Improving Semantic Parsing with Enriched Synchronous Context-Free Grammar

Junhui Li[1] **Muhua Zhu**[2] **Wei Lu**[3] **Guodong Zhou**[1]

[1]Natural Language Processing Lab, Soochow University, China

`{lijunhui, gdzhou}@suda.edu.cn`

[2] Alibaba Inc., Hangzhou, China

`muhua.zmh@alibaba-inc.com`

[3] Information Systems Technology and Design,

Singapore University of Technology and Design

`luwei@sutd.edu.sg`

Abstract

Semantic parsing maps a sentence in natural language into a structured meaning representation. Previous studies show that semantic parsing with synchronous context-free grammars (SCFGs) achieves favorable performance over most other alternatives. Motivated by the observation that the performance of semantic parsing with SCFGs is closely tied to the translation rules, this paper explores extending translation rules with high quality and increased coverage in three ways. First, we introduce structure informed non-terminals, better guiding the parsing in favor of well formed structure, instead of using a uninformed non-terminal in SCFGs. Second, we examine the difference between word alignments for semantic parsing and statistical machine translation (SMT) to better adapt word alignment in SMT to semantic parsing. Finally, we address the unknown word translation issue via synthetic translation rules. Evaluation on the standard GeoQuery benchmark dataset shows that our approach achieves the state-of-the-art across various languages, including English, German and Greek.

1 Introduction

Semantic parsing, the task of mapping natural language (NL) sentences into a formal meaning representation language (MRL), has recently received a significant amount of attention with various models proposed over the past few years. Consider the NL sentence paired with its corresponding MRL in Figure 1(a). Semantic parsing can be

---

NL: What is the area of Seattle
MRL: answer(area_1(cityid('seattle', _)))

(a) before pre-processing

NL': what be the area of seattle
MRL': answer@1 area_1@1 cityid@2 seattle@s _@0

(b) after pre-processing

Figure 1: Example of a sentence pair in NL and MRL.

naturally viewed as a statistical machine translation (SMT) task, which translates a sentence in NL (i.e., the source language in SMT) into its meaning representation in MRL (i.e., the target language in SMT). Indeed, many attempts have been made to directly apply statistical machine translation (SMT) systems (or methodologies) to semantic parsing (Papineni et al., 1997; Macherey et al., 2001; Wong and Mooney, 2006; Andreas et al., 2013). However, although recent studies (Wong and Mooney, 2006; Andreas et al., 2013) show that semantic parsing with SCFGs, which form the basis of most existing statistical syntax-based translation models (Yamada and Knight, 2001; Chiang, 2007), achieves favorable results, this approach is still behind the most recent state-of-the-art. For details, please see performance comparison in Andreas et al. (2013) and Lu (2014).

The key issues behind the limited success of applying SMT systems directly to semantic parsing lie in the difference between semantic parsing and SMT: *MRL is not a real natural language with different properties from natural language*. First, MRL is machine-interpretable and thus strictly structured with the meaning representation in a nested structure of functions and arguments. Second, the two languages are intrinsically asymmetric since each token in MRL carries specific mean-

1455

ing[1] while this does not hold in NL since auxiliary words and some function words usually have no counterparts in MRL. Third and finally, the expressions in NL are more flexible with respect to lexicon selection and token ordering. For example, since sentences in NL *'could you tell me the states that utah borders'*, *'what states does utah border'*, and *'utah borders what states'* convey the same meaning, they should have the same expression in MRL.

Motivated by the above observations, we believe that semantic parsing with *standard* SMT components is not an ideal approach. Alternatively, this paper proposes an effective, yet simple way to enrich SCFG in hierarchical phrase-based SMT for better semantic parsing. Specifically, since the translation rules play a critical role in SMT, we explore to improve translation rule quality and increase its coverage in three ways. First, we enrich non-terminal symbols as to capture contextual and structured information. The enrichment of non-terminal symbols not only guides the translation in favor of well formed structures, but also is beneficial to translation. Second, we examine the difference between word alignments for semantic parsing and SMT to better adapt word alignment in SMT to semantic parsing. Third, unlike most existing SMT systems that keep unknown words untranslated and intact in translation, we exploit the translation of unknown words via synthetic translation rules. Evaluation on Geo-Query benchmark dataset shows that our approach obtains consistent improvement and achieves the state-of-the-art across various languages, including English, German and Greek.

2 Background: Semantic Parsing as Statistical Machine Translation

In this section, we present the framework of semantic parsing as SMT, which was proposed in Andreas et al. (2013).

Pre-Processing Various semantic formalisms have been considered for semantic parsing. Examples include the variable-free semantic representations (that is, the meaning representation for each utterance is tree-shaped), the lambda calculus expressions, and dependency-based compositional semantic representations. In this work, we specifi-

cally focus on the variable-free semantic representations, as shown in Figure 1. On the target side, we convert these meaning representations to series of strings similar to NL. To do so, we simply take a preorder traversal of every functional form, and label every function with the number of arguments it takes. Figure 1(b) shows an example of converted meaning representation, where each token is in the format of $A@B$ where A is the symbol while B is either s indicating that the symbol is a string or a number indicating the symbol's arity (constants, including strings, are treated as zero-argument functions).

