

# Sentence Rewriting for Semantic Parsing

Bo Chen Le Sun Xianpei Han Bo An

State Key Laboratory of Computer Sciences

Institute of Software, Chinese Academy of Sciences, China.

{chenbo, sunle, xianpei, anbo}@nfs.iscas.ac.cn

## Abstract

A major challenge of semantic parsing is the vocabulary mismatch problem between natural language and target ontology. In this paper, we propose a sentence rewriting based semantic parsing method, which can effectively resolve the mismatch problem by rewriting a sentence into a new form which has the same structure with its target logical form. Specifically, we propose two sentence-rewriting methods for two common types of mismatch: a dictionary-based method for 1-N mismatch and a template-based method for N-1 mismatch. We evaluate our sentence rewriting based semantic parser on the benchmark semantic parsing dataset – WEBQUESTIONS. Experimental results show that our system outperforms the base system with a 3.4% gain in F1, and generates logical forms more accurately and parses sentences more robustly.

## 1 Introduction

Semantic parsing is the task of mapping natural language sentences into logical forms which can be executed on a knowledge base (Zelle and Mooney, 1996; Zettlemoyer and Collins, 2005; Kate and Mooney, 2006; Wong and Mooney, 2007; Lu et al., 2008; Kwiatkowski et al., 2010). Figure 1 shows an example of semantic parsing. Semantic parsing is a fundamental technique of natural language understanding, and has been used in many applications, such as question answering (Liang et al., 2011; He et al., 2014; Zhang et al., 2016) and information extraction (Krishnamurthy and Mitchell, 2012; Choi et al., 2015; Parikh et al., 2015).

Semantic parsing, however, is a challenging

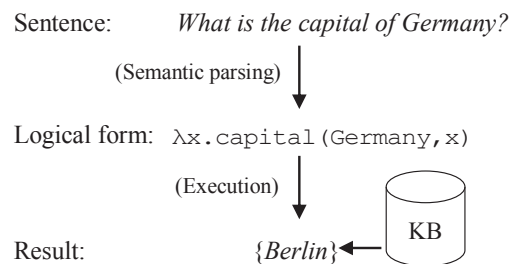


Figure 1: An example of semantic parsing.

task. Due to the variety of natural language expressions, the same meaning can be expressed using different sentences. Furthermore, because logical forms depend on the vocabulary of target-ontology, a sentence will be parsed into different logical forms when using different ontologies. For example, in below the two sentences  $s_1$  and  $s_2$  express the same meaning, and they both can be parsed into the two different logical forms  $lf_1$  and  $lf_2$  using different ontologies.

$s_1$  *What is the population of Berlin?*  
 $s_2$  *How many people live in Berlin?*  
 $lf_1$   $\lambda x.\text{population}(\text{Berlin}, x)$   
 $lf_2$   $\text{count}(\lambda x.\text{person}(x) \wedge \text{live}(x, \text{Berlin}))$

Based on the above observations, one major challenge of semantic parsing is the structural mismatch between a natural language sentence and its target logical form, which are mainly raised by the vocabulary mismatch between natural language and ontologies. Intuitively, if a sentence has the same structure with its target logical form, it is easy to get the correct parse, e.g., a semantic parser can easily parse  $s_1$  into  $lf_1$  and  $s_2$  into  $lf_2$ . On the contrary, it is difficult to parse a sentence into its logic form when they have different structures, e.g.,  $s_1 \rightarrow lf_2$  or  $s_2 \rightarrow lf_1$ .

To resolve the vocabulary mismatch problem,

(a) An example using traditional method

$s_0$ : <i>What is the name of Sonia Gandhis daughter?</i>
$l_0$ : $\lambda x. \text{child}(S.G., x)$
$r_0$ : { <i>Rahul Gandhi (Wrong answer), Priyanka Vadra</i> }

(b) An example using our method

$s_0$ : <i>What is the name of Sonia Gandhis daughter?</i>
$s_1$ : <i>What is the name of Sonia Gandhis female child?</i>
$l_1$ : $\lambda x. \text{child}(S.G., x) \wedge \text{gender}(x, \text{female})$
$r_1$ : { <i>Priyanka Vadra</i> }

Table 1: Examples of (a) sentences  $s_0$ , possible logical form  $l_0$  from traditional semantic parser, result  $r_0$  for the logical form  $l_0$ ; (b) possible sentence  $s_1$  from rewriting for the original sentence  $s_0$ , possible logical form  $l_1$  for sentence  $s_1$ , result  $r_1$  for  $l_1$ . *Rahul Gandhi* is a wrong answer, as he is the son of *Sonia Gandhi*.

this paper proposes a sentence rewriting approach for semantic parsing, which can rewrite a sentence into a form which will have the same structure with its target logical form. Table 1 gives an example of our rewriting-based semantic parsing method. In this example, instead of parsing the sentence “*What is the name of Sonia Gandhis daughter?*” into its structurally different logical form `childOf.S.G.^gender.female` directly, our method will first rewrite the sentence into the form “*What is the name of Sonia Gandhis female child?*”, which has the same structure with its logical form, then our method will get the logical form by parsing this new form. In this way, the semantic parser can get the correct parse more easily. For example, the parse obtained through traditional method will result in the wrong answer “*Rahul Gandhi*”, because it cannot identify the vocabulary mismatch between “*daughter*” and `child^female`<sup>1</sup>. By contrast, by rewriting “*daughter*” into “*female child*”, our method can resolve this vocabulary mismatch.

