{lf} \nabla_{\Theta_{LF2Q}} \log P(x_i y; \Theta_{LF2Q})$
--	---

- | | |
|---|---|
| <p>16: Compute stochastic gradient of Θ_{LF2Q}:</p> | <p>26: Compute stochastic gradient of Θ_{Q2LF}:</p> |
|---|---|

$\nabla_{\Theta_{LF2Q}} \hat{E}[r] = \frac{1 - \alpha}{k} \sum_{i=1}^k \nabla_{\Theta_{LF2Q}} \log P(x y_i; \Theta_{LF2Q})$	$\nabla_{\Theta_{Q2LF}} \hat{E}[r] = \frac{1 - \beta}{k} \sum_{i=1}^k \nabla_{\Theta_{Q2LF}} \log P(y x_i; \Theta_{Q2LF})$
---	--

- | | |
|-------------------------------|-------------------------------|
| <p>17: Model updates:</p> | <p>27: Model updates:</p> |
|-------------------------------|-------------------------------|

$\Theta_{Q2LF} \leftarrow \Theta_{Q2LF} + \eta_1 \cdot \nabla_{\Theta_{Q2LF}} \hat{E}[r]$ $\Theta_{LF2Q} \leftarrow \Theta_{LF2Q} + \eta_2 \cdot \nabla_{\Theta_{LF2Q}} \hat{E}[r]$	$\Theta_{LF2Q} \leftarrow \Theta_{LF2Q} + \eta_2 \cdot \nabla_{\Theta_{LF2Q}} \hat{E}[r]$ $\Theta_{Q2LF} \leftarrow \Theta_{Q2LF} + \eta_1 \cdot \nabla_{\Theta_{Q2LF}} \hat{E}[r]$
--	--

▷ After reinforcement learning process, use labeled data to fine-tune models

- 28: Sample a $\langle x, y \rangle$ pair from \mathcal{T} ;
 - 29: Update Θ_{Q2LF} by $\Theta_{Q2LF} \leftarrow \Theta_{Q2LF} + \eta_1 \cdot \nabla_{\Theta_{Q2LF}} \log P(y|x; \Theta_{Q2LF})$
 - 30: Update Θ_{LF2Q} by $\Theta_{LF2Q} \leftarrow \Theta_{LF2Q} + \eta_2 \cdot \nabla_{\Theta_{LF2Q}} \log P(x|y; \Theta_{LF2Q})$
 - 31: **until** $Q2LF$ model converges
-

B Examples of synthesized logical forms

Original	After Modification
Entity Replacement	
(lambda \$0 e (and (flight \$0) (meal \$0 lunch:me) (from \$0 ci0) (to \$0 ci1)))	(lambda \$0 e (and (flight \$0) (meal \$0 dinner:me) (from \$0 ci0) (to \$0 ci1)))
(= al0 (abbrev delta:al))	(= al0 (abbrev usair:al))
(lambda \$0 e (and (flight \$0) (class_type \$0 thrif:cl) (from \$0 ci1) (to \$0 ci0)))	(lambda \$0 e (and (flight \$0) (class_type \$0 business:cl) (from \$0 ci1) (to \$0 ci0)))
Unary Replacement	
(lambda \$0 e (exists \$1 (and (round_trip \$1) (from \$1 ci0) (to \$1 ci1) (= (fare \$1) \$0))))	(lambda \$0 e (exists \$1 (and (oneway \$1) (from \$1 ci0) (to \$1 ci1) (= (fare \$1) \$0))))
(lambda \$0 e (and (ground_transport \$0) (to_city \$0 ci0)))	(lambda \$0 e (and (has_meal \$0) (to_city \$0 ci0)))
(lambda \$0 e (and (taxi \$0) (to_city \$0 ci0) (from_airport \$0 ap0)))	(lambda \$0 e (and (limousine \$0) (to_city \$0 ci0) (from_airport \$0 ap0)))
Binary Replacement	
(lambda \$0 e (and (flight \$0) (airline \$0 al0) (approx_departure_time \$0 ti0) (from \$0 ci0) (to \$0 ci1)))	(lambda \$0 e (and (flight \$0) (airline \$0 al0) (approx_arrival_time \$0 ti0) (from \$0 ci0) (to \$0 ci1)))
(lambda \$0 e (and (flight \$0) (from \$0 ci0) (to \$0 ci1) (day_return \$0 da0) (day_number_return \$0 dn0) (month_return \$0 mn0)))	(lambda \$0 e (and (flight \$0) (from \$0 ci0) (to \$0 ci1) (day_arrival \$0 da0) (day_number_arrival \$0 dn0) (month_arrival \$0 mn0)))
(lambda \$0 e (and (flight \$0) (airline \$0 al0) (stop \$0 ci0)))	(lambda \$0 e (and (flight \$0) (airline \$0 al0) (from \$0 ci0)))