On the source side, we perform stemming (for English and German) and lowercasing to overcome data sparseness.

Hereafter, we refer to the pre-processed NL and MRL as NL$'$ and MRL$'$ respectively.

Translation Given a corpus of NL$'$ sentences paired with MRL$'$, we learn a semantic parser by adopting a string-to-string translation system. Typical components in such a translation system include word alignments between the source and the target languages, translation rule extraction, language model learning, parameter tuning and decoding. For more details about each component, please refer to (Chiang, 2007). In the rest of this paper, we refer to the source language (side) as NL$'$, and the target language (side) as MRL$'$.

Post-Processing We convert MRL$'$ back into MRL by recovering parentheses and commas to reconstruct the corresponding tree structure in MRL. This can be easily done by examining each symbol's arity. It eliminates any possible ambiguity from the tree reconstruction: given any sequence of tokens in MRL$'$, we can always reconstruct the tree structure (if one exists). For those translations that can not be successfully converted, we call them *ill-formed* translations.

3 Semantic Parsing with Enriched SCFG

In this section, we present the details of our enriched SCFG for semantic parsing.

3.1 Enriched SCFG

In hierarchical phrase-based (HPB) translation models, synchronous rules take the form $X \rightarrow \langle \gamma, \alpha, \sim \rangle$, where X is the non-terminal symbol, γ and α are strings of lexical items and non-terminals in the source and target side respectively, and $\sim$ indicates the one-to-one cor-

---

[1]As seen in Section 2, delimiters, including parentheses and commas which do not carry any meaning will be removed in pre-processing and be recovered in post-processing

respondence between non-terminals in γ and α. From an aligned phrase pair *<state that border, state@1 next_to_2@1>* in Figure 2(a), for example, we can get a synchronous rule $X \rightarrow \langle$state $X_{\boxed{1}}$, state@1 $X_{\boxed{1}}\rangle$, where we use boxed indices to indicate which nonterminal occurrences are linked by $\sim$. The fact that SCFGs in HPB models contain only one type of non-terminal symbol[2] is responsible for ill-formed translation (e.g., *answer@1 state@1*). To this end, we enrich the non-terminals to capture the tree structure information, guiding the translation in favor of well-formed translations. The enrichment of non-terminals is two-fold: first, it can handle MRL with a nested structure to guarantee the well-formed translations; second, related studies in SMT have shown that introducing multiple non-terminal symbols in SCFGs benefits translation (Zollmann and Venugopal, 2006; Li et al., 2012).

Given a word sequence e_j^i from position i to position j in MRL$'$, we enrich the non-terminal symbol X to reflect the internal structure of the word sequence of e_j^i. A correct translation rule selection therefore not only maps source terminals into target terminals, but is both constrained and guided by structure information in the non-terminals. As mentioned earlier, we regard the nested structure in MRL$'$ as function-argument structure, where each function takes one or more arguments as input while its return serves as an argument to the outside function. As in Figure 1, function *cityid* holds two arguments and returns as an argument to function *area_1*. For a word sequence e_j^i, we examine its completeness, which is defined as:

Definition 1. *For word sequence e_j^i, it is regarded as complete if it satisfies 1) every function (if exists) meets its argument requirement; and 2) it can serve as **one** argument to another function.*

We use symbol C to label word sequences which are complete. For an incomplete word sequence, we examine 1) the number of arguments it requires on the right to be complete; and 2) the arity of a function it requires on the left to be complete. Then the sequence is labeled as $(C\backslash Fm)/An$, indicating it requires n arguments on the right and a function with m arities on the left. [3]

---

[2]In practice, non-terminal symbol S is used in glue rules. However, this is not relevant in the present discussion.

[3]This is similar to the naming convention in combinatory categorial grammar (CCG) (Steedman, 2000)

texas, stateid@1 texas@s
$C\rightarrow \langle$texas, stateid@1 texas@s$\rangle$
seattle, seattle@s _@0
$C\backslash F2\rightarrow \langle$seattle, seattle@s _@0$\rangle$
that border, next_to_2@1
$C/A1\rightarrow \langle$that border, next_to_2@1$\rangle$
state that border, state@1 next_to_2@1
$C/A1\rightarrow \langle$state $C/A1_{\boxed{1}}$, state@1 $C/A1_{\boxed{1}}\rangle$

(a) Examples of phrase pairs in enriched SCFG.

state that border , state@1 next_to_2@1
$C/A1\rightarrow \langle C/A1_{\boxed{1}}\,C/A1_{\boxed{2}},\ C/A1_{\boxed{1}}\,C/A1_{\boxed{2}}\rangle$
state that border texas have the highest population,
l.._one@1 p.._1@1 state@1 n..@1 s..@1 t..@s
$C\rightarrow \langle C_{\boxed{1}}\,C/A1_{\boxed{2}},\ C/A1_{\boxed{2}}\,C_{\boxed{1}}\rangle$

(b) Examples of glue rules in enriched SCFG.