Specifically, we identify two common types of vocabulary mismatch in semantic parsing:

1. 1-N mismatch: a simple word may correspond to a compound formula. For example, the word “*daughter*” may correspond to the compound formula `child^female`.
2. N-1 mismatch: a logical constant may correspond to a complicated natural language expression, e.g., the formula `population` can be expressed using many phrases such as “*how many people*” and “*live in*”.

<sup>1</sup>In this paper, we may simplify logical forms for readability, e.g., `female` for `gender.female`.

To resolve the above two vocabulary mismatch problems, this paper proposes two sentence rewriting algorithms: One is a dictionary-based sentence rewriting algorithm, which can resolve the 1-N mismatch problem by rewriting a word using its explanation in a dictionary. The other is a template-based sentence rewriting algorithm, which can resolve the N-1 mismatch problem by rewriting complicated expressions using paraphrase template pairs.

Given the generated rewritings of a sentence, we propose a ranking function to jointly choose the optimal rewriting and the correct logical form, by taking both the rewriting features and the semantic parsing features into consideration.

We conduct experiments on the benchmark WEBQUESTIONS dataset (Berant et al., 2013). Experimental results show that our method can effectively resolve the vocabulary mismatch problem and achieve accurate and robust performance.

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 describes our sentence rewriting method for semantic parsing. Section 4 presents the scoring function which can jointly ranks rewritings and logical forms. Section 5 discusses experimental results. Section 6 concludes this paper.

## 2 Related Work

Semantic parsing has attracted considerable research attention in recent years. Generally, semantic parsing methods can be categorized into synchronous context free grammars (SCFG) based methods (Wong and Mooney, 2007; Arthur et al., 2015; Li et al., 2015), syntactic structure based methods (Ge and Mooney, 2009; Reddy et al., 2014; Reddy et al., 2016), combinatory categorical grammars (CCG) based methods (Zettlemoyer and Collins, 2007; Kwiatkowski et al., 2010; Kwiatkowski et al., 2011; Krishnamurthy and Mitchell, 2014; Wang et al., 2014; Artzi et al., 2015), and dependency-based compositional semantics (DCS) based methods (Liang et al., 2011; Berant et al., 2013; Berant and Liang, 2014; Berant and Liang, 2015; Pasupat and Liang, 2015; Wang et al., 2015).

One major challenge of semantic parsing is how to scale to open-domain situation like Freebase and Web. A possible solution is to learn lexicons from large amount of web text and a knowledge base using a distant supervised method (Krishna-

murthy and Mitchell, 2012; Cai and Yates, 2013a; Berant et al., 2013). Another challenge is how to alleviate the burden of annotation. A possible solution is to employ distant-supervised techniques (Clarke et al., 2010; Liang et al., 2011; Cai and Yates, 2013b; Artzi and Zettlemoyer, 2013), or unsupervised techniques (Poon and Domingos, 2009; Goldwasser et al., 2011; Poon, 2013).

There were also several approaches focused on the mismatch problem. Kwiatkowski et al. (2013) addressed the ontology mismatch problem (i.e., two ontologies using different vocabularies) by first parsing a sentence into a domain-independent underspecified logical form, and then using an ontology matching model to transform this underspecified logical form to the target ontology. However, their method is still hard to deal with the 1-N and the N-1 mismatch problems between natural language and target ontologies. Berant and Liang (2014) addressed the structure mismatch problem between natural language and ontology by generating a set of canonical utterances for each candidate logical form, and then using a paraphrasing model to rerank the candidate logical forms. Their method addresses mismatch problem in the reranking stage, cannot resolve the mismatch problem when constructing candidate logical forms. Compared with these two methods, we approach the mismatch problem in the parsing stage, which can greatly reduce the difficulty of constructing the correct logical form, through rewriting sentences into the forms which will be structurally consistent with their target logic forms.

Sentence rewriting (or paraphrase generation) is the task of generating new sentences that have the same meaning as the original one. Sentence rewriting has been used in many different tasks, e.g., used in statistical machine translation to resolve the word order mismatch problem (Collins et al., 2005; He et al., 2015). To our best knowledge, this paper is the first work to apply sentence rewriting for vocabulary mismatch problem in semantic parsing.

### 3 Sentence Rewriting for Semantic Parsing

As discussed before, the vocabulary mismatch between natural language and target ontology is a big challenge in semantic parsing. In this section, we describe our sentence rewriting algorithm for

Word	Logical Form	Wiktionary Explanation
<i>son</i>	child $\wedge$ male	<i>male child</i>
<i>actress</i>	actor $\wedge$ female	<i>female actor</i>
<i>father</i>	parent $\wedge$ male	<i>male parent</i>
<i>grandaprent</i>	parent $\wedge$ parent	<i>parent of one's parent</i>
<i>brother</i>	sibling $\wedge$ male	<i>male sibling</i>

Table 2: Several examples of words, their logical forms and their explanations in Wiktionary.

solving the mismatch problem. Specifically, we solve the 1-N mismatch problem by dictionary-based rewriting and solve the N-1 mismatch problem by template-based rewriting. The details are as follows.

#### 3.1 Dictionary-based Rewriting

In the 1-N mismatch case, a word will correspond to a compound formula, e.g., the target logical form of the word “*daughter*” is `child $\wedge$ female` (Table 2 has more examples).