Table 6: Examples of synthesized logical forms on ATIS.

Domain	Logical Forms	
Bas.	pre	(call SW.listValue (call SW.getProperty ((lambda s (call SW.filter (var s) (string position) (string !=) en.position.point_guard)) (call SW.domain (string player))) (string player)))
	new	(call SW.listValue (call SW.getProperty ((lambda s (call SW.filter (var s) (string position) (string !=) en.position.forward)) (call SW.domain (string player))) (string player)))
Blo.	pre	(call SW.listValue (call SW.filter (call SW.getProperty (call SW.singleton en.block) (string ! type)) (string shape) (string =) en.shape.pyramid))
	new	(call SW.listValue (call SW.filter (call SW.getProperty (call SW.singleton en.block) (string ! type)) (string shape) (string =) en.shape.cube))
Cal.	pre	(call SW.listValue (call SW.filter (call SW.getProperty (call SW.singleton en.location) (string ! type)) (call SW.reverse (string location)) (string =) en.meeting.weekly_standup))
	new	(call SW.listValue (call SW.filter (call SW.getProperty (call SW.singleton en.location) (string ! type)) (call SW.reverse (string location)) (string =) en.meeting.annual_review))
Hou.	pre	(call SW.listValue (call SW.filter (call SW.getProperty (call SW.singleton en.housing_unit) (string ! type)) (string housing_type) (string =) (call SW.concat en.housing.apartment en.housing.condo)))
	new	(call SW.listValue (call SW.filter (call SW.getProperty (call SW.singleton en.housing_unit) (string ! type)) (string housing_type) (string =) (call SW.concat en.housing.condo en.housing.apartment)))
Pub.	pre	(call SW.listValue (call SW.filter (call SW.getProperty (call SW.singleton en.article) (string ! type)) (string author) (string =) en.person.efron))
	new	(call SW.listValue (call SW.filter (call SW.getProperty (call SW.singleton en.article) (string ! type)) (string author) (string =) en.person.lakoff))
Rec.	pre	(call SW.listValue (call SW.getProperty en.recipe.rice_pudding (string cuisine)))
	new	(call SW.listValue (call SW.getProperty en.recipe.quiche (string cuisine)))
Res.	pre	(call SW.listValue (call SW.getProperty en.restaurant.thai_cafe (string neighborhood)))
	new	(call SW.listValue (call SW.getProperty en.restaurant.pizzeria_juno (string neighborhood)))
Soc.	pre	(call SW.listValue (call SW.getProperty ((lambda s (call SW.filter (var s) (string field_of_study) (string !=) en.field.computer_science)) (call SW.domain (string student))) (string student)))
	new	(call SW.listValue (call SW.getProperty ((lambda s (call SW.filter (var s) (string field_of_study) (string !=) en.field.history)) (call SW.domain (string student))) (string student)))

Table 7: Examples of synthesized logical forms on OVERNIGHT.