Table 1: Example of translation rules in enriched SCFG, where underline and underwave indicate the first and the second phrases respectively.

Specifically, we omit $\backslash Fm$ and $/An$ if $m = 0$ and $n = 0$ respectively.[4]

Table 1(a) demonstrates examples of phrase pairs in our enriched SCFG. For instance, word sequence *stateid@1 texas@s* is complete, and thus labeled as C. Similarly, to be complete, word sequence *next_to_2@1* requires one argument on the right side, labeled as $C/A1$ accordingly.

When extracting translation rules from aligned datasets, we follow Chiang (2007) except that we use enriched non-terminal symbols rather than X. Each translation rule is associated with a set of translation model features $\{\phi_i\}$, including phrase translation probability $p(\alpha \mid \gamma)$ and its inverse $p(\gamma \mid \alpha)$, the lexical translation probability $p_{lex}(\alpha \mid \gamma)$ and its inverse $p_{lex}(\gamma \mid \alpha)$, and a rule penalty that learns the preference for longer or shorter derivations.

Inverted Glue Rules In SMT decoding (Chiang, 2007), if no rule (e.g., a rule whose left-hand side is X) can be applied or the length of the potential source span is larger than a pre-defined length (e.g., 10 as in Chiang (2007)), a glue rule (either $S \rightarrow \langle X_{\boxed{1}}, X_{\boxed{1}}\rangle$ or $S \rightarrow \langle S_{\boxed{1}}X_{\boxed{2}}, S_{\boxed{1}}X_{\boxed{2}}\rangle$) will be used to simply stitch two consequent translated phrases together in a *monotone* way. Although this will reduce computational and modeling challenges, it obviously prevents some reasonable translation derivations because in certain cases, the order of phrases may be inverted on the target

---

[4]If $m = 0$, it indicates that no function is needed.

side. In this work, we additionally use an inverted glue rule which combines two non-terminals in a *swapped* way. Each glue rule, either *straight* or *inverted*, contains only two non-terminal symbols and is associated with two features, including phrase translation probability $p(\alpha \mid \gamma)$, and a glue rule penalty. Table 1(b) shows examples of a straight and an inverted glue rules. Moreover, these glue rules can be applied to any two neighboring translation nodes if the non-terminal symbols are matched.

3.2 Word Alignment for Semantic Parsing

Word alignment is an essential step for rule extraction in SMT, where recognizing that *wo shi* in Chinese is a good translation for *I am* in English requires establishing a correspondence between *wo* and *I*, and between *shi* and *am*. In the SMT community, researchers have developed standard, proven alignment tools such as GIZA++ (Och and Ney, 2003), which can be used to train IBM Models 1-5. However, there is one fundamental problem with the IBM models (Brown et al., 1993): each word on one side can be traced back to exactly one particular on the other word (or the *null* token which indicates the word aligns to no word on the other side). Figure 2(a) shows an example of GIZA++ alignment output from source side to target side, from which we can see that each source word aligns to exactly one target word. While alignment of multiple target words to one source word is common in SMT, a trick is then to run IBM model training in both directions. Then two resulting word alignments can be symmetrized, for instance, taking the intersection or the union of alignment points of each alignment. For example, Figure 2(b) shows GIZA++ alignment output from target side to source side while Figure 2(c) shows the symmetrization result with widely used grow-diag-final-and strategy.

Although symmetrization of word alignments works for SMT, can it be applied to semantic parsing? There are reasons to be doubtful. Word alignment for semantic parsing differs from alignment for SMT in several important aspects, at least including:

1. It is intrinsically asymmetric: within the semantic formalism used in this paper, NL$'$ is often longer than MRL$'$, and commonly contains words which have no counterpart in MRL$'$.

2. Little training data is available. SMT alignment models are typically trained in unsupervised fashion, inducing lexical correspondences from massive quantities of sentence-aligned bitexts.

Consequently, the symmetrization of word alignments may not work perfectly for semantic parsing. According to word alignment in Figure 2(c), a phrase extractor will generate a phrase pair ⟨*have the highest, largest_one@1*⟩, which is non-intuitive. By contrast, a more useful and general phrase pair ⟨*highest, largest_one@1*⟩ is typically excluded because *largest_one@1* aligns to all of *have*, *the*, and *highest*. Similarly, another useful phrase pair ⟨*texas, texas@s*⟩ is prohibited since *texas* aligns to both *stateid@1* and *texas@s*.

Ideally a new semantic parsing aligner should be able to capture the semantic equivalence. Unfortunately we are not aware of any research on alignment for semantic parsing, possibly due to lack of a paucity of high quality, publicly available data from which to learn. Instead of developing new alignment algorithm for semantic parsing, we make use of all the alignments as shown in Figure 2. That is to say, we triple the training data with each sentence pair having three alignments, i.e., two alignments in both directions, and the symmetrization alignment.[5] The advantages include: first, considering more possible alignments would increase the phrase coverage, especially when the training data is little; second, including the alignment from both directions would alleviate the error propagation caused by mis-aligned stop words (e.g., *be*, *the* in NL$'$ and *stateid@1* in MRL$'$). As a result, the phrase extractor will include phrase pairs of both ⟨*highest, largest_one@1*⟩ and ⟨*texas, texas@s*⟩. Our experiment shows that using the combination of all the three alignments achieve better performance than using any one, or any combination of two. Moreover, we found that we could achieve comparable performance even with manual alignment.