To resolve the 1-N mismatch problem, we rewrite the original word (“*daughter*”) into an expression (“*female child*”) which will have the same structure with its target logical form (`child $\wedge$ female`). In this paper, we rewrite words using their explanations in a dictionary. This is because each word in a dictionary will be defined by a detailed explanation using simple words, which often will have the same structure with its target formula. Table 2 shows how the vocabulary mismatch between a word and its logical form can be resolved using its dictionary explanation. For instance, the word “*daughter*” is explained as “*female child*” in Wiktionary, which has the same structure as `child $\wedge$ female`.

In most cases, only common nouns will result in the 1-N mismatch problem. Therefore, in order to control the size of rewritings, this paper only rewrite the common nouns in a sentence by replacing them with their dictionary explanations. Because a sentence usually will not contain too many common nouns, the size of candidate rewritings is thus controllable. Given the generated rewritings of a sentence, we propose a sentence selection model to choose the best rewriting using multiple features (See details in Section 4).

Table 3 shows an example of the dictionary-based rewriting. In Table 3, the example sentence *s* contains two common nouns (“*name*” and “*daughter*”), therefore we will generate three rewritings  $r_1$ ,  $r_2$  and  $r_3$ . Among these rewritings,

$s$ : What is the name of Sonia Gandhis daughter?
$r_1$ : What is the reputation of Sonia Gandhis daughter?
$r_2$ : What is the name of Sonia Gandhis female child?
$r_3$ : What is the reputation of Sonia Gandhis female child?

Table 3: An example of the dictionary-based sentence rewriting.

the candidate rewriting  $r_2$  is what we expected, as it has the same structure with the target logical form and doesn't bring extra noise (i.e., replacing "name" with its explanation "reputation").

For the dictionary used in rewriting, this paper uses Wiktionary. Specifically, given a word, we use its "Translations" part in the Wiktionary as its explanation. Because most of the 1-N mismatch are caused by common nouns, we only collect the explanations of common nouns. Furthermore, for polysomic words which have several explanations, we only use their most common explanations. Besides, we ignore explanations whose length are longer than 5.

### 3.2 Template-based Rewriting

In the N-1 mismatch case, a complicated natural language expression will be mapped to a single logical constant. For example, considering the following mapping from the natural language sentence  $s$  to its logical form  $lf$  based on Freebase ontology:

$s$ : How many people live in Berlin?  
 $lf$ :  $\lambda x. \text{population}(\text{Berlin}, x)$

where the three words: "how many" (count), "people" (people) and "live in" (live) will map to the predicate `population` together. Table 4 shows more N-1 examples.

Expression	Logical constant
<i>how many, people, live in</i>	<code>population</code>
<i>how many, people, visit, annually</i>	<code>annual-visit</code>
<i>what money, use</i>	<code>currency</code>
<i>what school, go to</i>	<code>education</code>
<i>what language, speak, officially</i>	<code>official-language</code>

Table 4: Several N-1 mismatch examples.

To resolve the N-1 mismatch problem, we propose a template rewriting algorithm, which can rewrite a complicated expression into its simpler form. Specifically, we rewrite sentences based on a set of paraphrase template pairs  $P = \{(t_{i1}, t_{i2}) | i = 1, 2, \dots, n\}$ , where each template  $t$

Template 1	Template 2
<i>How many people live in \$y</i>	<i>What is the population of \$y</i>
<i>What money in \$y is used</i>	<i>What is the currency of \$y</i>
<i>What school did \$y go to</i>	<i>What is the education of \$y</i>
<i>What language does \$y speak officially</i>	<i>What is the official language of \$y</i>

Table 5: Several examples of paraphrase template pairs.

is a sentence with an argument slot  $\$y$ , and  $t_{i1}$  and  $t_{i2}$  are paraphrases. In this paper, we only consider single-slot templates. Table 5 shows several paraphrase template pairs.

Given the template pair database and a sentence, our template-based rewriting algorithm works as follows:

1. Firstly, we generate a set of candidate templates  $ST = \{st_1, st_2, \dots, st_n\}$  of the sentence by replacing each named entity within it by "\$y". For example, we will generate template "How many people live in \$y" from the sentence "How many people live in Berlin".
2. Secondly, using the paraphrase template pair database, we retrieve all possible rewriting template pairs  $(t_1, t_2)$  with  $t_1 \in ST$ , e.g., we can retrieve template pair ("How many people live there in \$y", "What is the population of \$y" for  $t_2$ ) using the above  $ST$ .
3. Finally, we get the rewritings by replacing the argument slot "\$y" in template  $t_2$  with the corresponding named entity. For example, we get a new candidate sentence "What is the population of Berlin" by replacing "\$y" in  $t_2$  with Berlin. In this way we can get the rewriting we expected, since this rewriting will match its target logical form `population(Berlin)`.

To control the size and measure the quality of rewritings using a specific template pair, we also define several features and the similarity between template pairs (See Section 4 for details).

To build the paraphrase template pair database, we employ the method described in Fader et al. (2014) to automatically collect paraphrase template pairs. Specifically, we use the WikiAnswers paraphrase corpus (Fader et al., 2013), which contains 23 million question-clusters, and all ques-

<i>How many people live in chembakolli?</i> <i>How many people is in chembakolli?</i> <i>How many people live in chembakolli india?</i> <i>How many people live there chembakolli?</i> <i>How many people live there in chembakolli?</i> <i>What is the population of Chembakolli india?</i>
<i>What currency is used on St Lucia?</i> <i>What is st lucia money?</i> <i>What is the money used in st lucia?</i> <i>What kind of money did st lucia have?</i> <i>What money do st Lucia use?</i> <i>Which money is used in St Lucia?</i>

Table 6: Two paraphrase clusters from the WikiAnswers corpus.

tions in the same cluster express the same meaning. Table 6 shows two paraphrase clusters from the WikiAnswers corpus. To build paraphrase template pairs, we first replace the shared noun words in each cluster with the placeholder “\$y”, then each two templates in a cluster will form a paraphrase template pair. To filter out noisy template pairs, we only retain salient paraphrase template pairs whose co-occurrence count is larger than 3.