---

[5]Combining multiple alignments from different alignment models usually improves translation performance in SMT (Tu et al., 2012). However, our preliminary experiments showed that this did not yield higher improvement in semantic parsing, which in turn also demonstrates the difference in alignments for semantic parsing and SMT.
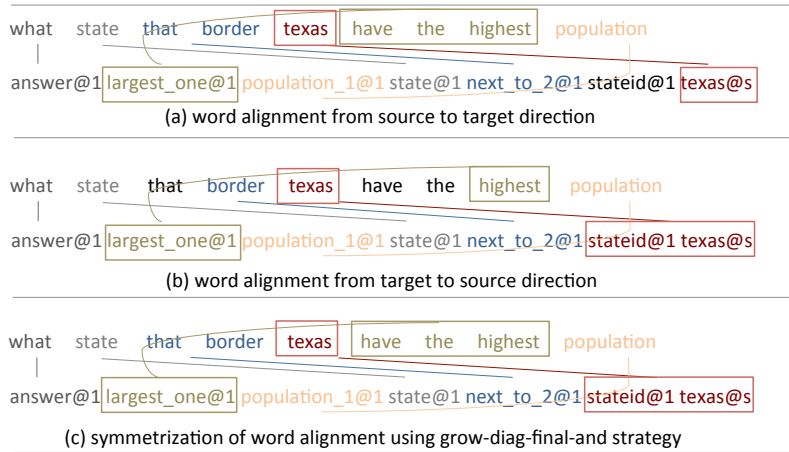
(a) word alignment from source to target direction

(b) word alignment from target to source direction

(c) symmetrization of word alignment using grow-diag-final-and strategy

Figure 2: Example of a sentence pair with different alignments.

3.3 Synthetic Translation Rules for Unknown Word Translation

Most NLP tasks face the problem of unknown words, especially if only little training data is available. For example, it is estimated that 5.7% sentences in the (English) test data in our experiments have unknown words. Unknown words usually remain intact in the translation in most machine translation systems (Koehn et al., 2007; Dyer et al., 2010), resulting in the fact that certain translations can not be converted back to tree structures. This indicates that in semantic parsing the translation of a word can be from two categories: 1) a token in MRL; or 2) *null* (i.e., not translated at all), we generate synthetic translation rules for unknown word translation.

As a baseline, we simply skip unknown words as Kwiatkowski et al. (2010) by adding translation rules that translate them to *null* in MRL'. Each such rule is accompanied with one feature indicating that it is a translation rule for unknown word.

Alternatively, taking advantage of publicly available resources, we generate synthetic translation rules for unknown words pivoted by their semantically close words. Algorithm 1 illustrates the process to generate synthetic translation rules for unknown word translation. Given an unknown word w_u, it generates its synthetic rules in two steps: 1) finding top n (e.g., 5 as in our experiments) close words via Word2Vec;[6] and 2) generating synthetic translation rules based on the close

---

Algorithm 1: Generating synthetic translation rules for unknown words

Input: Unknown word w_u in the source language
Source side training data vocabulary: W
Lexical translation tables T1 and T2 (two directions)

Output: Synthetic translation rule set R for w_u

1. **foreach** word w_i in W
2. $s_i = sim(w_u, w_i)$
3. get the top n words $WB = \{wb_1...wb_n\}$
    with the highest $\{s_i\}$
4. $R = \phi$
5. **foreach** wb_i in WB
6.   **foreach** t_j such $\langle wb_i, t_j \rangle$ in T1 and T2
7. $R \cup = generate\_rule(w_u, wb_i, t_j, T_1, T_2)$
8. **return** R

---

sim: returns the similarity between w_u and w_i.

$generate\_rule$: returns rule $\langle w_u, t_j \rangle$ with a feature indicating the similarity between w_u and wb_i, and two features indicating the lexical translation probabilities from wb_i to t_j and the way around.

---

words. Note that it may generate a synthetic rule with *null* at the target side since the lexical translation table derived from aligned training data contains translation to *null*. Each synthetic translation rule for unknown words is associated with three features returned from function $generate\_rule$.

4 Experimentation

In this section, we test our approach on the Geo-Query dataset, which is publicly available.

4.1 Experimental Settings

Data GeoQuery dataset consists of 880 questions paired with their corresponding tree structured semantic representations. Following the experimental setup in Jones et al. (2012), we use the 600 question pairs to train and tune our SMT de-

---

[6] It is available at http://code.google.com/p/word2vec/. We use Word2Vec rather than other linguistic resources like WordNet because the approach can be easily adopted to other languages only if there exists large monolingual data to train Word2Vec models.

coder, and evaluated on the remaining 280. Note that there is another version of GeoQuery dataset where the semantic representation is annotated with lambda calculus expressions and which is extensively studied (Zettlemoyer and Collins, 2005; Wong and Mooney, 2007; Liang et al., 2011; Kwiatkowski et al., 2013). Performance on the version of lambda calculus is higher than that on the tree structured version, however, the results obtained over the two versions are not directly comparable.