## 4 Sentence Rewriting based Semantic Parsing

In this section we describe our semantic rewriting based semantic parsing system. Figure 2 presents the framework of our system. Given a sentence, we first rewrite it into a set of new sentences, then we generate candidate logical forms for each new sentence using a base semantic parser, finally we score all logical forms using a scoring function and output the best logical form as the final result. In following, we first introduce the used base semantic parser, then we describe the proposed scoring function.

### 4.1 Base Semantic Parser

In this paper, we produce logical forms for each sentence rewritings using an agenda-based semantic parser (Berant and Liang, 2015), which is based on the lambda-DCS proposed by Liang (2013). For parsing, we use the lexicons and the grammars released by Berant et al. (2013), where lexicons are used to trigger unary and binary predicates, and grammars are used to conduct logical forms. The only difference is that we also use the composition rule to make the parser can handle complicated questions involving two binary predicates, e.g., `child.obama^gender.female`.

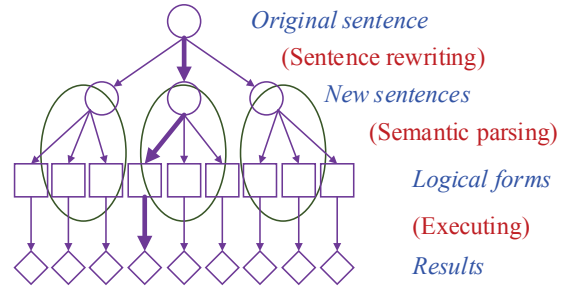


Figure 2: The framework of our sentence rewriting based semantic parsing.

For model learning and sentence parsing, the base semantic parser learned a scoring function by modeling the policy as a log-linear distribution over (partial) agenda derivations  $Q$ :

$$p_{\theta}(a|s) = \frac{\exp\{\phi(a)^T \theta\}}{\sum_{a' \in A} \exp\{\phi(a')^T \theta\}} \quad (1)$$

The policy parameters are updated as follows:

$$\theta \leftarrow \theta + \eta R(h_{target}) \sum_{t=1}^T \delta(h_{target}) \quad (2)$$

$$\begin{aligned} \delta_t(h) &= \nabla_{\theta} \log p_{\theta}(a_t|s_t) \\ &= \phi(a_t) - E_{p_{\theta}(a'_t|s_t)}[\phi(a'_t)] \end{aligned} \quad (3)$$

The reward function  $R(h)$  measures the compatibility of the resulting derivation, and  $\eta$  is the learning rate which is set using the AdaGrad algorithm (Duchi et al., 2011). The target history  $h_{target}$  is generated from the root derivation  $d^*$  with highest reward out of the  $K$  (beam size) root derivations, using local reweighting and history compression.

### 4.2 Scoring Function

To select the best semantic parse, we propose a scoring function which can take both sentence rewriting features and semantic parsing features into consideration. Given a sentence  $x$ , a generated rewriting  $x'$  and the derivation  $d$  of  $x'$ , we score them using follow function:

$$\begin{aligned} score(x, x', d) &= \theta \cdot \phi(x, x', d) \\ &= \theta_1 \cdot \phi(x, x') + \theta_2 \cdot \phi(x', d) \end{aligned}$$

This scoring function is decomposed into two parts: one for sentence rewriting  $-\theta_1 \cdot \phi(x, x')$  and the other for semantic parsing  $-\theta_2 \cdot \phi(x', d)$ . Following Berant and Liang (2015), we update the parameters  $\theta_2$  of semantic parsing features as the

---

**Input:** Q/A pairs  $\{(x_i, y_i) : i = 1 \dots n\}$ ; Knowledge base  $\mathcal{K}$ ; Number of sentences  $N$ ; Number of iterations  $T$ .

**Definitions:** The function  $REWRITING(x_i)$  returns a set of candidate sentences by applying sentence rewriting on sentence  $x$ ;  $PARSE(p_\theta, x)$  parses the sentence  $x$  based on current parameters  $\theta$ , using agenda-based parsing;  $CHOOSEORACLE(h_0)$  chooses the derivation with highest reward from the root of  $h_0$ ;  $CHOOSEORACLE(H_{target})$  chooses the derivation with highest reward from a set of derivations.  $CHOOSEORACLE(h_{target}^*)$  chooses the new sentence that results in derivation with highest reward.