SMT Setting We use cdec (Dyer et al., 2010) as our HPB decoder. As mentioned above, 600 instances are used to train and tune our decoder. To get fair results, we split the 600 instances into 10 folds, each having 60 instances. Then for each fold, we use it as the tuning data while the other 540 instances and the NP list are used as training data.[7] We use IRSTLM toolkit (Federico et al., 2008) to train a 5-gram LM on the MRL' side of the training data, using modified Kneser-Ney smoothing. We use Mira (Chiang et al., 2008) to tune the parameters of the system to maximize BLEU (Papineni et al., 2002). When extracting translation rules from aligned training data, we include both *tight* and *untight* phrases.

Evaluation We use the standard evaluation criteria for evaluation by executing both the predicted MRL and the gold standard against the database and obtaining their respective answer. Specifically, we convert a translation from MRL' into MRL (if exists). The translation then is considered correct if and only if its MRL retrieves the same answers as the gold standard MRL (Jones et al., 2012), allowing for a fair comparison between our systems and previous works. As in Jones et al. (2012), we report accuracy, i.e. the percentage of translations with correct answers, and F1, i.e. the harmonic mean of precision (the proportion of correct answers out of translations with an answer) and recall (the proportion of correct answers out of all translations). In this section, we report our performance scores and analysis numbers averaged on our 10 SMT models.

---

[7]The NP list is from GeoQUery dataset in Jones et al. (2012), which contains MRs for every noun phrase that appears in the NL utterances of each language. As in Andreas et al. (2013), the NP list is included by appending all entries as extra training sentences with 50 times the weight of regular training examples, to ensure that they are learned as translation rules.

SCFG	alignment	Acc.	F1
non-enriched	src2tgt	75.0	82.5
	tgt2src	78.5	82.5
	gdfa	77.5	83.5
	src2tgt + tgt2src	81.4	85.0
	src2tgt + gdfa	77.1	83.1
	tgt2src + gdfa	80.6	83.9
	all	81.5	85.2
	gold	82.4	86.2
enriched	src2tgt	76.3	82.9
	tgt2src	82.0	85.2
	gdfa	78.9	83.9
	src2tgt + tgt2src	82.6	85.9
	src2tgt + gdfa	78.8	83.7
	tgt2src + gdfa	83.1	86.1
	all	82.9	86.1
	gold	84.1	87.1

Table 2: Performance of our (non-) enriched SCFG systems with different alignment settings.

4.2 Experimental Results

Table 2 shows the results of (non-) enriched SCFG systems over different alignment settings. In Table 2, *src2tgt* and *tgt2src* indicate alignment of source to target direction and alignment of target to source direction, respectively; *gdfa* indicates symmetrization of alignment with grow-diag-final-and strategy; *src2tgt+tgt2src* indicates doubling the training data with each sentence pair having both *src2tgt* and *tgt2src* alignments, similar for *src2tgt+gdfa* and *tgt2src+gdfa*; *all* indicates tripling the training data with each sentence pair having three alignments. Finally, *gold* indicates using gold alignment. [8]

Effect of Enriched SCFG From Table 2, we observe that enriched SCFG systems outperform non-enriched SCFG systems over all alignment settings, indicating the effect of enriching nonterminals. In particular for *tgt2src* alignment, it obtains improvements of 3.5% in accuracy and 2.7% in F1.

As mentioned earlier, the non-enriched SCFG system may result in ill-formed translations, which can not be converted back to tree structure. One natural way to overcome this issue, as in Andreas et al. (2013), would be to simply filter n-best translation till a well-formed one is found. However, we see very limited performance changes in accuracy and F1, suggesting that the effect of using n-best translation is very limited. For example, after using n-best translation, the non-enriched SCFG system with *all* alignment obtains 82.0 in accuracy (increased from 81.5) and 84.5 in

---

[8]We manually aligned sentence pairs in NL' and MRL'.

alignment	Rec.	Pre.	F1
src2tgt	96.3	68.8	80.3
tgt2src	90.0	77.0	83.0
gdfa	95.1	62.7	75.6

Table 3: Alignment performance.

F1 (reduced from 85.2).

Effect of Word Alignment With respect to the performance over different alignment settings, we have the following observations from Table 2:

- Semantic parsing is substantially sensitive to alignment. Surprisingly, *gdfa* alignment, which is widely adopted in SMT, is inferior to *tgt2src* alignment. As expected, *src2tgt* alignment achieves the worst performance.

- Thanks to the increased coverage, doubling the training data (e.g., rows of *src2tgt+tgt2src*, *src2tgt+gdfa*, and *tgt2src+gdfa*) usually outperforms its corresponding single alignment. Moreover, tripling the training data (e.g., rows of *all*) achieves slightly better performance than any way of doubling the training data. This is expected since the *gdfa* alignment actually comes from the alignments of *src2tgt* and *tgt2src*, thus doubling the training with *src2tgt* and *tgt2src* have already included most aligns in *gdfa* alignment.