**Algorithm:**

```

 $\theta_1 \leftarrow \mathbf{0}, \theta_2 \leftarrow \mathbf{0}$ 
for  $t = 1 \dots T, i = 1 \dots N$ :
   $X = REWRITING(x_i)$ 
  for each  $x'_i \in X$  :
     $h_0 \leftarrow PARSE(p_\theta, x'_i)$ 
     $d^* \leftarrow CHOOSEORACLE(h_0)$ 
     $h_{target} \leftarrow PARSE(p_\theta^{+cw}, x'_i)$ 
     $h_{target}^* \leftarrow CHOOSEORACLE(H_{target})$ 
     $x_i^* \leftarrow CHOOSEORACLE(h_{target}^*)$ 
     $\theta_2 \leftarrow \theta_2 + \eta R(h_{target}^*) \sum_{t=1}^T \delta(h_{target}^*)$ 
     $\theta_1 \leftarrow \theta_1 + \eta R(h_{target}^*) \delta(x_i, x_i^*)$ 

```

**Output:** Estimated parameters  $\theta_1$  and  $\theta_2$ .

---

Table 7: Our learning algorithm for parameter estimation from question-answer pairs.

same as (2). Similarly, the parameters  $\theta_1$  of sentence rewriting features are updated as follows:

$$\theta_1 \leftarrow \theta_1 + \eta R(h_{target}^*) \delta(x, x'^*)$$

$$\delta(x, x'^*) = \nabla \log p_{\theta_1}(x'^*|x)$$

$$= \phi(x, x'^*) - E_{p_{\theta_1}(x'|x)}[\phi(x, x')]$$

where the learning rate  $\eta$  is set using the same algorithm in Formula (2).

### 4.3 Parameter Learning Algorithm

To estimate the parameters  $\theta_1$  and  $\theta_2$ , our learning algorithm uses a set of question-answer pairs  $(x_i, y_i)$ . Following Berant and Liang (2015), our updates for  $\theta_1$  and  $\theta_2$  do not maximize reward nor the log-likelihood. However, the reward provides a way to modulate the magnitude of the updates. Specifically, after each update, our model results in making the derivation, which has the highest reward, to get a bigger score. Table 7 presents our learning algorithm.

### 4.4 Features

As described in Section 4.3, our model uses two kinds of features. One for the semantic parsing module which are simply the same features described in Berant and Liang (2015). One for the

sentence rewriting module these features are defined over the original sentence, the generated sentence rewritings and the final derivations:

**Features for dictionary-based rewriting.** Given a sentence  $s_0$ , when the new sentence  $s_1$  is generated by replacing a word to its explanation  $w \rightarrow ex$ , we will generate four features: The first feature indicates the word replaced. The second feature indicates the replacement  $w \rightarrow ex$  we used. The final two features are the POS tags of the left word and the right word of  $w$  in  $s_0$ .

**Features for template-based rewriting.** Given a sentence  $s_0$ , when the new sentence  $s_1$  is generated through a template based rewriting  $t_1 \rightarrow t_2$ , we generate four features: The first feature indicates the template pair  $(t_1, t_2)$  we used. The second feature is the similarity between the sentence  $s_0$  and the template  $t_1$ , which is calculated using the word overlap between  $s_0$  and  $t_1$ . The third feature is the compatibility of the template pair, which is the pointwise mutual information (PMI) between  $t_1$  and  $t_2$  in the WikiAnswers corpus. The final feature is triggered when the target logical form only contains an atomic formula (or predicate), and this feature indicates the mapping from template  $t_2$  to the predicate  $p$ .

## 5 Experiments

In this section, we assess our method and compare it with other methods.

### 5.1 Experimental Settings

**Dataset:** We evaluate all systems on the benchmark WEBQUESTIONS dataset (Berant et al., 2013), which contains 5,810 question-answer pairs. All questions are collected by crawling the Google Suggest API, and their answers are obtained using Amazon Mechanical Turk. This dataset covers several popular topics and its questions are commonly asked on the web. According to Yao (2015), 85% of questions can be answered by predicting a single binary relation. In our experiments, we use the standard train-test split (Berant et al., 2013), i.e., 3,778 questions (65%) for training and 2,032 questions (35%) for testing, and divide the training set into 3 random 80%-20% splits for development.

Furthermore, to verify the effectiveness of our method on solving the vocabulary mismatch problem, we manually select 50 mismatch test examples from the WEBQUESTIONS dataset, where

all sentences have different structure with their target logical forms, e.g., “*Who is keyshia cole dad?*” and “*What countries have german as the official language?*”.

**System Settings:** In our experiments, we use the Freebase Search API for entity lookup. We load Freebase using Virtuoso, and execute logical forms by converting them to SPARQL and querying using Virtuoso. We learn the parameters of our system by making three passes over the training dataset, with the beam size  $K = 200$ , the dictionary rewriting size  $K_D = 100$ , and the template rewriting size  $K_T = 100$ .

**Baselines:** We compare our method with several traditional systems, including semantic parsing based systems (Berant et al., 2013; Berant and Liang, 2014; Berant and Liang, 2015; Yih et al., 2015), information extraction based systems (Yao and Van Durme, 2014; Yao, 2015), machine translation based systems (Bao et al., 2014), embedding based systems (Bordes et al., 2014; Yang et al., 2014), and QA based system (Bast and Haussmann, 2015).

**Evaluation:** Following previous work (Berant et al., 2013), we evaluate different systems using the fraction of correctly answered questions. Because golden answers may have multiple values, we use the average F1 score as the main evaluation metric.

## 5.2 Experimental Results

Table 8 provides the performance of all base-lines and our method. We can see that:

1. Our method achieved competitive performance: Our system outperforms all baselines and get the best F1-measure of 53.1 on WE-BQUESTIONS dataset.
2. Sentence rewriting is a promising technique for semantic parsing: By employing sentence rewriting, our system gains a 3.4% F1 improvement over the base system we used (Berant and Liang, 2015).
3. Compared to all baselines, our system gets the highest precision. This result indicates that our parser can generate more-accurate logical forms by sentence rewriting. Our system also achieves the second highest recall, which is a competitive performance. Interestingly, both the two systems with the highest recall (Bast and Haussmann, 2015; Yih et al.,

System	Prec.	Rec.	F1 (avg)
Berant et al., 2013	48.0	41.3	35.7
Yao and Van-Durme, 2014	51.7	45.8	33.0
Berant and Liang, 2014	40.5	46.6	39.9
Bao et al., 2014	–	–	37.5
Bordes et al., 2014a	–	–	39.2
Yang et al., 2014	–	–	41.3
Bast and Haussmann, 2015	49.8	60.4	49.4
Yao, 2015	52.6	54.5	44.3
Berant and Liang, 2015	50.5	55.7	49.7
Yih et al., 2015	52.8	<b>60.7</b>	52.5
Our approach	<b>53.7</b>	60.0	<b>53.1</b>

Table 8: The results of our system and recently published systems. The results of other systems are from either original papers or the standard evaluation web.

2015) rely on extra-techniques such as entity linking and relation matching.

**The effectiveness on mismatch problem.** To analyze the commonness of mismatch problem in semantic parsing, we randomly sample 500 questions from the training data and do manually analysis, we found that 12.2% out of the sampled questions have mismatch problems: 3.8% out of them have 1-N mismatch problem and 8.4% out of them have N-1 mismatch problem.

To verify the effectiveness of our method on solving the mismatch problem, we conduct experiments on the 50 mismatch test examples and Table 9 shows the performance. We can see that our system can effectively resolve the mismatch between natural language and target ontology: compared to the base system, our system achieves a significant 54.5% F1 improvement.

System	Prec.	Rec.	F1 (avg)
Base system	31.4	43.9	29.4
Our system	<b>83.3</b>	<b>92.3</b>	<b>83.9</b>

Table 9: The results on the 50 mismatch test dataset.

When scaling a semantic parser to open-domain situation or web situation, the mismatch problem will be more common as the ontology and language complexity increases (Kwiatkowski et al., 2013). Therefore we believe the sentence rewriting method proposed in this paper is an important technique for the scalability of semantic parser.

**The effect of different rewriting algorithms.** To analyze the contribution of different rewriting methods, we perform experiments using different sentence rewriting methods and the results are presented in Table 10. We can see that:

Method	Prec.	Rec.	F1 (avg)
base	49.8	55.3	49.1
+ dictionary SR (only)	51.6	57.5	50.9
+ template SR (only)	52.9	59.0	52.3
+ both	<b>53.7</b>	<b>60.0</b>	<b>53.1</b>

Table 10: The results of the base system and our systems on the 2032 test questions.

1. Both sentence rewriting methods improved the parsing performance, they resulted in 1.8% and 3.2% F1 improvements respectively<sup>2</sup>.
2. Compared with the dictionary-based rewriting method, the template-based rewriting method can achieve higher performance improvement. We believe this is because N-1 mismatch problem is more common in the WEBQUESTIONS dataset.
3. The two rewriting methods are good complementary of each other. The semantic parser can achieve a higher performance improvement when using these two rewriting methods together.

**The effect on improving robustness.** We found that the template-based rewriting method can greatly improve the robustness of the base semantic parser. Specially, the template-based method can rewrite similar sentences into a uniform template, and the (template, predicate) feature can provide additional information to reduce the uncertainty during parsing. For example, using only the uncertain alignments from the words “*people*” and “*speak*” to the two predicates `official_language` and `language_spoken`, the base parser will parse the sentence “*What does jamaican people speak?*” into the incorrect logical form `official_language.jamaican` in our experiments, rather than into the correct form `language_spoken.jamaican` (See the final example in Table 11). By exploiting the alignment from the template “*what language does \$y people speak*” to the predicate , our system can parse the above sentence correctly.

**The effect on OOV problem.** We found that the sentence rewriting method can also provide extra

<sup>2</sup>Our base system yields a slight drop in accuracy compared to the original system (Berant and Liang, 2015), as we parallelize the learning algorithm, and the order of the data for updating the parameter is different to theirs.

<b>O</b>	<i>Who is willow smith mom name?</i>
<b>R</b>	<i>Who is willow smith female parent name?</i>
<b>LF</b>	<code>parentOf.willow.smith^gender.female</code>
<b>O</b>	<i>Who was king henry viii son?</i>
<b>R</b>	<i>Who was king henry viii male child?</i>
<b>LF</b>	<code>childOf.king_henry^gender.male</code>
<b>O</b>	<i>What are some of the traditions of islam?</i>
<b>R</b>	<i>What is of the religion of islam?</i>
<b>LF</b>	<code>religionOf.islam</code>
<b>O</b>	<i>What does jamaican people speak?</i>
<b>R</b>	<i>What language does jamaican people speak?</i>
<b>LF</b>	<code>language_spoken.jamaica</code>

Table 11: Examples which our system generates more accurate logical form than the base semantic parser. **O** is the original sentence; **R** is the generated sentence from sentence rewriting (with the highest score for the model, including rewriting part and parsing part); **LF** is the target logical form.

profit for solving the OOV problem. Traditionally, if a sentence contains a word which is not covered by the lexicon, it will cannot be correctly parsed. However, with the help of sentence rewriting, we may rewrite the OOV words into the words which are covered by our lexicons. For example, in Table 11 the 3rd question “*What are some of the traditions of islam?*” cannot be correctly parsed as the lexicons dont cover the word “*tradition*”. Through sentence rewriting, we can generate a new sentence “*What is of the religion of islam?*”, where all words are covered by the lexicons, in this way the sentence can be correctly parsed.