- Our approach of tripling the training data achieves comparable performance to the one with gold alignment, suggesting that instead of developing a brand new algorithm for semantic parsing alignment, we can simply make use of GIZA++ alignment output.

In terms of the *src2tgt*, *tgt2src* and *gdfa* alignments, the trend of the results is consistent over both non-enriched and enriched SCFG systems: the systems with *tgt2src* alignment work best while the systems with *src2tgt* alignment work worst. Next we look at the non-enriched SCFG systems to explore the behavior differences among the three alignments.

We examine the alignment accuracy against the gold alignment on training data (except the NP list part). As shown in Table 3, *src2tgt* has the highest recall while *tgt2src* has the highest precision. This is partly due to: 1) In *src2tgt* alignment, each source word aligns to exactly one particular target word (or the *null* token), resulting in frequent

		enriched + gdfa	
		correct	wrong
non-enriched + gdfa	correct	211	6
	wrong	10	53
enriched + all	correct	215	17
	wrong	6	42

Table 4: Confusion matrices of three SMT systems on English test sentences.

alignment errors for source side words that have no counterpart in target side. For example, both words of *the* and *be* on source side, which play functional roles in NL, rather than semantic roles, align to 15 different target words. 2) Except for a few words on target side, including *stateid@1*, *all@0* which have strong occurrence patterns (e.g., *stateid@1* is always followed by a state name), each word has counterpart on source side.

As to have a clearer understanding on the individual contribution of using enriched non-terminals and multiple word alignments, Table 4 presents two confusion matrices which show numbers of sentences that are correctly/wrongly parsed by three SMT systems on English test sentences. It shows that, for example, 211 sentences are correctly parsed by both non-enriched and enriched SCFG systems with *gdfa* alignment. Moving from performance of the non-enriched SMT system with *gdfa* alignment to that of the enriched SMT system with *all* alignment, we observe that on average more than half of the improvement comes from using multiple word alignments, the rest from using enriched non-terminals.

Effect of Unknown Word Translation Since each of our SMT model is actually trained on 540 instances (plus the NP list), the rate of unknown words in the test data tends to be higher than that in a system trained with the whole 600 instances. Based on the system of enriched SCFG with *all* alignment, Table 5 shows the results of applying unknown word translation. It shows that translating all unknown words into *null* obtains 2.4 points in accuracy over the system without it (e.g., 85.3 *vs.* 82.9). However, the slight improvement in F1 (e.g., 86.3 *vs.* 86.1) suggests that there are many scenarios that translating unknown words into *null* is incorrect. Fortunately, our semantic approach is partially able to generate correct translation rules for those unknown words which have translation in MRL$'$. Actually, the effect of our approach is highly dependent on the quality of the close words found via Word2Vec. With a manual examination

System	Acc.	F1
No unknown word translation	82.9	86.1
null: baseline	85.3	86.3
semantic: ours	86.3	87.1

Table 5: Performance with unknown word translation.

on the test data, we found that 11 out of all 17 unknown words should be translated into a corresponding token in MRL. For 8 of them, the synthetic translation rule set returned by Algorithm 1 contains correct translation rules.

Effect across Different Languages We have also tested our approach on the same dataset with other three languages. Specifically, while we are not aware of public resources to looking for semantically close words in German, Greek and Thai, we translate unknown words into *null* for the three languages. Table 6 shows the performance over four different languages. It shows that our approach, including enriched SCFG, tripling training data with three alignments, and unknown word translation, obtains consistent improvement over the four languages.

Decoding Time Analysis We analyze the effect on the decoding time of our approach, which is closely related to the size of phrase tables. Firstly, splitting non-terminal X into enriched ones increases the size of phrase tables. [9] This is not surprising since a phrase with non-terminal X (e.g., *the X* on the source side) may be further specified as multiple phrases with various non-terminals (e.g., *the C*, *the C/A1*, etc.). As a result, the average number of phrases per sentence (in English test data, hereafter) increases from 453 to 893 while the decoding time of the SMT decoder increases from 0.11 seconds to 0.19 seconds per sentence on average. Secondly, using multiple alignments also leads to larger phrase tables. This is illustrated by the increase of average number of phrases per sentence from 893 to 2055 while the decoding time moves from 0.19 seconds to 0.38 seconds per sentence on average. Finally, finding similar words via Word2Vec, however, is quite fast since this is bounded by the vocabulary size of our training set. Thanks to the small size of unknown words, adding unknown word translation rules has a very limited impact on the size of phrase table, consequently negligible changes on decoding time.

---

[9] In cdec, we generate a phrase table for each sentence.

5 Related Work

While there has been substantial work on semantic parsing, we focus our discussions on several approaches (e.g., SCFG approach, hybrid tree approach, and others approaches) that focus on the variable-free semantic representations.