### 5.3 Error Analysis

To better understand our system, we conduct error analysis on the parse results. Specifically, we randomly choose 100 questions which are not correctly answered by our system. We found that the errors are mainly raised by following four reasons (See Table 12 for detail):

Reason	#(Ratio)	Sample Example
Label issue	38	<i>What band was george clinton in?</i>
N-ary predicate( $n > 2$ )	31	<i>What year did the seahawks win the superbowl?</i>
Temporal clause	15	<i>Who was the leader of the us during wwii?</i>
Superlative	8	<i>Who was the first governor of colonial south carolina?</i>
Others	8	<i>What is arkansas state capitol?</i>

Table 12: The main reasons of parsing errors, the ratio and an example for each reason are also provided.



The first reason is the label issue. The main label issue is incompleteness, i.e., the answers of a question may not be labeled completely. For example, for the question “*Who does nolan ryan play for?*”, our system returns 4 correct teams but the golden answer only contain 2 teams. One another label issue is the error labels. For example, the gold answer of the question “*What state is barack obama from?*” is labeled as “*Illinois*”, however, the correct answer is “*Hawaii*”.

The second reason is the n-ary predicate problem ( $n > 2$ ). Currently, it is hard for a parser to conduct the correct logical form of n-ary predicates. For example, the question “*What year did the seahawks win the superbowl?*” describes an n-ary championship event, which gives the championship and the champion of the event, and expects the season. We believe that more research attentions should be given on complicated cases, such as the n-ary predicates parsing.

The third reason is temporal clause. For example, the question “*Who did nasri play for before arsenal?*” contains a temporal clause “*before*”. We found temporal clause is complicated and makes it strenuous for the parser to understand the sentence.

The fourth reason is superlative case, which is a hard problem in semantic parsing. For example, to answer “*What was the name of henry viii first wife?*”, we should choose the first one from a list ordering by time. Unfortunately, it is difficult for the current parser to decide what to be ordered and how to order.

There are also many other miscellaneous error cases, such as spelling error in the question, e.g., “*capitol*” for “*capital*”, “*mary*” for “*marry*”.

## 6 Conclusions

In this paper, we present a novel semantic parsing method, which can effectively deal with the mismatch between natural language and target ontology using sentence rewriting. We resolve two common types of mismatch (i) one word in natural language sentence vs one compound formula in target ontology (1-N), (ii) one complicated expression in natural language sentence vs one formula in target ontology (N-1). Then we present two sentence rewriting methods, dictionary-based method for 1-N mismatch and template-based method for N-1 mismatch. The resulting system significantly outperforms the base system on the WEBQUES-

TIONS dataset.

Currently, our approach only leverages simple sentence rewriting methods. In future work, we will explore more advanced sentence rewriting methods. Furthermore, we also want to employ sentence rewriting techniques for other challenges in semantic parsing, such as the spontaneous, unedited natural language input, etc.

## Acknowledgments

We sincerely thank the reviewers for their valuable comments and suggestions. This work is supported by the National High Technology Development 863 Program of China under Grants no. 2015AA015405, and the National Natural Science Foundation of China under Grants no. 61433015, 612722324 and 61572477.

## References

- Philip Arthur, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. 2015. Semantic parsing of ambiguous input through paraphrasing and verification. *Transactions of the Association for Computational Linguistics*, 3:571–584.
- Yoav Artzi and Luke Zettlemoyer. 2013. Weakly supervised learning of semantic parsers for mapping instructions to actions. *Transactions of the Association for Computational Linguistics*, 1(1):49–62.
- Yoav Artzi, Kenton Lee, and Luke Zettlemoyer. 2015. Broad-coverage ccg semantic parsing with amr. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1699–1710, Lisbon, Portugal, September. Association for Computational Linguistics.
- Junwei Bao, Nan Duan, Ming Zhou, and Tiejun Zhao. 2014. Knowledge-based question answering as machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 967–976, Baltimore, Maryland, June. Association for Computational Linguistics.
- Hannah Bast and Elmar Haussmann. 2015. More accurate question answering on freebase. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015*, pages 1431–1440.
- Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1415–1425, Baltimore, Maryland, June. Association for Computational Linguistics.