WASP (Wong and Mooney, 2006) was strongly influenced by SMT techniques. Although WASP was also using multiple non-terminal symbols in SCFG to guarantee well-formed translations, our work differs from theirs in at least three ways. First, we use a different inventory of non-terminal symbols from theirs which was derived from MRL parses in the GeoQuery dataset. Second, to avoid the issues caused by word alignment between NL and MRL, we triple training data with each sentence pair having multiple alignments. However, WASP used a sequence of productions to represent MRL before running GIZA++. Third, we use typical features in HPB SMT (e.g., phrase translation probabilities, lexical translation probabilities, language model feature, etc.) while WASP used rule identity features. SMT-SemParse (Andreas et al., 2013) adapted standard SMT components for semantic parsing. The present work is based on theirs with all the extensions detailed in Section 3.

HYBRIDTREE+ (Lu et al., 2008) learned a synchronous generative model which simultaneously generated a NL sentence and an MRL tree. tsVB (Jones et al., 2012) used tree transducers, which were similar to the hybrid tree structures, to learn a generative process under a Bayesian framework. RHT (Lu, 2014) defined distributions over *relaxed hybrid tree* structures that jointly represented both sentences and semantics. Most recently, f-RHT (Lu, 2015) introduced constrained semantic forests to improve RHT model.

SCISSOR (Ge and Mooney, 2005) augmented syntactic parse tree with semantic information and then performed integrated semantic and syntactic parsing to NL sentences. KRISP (Mooney, 2006) used string classifiers to label substrings of an NL with entities from the meaning representation. UBL (Kwiatkowski et al., 2010) performed semantic parsing with an automatically-induced CCG lexicon.

Table 7 shows the evaluation results of our system as well as those of several other comparable related works which share the same experiment setup as ours. We can observe from Table 7 that semantic parsing with SMT components gives

System	English		German		Greek		Thai	
	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
non-enriched + gdfa	77.5	83.5	66.0	74.9	65.6	74.1	65.4	72.4
non-enriched + all	81.5	85.2	72.1	76.8	75.2	80.5	72.7	76.4
enriched + gdfa	78.9	83.9	66.7	74.6	67.8	76.1	68.5	74.1
enriched + all	82.9	86.1	75.4	79.5	76.5	81.2	75.2	77.9
enriched + all + unknown word translation	86.3	87.1	79.1	80.3	80.5	81.6	76.3	77.9

Table 6: Performance for the multilingual GeoQuery test set.

System	English		German		Greek		Thai	
	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
WASP	71.1	77.7	65.7	74.9	70.7	78.6	71.4	75.0
SMT-SemParse	80.5	81.8	68.9	71.8	69.1	72.3	70.4	70.7
HYBRIDTREE+	76.8	81.0	62.1	68.5	69.3	74.6	73.6	76.7
tsVB	79.3	79.3	74.6	74.6	75.4	75.4	78.2	78.2
RHT	83.6	83.6	74.3	74.3	78.2	78.2	79.3	79.3
f-RHT	**86.8**	86.8	75.7	75.7	79.3	79.3	**80.7**	**80.7**
UBL	82.1	82.1	73.6	73.7	75.0	75.0	66.4	66.4
this work	86.3	**87.1**	**79.1**	**80.3**	**80.5**	**81.6**	76.3	77.9

Table 7: Performance comparison for the multilingual GeoQuery test set. The performance of WASP, HYBRIDTREE+, tsVB and UBL is taken from Jones et al. (2012).

competitive performance when all the extensions (described in Section 3) are used. Specifically, it significantly outperforms the semantic parser with standard SMT components (Andreas et al., 2013). Our approach reports the best accuracy and F1 scores on English, German, and Greek. While we are able to obtain improvement on Thai, the performance is still lower than those of RHT and TREETRANS. This is probably because of the low quality of word alignment output between this Asian language and MRL.

6 Conclusion and Future Work

In this paper, we have presented an enriched SCFG approach for semantic parsing which realizes the potential of the SMT approach. The performance improvement is contributed from the extension of translation rules with informative symbols and increased coverage. Such an extension share a similar spirit as generalization of a CCG lexicon for CCG-based semantic parser (Kwiatkowski et al., 2011; Wang et al., 2014). Experiments on benchmark data have shown that our model is competitive to previous work and achieves state-of-the-art performance across a few different languages.

Recently the research of semantic parsing in open domain with weakly (or un-) supervised setups, under different settings where the goal was to optimize the performance of certain downstream NLP tasks such as answering questions, has received a significant amount of attention (Poon and Domingos, 2009; Clarke et al., 2010; Berant et al., 2013; Berant and Liang, 2014). One direc-

tion of our future work is to extend the current framework to support the generation of synthetic translation rules from weaker signals (e.g., from question-answer pairs), rather than from aligned parallel data.

We also noticed recent advance in tree-based SMT. Applying such string-to-tree or tree-to-tree translation models (Yamada and Knight, 2001; Shen et al., 2008) to semantic parsing will naturally resolve the inconsistent semantic structure issue, though they require additional information to generate tree labels on the target side. However, due to the constraint that each target phrase needs to map to a syntactic constituent, phrase tables in tree-based translation models usually suffer from the low coverage issue, especially if the training data size is small. Therefore, another direction of our future work is to explore specific problems that will emerge when employing tree-based SMT systems to semantic parsing, and provide solutions to them.