- Jonathan Berant and Percy Liang. 2015. Imitation learning of agenda-based semantic parsers. *Transactions of the Association for Computational Linguistics*, 3:545–558.
- 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, Seattle, Washington, USA, October. Association for Computational Linguistics.
- Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. Question answering with subgraph embeddings. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 615–620, Doha, Qatar, October. Association for Computational Linguistics.
- Qingqing Cai and Alexander Yates. 2013a. Large-scale semantic parsing via schema matching and lexicon extension. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 423–433, Sofia, Bulgaria, August. Association for Computational Linguistics.
- Qingqing Cai and Alexander Yates. 2013b. Semantic parsing freebase: Towards open-domain semantic parsing. In *Second Joint Conference on Lexical and Computational Semantics (\*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 328–338, Atlanta, Georgia, USA, June. Association for Computational Linguistics.
- Eunsol Choi, Tom Kwiatkowski, and Luke Zettlemoyer. 2015. Scalable semantic parsing with partial ontologies. 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 1311–1320, Beijing, China, July. Association for Computational Linguistics.
- James Clarke, Dan Goldwasser, Ming-Wei Chang, and Dan Roth. 2010. Driving semantic parsing from the world’s response. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 18–27, Uppsala, Sweden, July. Association for Computational Linguistics.
- Michael Collins, Philipp Koehn, and Ivona Kucerova. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL’05)*, pages 531–540, Ann Arbor, Michigan, June. Association for Computational Linguistics.
- John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159, July.
- Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2013. Paraphrase-driven learning for open question answering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1608–1618, Sofia, Bulgaria, August. Association for Computational Linguistics.
- Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2014. Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’14*, pages 1156–1165, New York, NY, USA. ACM.
- Ruifang Ge and Raymond Mooney. 2009. Learning a compositional semantic parser using an existing syntactic parser. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 611–619, Suntec, Singapore, August. Association for Computational Linguistics.
- Dan Goldwasser, Roi Reichart, James Clarke, and Dan Roth. 2011. Confidence driven unsupervised semantic parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1486–1495, Portland, Oregon, USA, June. Association for Computational Linguistics.
- Shizhu He, Kang Liu, Yuanzhe Zhang, Liheng Xu, and Jun Zhao. 2014. Question answering over linked data using first-order logic. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1092–1103, Doha, Qatar, October. Association for Computational Linguistics.
- He He, Alvin Grissom II, John Morgan, Jordan Boyd-Graber, and Hal Daumé III. 2015. Syntax-based rewriting for simultaneous machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 55–64, Lisbon, Portugal, September. Association for Computational Linguistics.
- Rohit J. Kate and Raymond J. Mooney. 2006. Using string-kernels for learning semantic parsers. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 913–920, Sydney, Australia, July. Association for Computational Linguistics.
- Jayant Krishnamurthy and Tom 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, Jeju Island, Korea, July. Association for Computational Linguistics.

- Jayant Krishnamurthy and Tom M. Mitchell. 2014. Joint syntactic and semantic parsing with combinatory categorial grammar. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1188–1198, Baltimore, Maryland, June. Association for Computational Linguistics.
- Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2010. Inducing probabilistic CCG grammars from logical form with higher-order unification. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1223–1233, Cambridge, MA, October. Association for Computational Linguistics.
- 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, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.
- Tom Kwiatkowski, Eunsol Choi, Yoav Artzi, and Luke Zettlemoyer. 2013. Scaling semantic parsers with on-the-fly ontology matching. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1545–1556, Seattle, Washington, USA, October. Association for Computational Linguistics.
- Junhui Li, Muhua Zhu, Wei Lu, and Guodong Zhou. 2015. Improving semantic parsing with enriched synchronous context-free grammar. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1455–1465, Lisbon, Portugal, September. Association for Computational Linguistics.
- Percy Liang, Michael Jordan, and Dan Klein. 2011. Learning dependency-based compositional semantics. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 590–599, Portland, Oregon, USA, June. Association for Computational Linguistics.
- P. Liang. 2013. Lambda dependency-based compositional semantics. *arXiv preprint arXiv:1309.4408*.
- 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 2008 Conference on Empirical Methods in Natural Language Processing*, pages 783–792, Honolulu, Hawaii, October. Association for Computational Linguistics.
- Ankur P. Parikh, Hoifung Poon, and Kristina Toutanova. 2015. Grounded semantic parsing for complex knowledge extraction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 756–766, Denver, Colorado, May–June. Association for Computational Linguistics.
- P. Pasupat and P. Liang. 2015. Compositional semantic parsing on semi-structured tables. In *Association for Computational Linguistics (ACL)*.
- Hoifung Poon and Pedro Domingos. 2009. Unsupervised semantic parsing. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1–10, Singapore, August. Association for Computational Linguistics.
- Hoifung Poon. 2013. Grounded unsupervised semantic parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 933–943, Sofia, Bulgaria, August. Association for Computational Linguistics.
- Siva Reddy, Mirella Lapata, and Mark Steedman. 2014. Large-scale semantic parsing without question-answer pairs. *Transactions of the Association for Computational Linguistics*, 2:377–392.
- Siva Reddy, Oscar Täckström, Michael Collins, Tom Kwiatkowski, Dipanjan Das, Mark Steedman, and Mirella Lapata. 2016. Transforming Dependency Structures to Logical Forms for Semantic Parsing. *Transactions of the Association for Computational Linguistics*, 4:127–140.
- 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, October. 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, Beijing, China, July. Association for Computational Linguistics.
- 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, Prague, Czech Republic, June. Association for Computational Linguistics.
- Min-Chul Yang, Nan Duan, Ming Zhou, and Hae-Chang Rim. 2014. Joint relational embeddings for knowledge-based question answering. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 645–650, Doha, Qatar, October. Association for Computational Linguistics.

- Xuchen Yao and Benjamin Van Durme. 2014. Information extraction over structured data: Question answering with freebase. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 956–966, Baltimore, Maryland, June. Association for Computational Linguistics.
- Xuchen Yao. 2015. Lean question answering over freebase from scratch. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 66–70, Denver, Colorado, June. Association for Computational Linguistics.
- Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base. 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 1321–1331, Beijing, China, July. Association for Computational Linguistics.
- John M. Zelle and Raymond J. Mooney. 1996. Learning to parse database queries using inductive logic programming. In *AAAI/IAAI*, pages 1050–1055, Portland, OR, August. AAAI Press/MIT Press.
- Luke S. Zettlemoyer and Michael Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *UAI '05, Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence, Edinburgh, Scotland, July 26-29, 2005*, pages 658–666.
- 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)*, pages 678–687, Prague, Czech Republic, June. Association for Computational Linguistics.
- Yuanzhe Zhang, Shizhu He, Kang Liu, and Jun Zhao. 2016. A joint model for question answering over multiple knowledge bases. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 3094–3100.