Acknowledgments

References

Jacob Andreas, Andreas Vlachos, and Stephen Clark. 2013. Semantic parsing as machine translation. In *Proceedings of ACL 2013*, pages 47–52.

Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing. In *Proceedings of ACL 2014*, pages 1415–1425.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of EMNLP 2013*, pages 1533–1544.

Peter E. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–313.

David Chiang, Yuval Marton, and Philip Resnik. 2008. Online large-margin training of syntactic and structural translation features. In *Proceedings of EMNLP 2008*, pages 224–233.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

James Clarke, Dan Goldwasser, Ming-Wei Chang, and Dan Roth. 2010. Driving semantic parsing from the worlds response. In *Proceedings of CoNLL 2010*, pages 18–27.

Chris Dyer, Adam Lopez, Juri Ganitkevitch, Jonathan Weese, Ferhan Ture, Phil Blunsom, Hendra Setiawan, Vladimir Eidelman, and Philip Resnik. 2010. cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *Proceedings of ACL 2010 System Demonstrations*, pages 7–12.

Marcello Federico, Nicola Bertoldi, and Mauro Cettolo. 2008. IRSTLM: an open source toolkit for handling large scale language models. In *Proceedings of Interspeech 2008*, pages 1618–1621.

Ruifang Ge and Raymond Mooney. 2005. A statistical semantic parser that integrates syntax and semantics. In *Proceedings of CoNLL 2006*, pages 9–16.

Bevan Jones, Mark Johnson, and Sharon Goldwater. 2012. Semantic parsing with bayesian tree transducers. In *Proceedings of ACL 2012*, pages 488–496.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of ACL 2010 System Demonstrations*, pages 177–180.

Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2010. Inducing probabilistic CCG grammars from logical form with higher-order unification. In *Proceedings of EMNLP 2010*, pages 1223–1233.

Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2011. Lexical generalization in ccg grammar induction for semantic parsing. In *Proceedings of EMNLP 2011*, pages 1512–1523.

Tom Kwiatkowski, Eunsol Choi, Yoav Artzi, and Luke Zettlemoyer. 2013. Scaling semantic parsers with on-the-fly ontology matching. In *Proceedings of EMNLP 2013*, pages 1545–1556.

Junhui Li, Zhaopeng Tu, Guodong Zhou, and Josef van Genabith. 2012. Using syntactic head information in hierarchical phrase-based translation. In *Proceedings of WMT 2012*, pages 232–242.

Percy Liang, Michael I. Jordan, and Dan Klein. 2011. Learning dependency-based compositional semantics. In *Proceedings of ACL 2011*, pages 590–599.

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 EMNLP 2008*, pages 783–792.

Wei Lu. 2014. Semantic parsing with relaxed hybrid trees. In *Proceedings of EMNLP 2014*, pages 1308–1318.

Wei Lu. 2015. Constrained semantic forests for improved discriminative semantic parsing. In *Proceedings of ACL-IJCNLP 2015*, pages 737–742.

Klaus Macherey, Franz Josef Och, and Hermann Ney. 2001. Natural language understanding using statistical machine translation. In *Proceedings of EuroSpeech 2001*, pages 2205–2208.

Rohit J. Kate; Raymond J. Mooney. 2006. Using string-kernels for learning semantic parsers. In *Proceedings of ACL-COLING 2006*, pages 913–920.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Kishore A. Papineni, Salim Roukos, and Todd Ward. 1997. Feature-based language understanding. In *Proceedings of EuroSpeech 1997*, pages 1435–1438.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of ACL 2002*, pages 311–318.

Hoifung Poon and Pedro Domingos. 2009. Unsupervised semantic parsing. In *Proceedings of EMNLP 2009*, pages 1–10.

Libin Shen, Jinxi Xu, and Ralph Weischedel. 2008. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proceedings of ACL 2008*, pages 577–585.

Mark Steedman. 2000. *The syntactic process*. The MIT Press.

Zhaopeng Tu, Yang Liu, Yifan He, Josef van Genabith, Qun Liu, and Shouxun Lin. 2012. Combining multiple alignments to improve machine translation. In *Proceedings of COLING 2012: Posters*, pages 1249–1260.

Adrienne Wang, Tom Kwiatkowski, and Luke Zettlemoyer. 2014. Morpho-syntactic lexical generalization for ccg semantic parsing. In *Proceedings of EMNLP 2014*, pages 1284–1295.

Yuk Wah Wong and Raymond Mooney. 2006. Learning for semantic parsing with statistical machine translation. In *Proceedings of HLT-NAACL 2006*, pages 439–446.

Yuk Wah Wong and Raymond Mooney. 2007. Learning synchronous grammars for semantic parsing with lambda calculus. In *Proceedings of ACL 2007*, pages 960–967.

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proceedings of ACL 2001*, pages 523–530.

Luke S. Zettlemoyer and Michael Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proceedings of UAI 2005*, pages 658–666.

Andreas Zollmann and Ashish Venugopal. 2006. Syntax augmented machine translation via chart parsing. In *Proceedings of WMT 2006*, pages 138–141.