Semantic Parsing of Ambiguous Input using Multi Synchronous Grammars

Philip Arthur, Graham Neubig, Sakriani Sakti, Satoshi Nakamura
{philip.arthur, neubig, ssakti, s-nakamura}.om0@is.naist.jp
Graduage School of Information Science, Nara Institute of Science and Technology

Tomoki Toda
tomoki@icts.nagoya-u.ac.jp
Information Technology Center, Nagoya University

1 Introduction

Semantic parsing (SP) is the problem of parsing a given natural language (NL) sentence into a meaning representation (MR) conducive to further processing by applications. One of the major challenges in SP stems from the fact that NL is rife with ambiguities. Previous works using statistical models along with formalisms such as combinatorial categorial grammars, synchronous context free grammars, and dependency-based compositional semantics have shown notable success in resolving these ambiguities [7, 9, 12, 11, 14].

However, in many cases, the input for NL applications is underspecified and ungrammatical. We illustrate the example of search queries in Table 1. From these queries (Column 1) and their MRs (Column 2), we can see that there are several kinds of ambiguity. For example the parser must make the distinction between Kobe as a city or a basketball player. This problem of word sense ambiguity occurs in standard semantic parsing tasks, but there are also more pernicious problems unique to the more ambiguous input posed by search queries. Focusing on the queries "Kobe hotels" and "Kobe flight" we can see that it is also necessary to estimate the latent relationship between words, such as "location" or "destination." However it should be noted that if we take the keyword query and re-express it as a more explicit paraphrase, we can reduce this ambiguity to the point where there is only one reasonable interpretation. For example, in the second line, if we add the preposition "to" the user is likely asking for flights that arriving in Kobe, and if we add "from" the user is asking for departures.

In this paper, we focus on SP of ambiguous input and propose a new method for dealing with the problem of ambiguity. Specifically we describe a framework where an ambiguous input (Column 1) is simultaneously transformed into both its MR (Column 2) and a more explicit, less ambiguous paraphrase (Column 3). The advantage of this method is that it is then possible to verify that the paraphrase indeed expresses the intended meaning of the underspecified input. Specifically, this verification can be done either manually by the system user or automatically using a probabilistic model trained to judge the naturalness of the paraphrases.

As a concrete approach, we build upon the formalism of synchronous context free grammars (SCFG). Unlike traditional SCFGs, which usually only generate one target string (in semantic parsing, an MR), we introduce a new variety of SCFGs called "tri-synchronous" grammars, which generate multiple strings on the target side. This allows us to not only generate the MR, but also jointly generate the more explicit paraphrase. We then use a language model (LM) over the paraphrases generated by each derivation to help determine which derivations, and consequently which MRs, are more likely.

Evaluation was performed using the Geoquery benchmark of 880 query-logic pairs representing questions about US geography [13]. The baseline SCFG parser achieves reasonable accuracy on regular questions but when the same method is used with underspecified input, the system accuracy decreases significantly. On the other hand, when incorporating the proposed tri-synchronous grammar to generate paraphrases and verify them with an LM, we find that it is possible to recover the loss of accuracy, resulting in a model that is able to parse the ambiguous input with significantly better accuracy. [1]

2 Semantic Parsing using SCFGs

Synchronous context free grammars are a generalization of context-free grammars (CFGs) that generate pairs of related strings instead of single strings. Slightly modifying the notation of Chiang [3], we can formalize SCFG rules as:

$$X \rightarrow \langle \gamma_s, \gamma_t \rangle \quad (1)$$

where X is a non-terminal and γ_s and γ_t are strings of terminals and indexed non-terminals on the source and tar-

[1] This is a shortened version of an article previously appearing in the Transaction of the Association for Computational Linguistics [1]. Tools to replicate our experiments can be found at http://isw3.naist.jp/~philip-a/tacl2015/index.html

Copyright(C) 2016 The Association for Natural Language Processing.
All Rights Reserved.

Search Query	Meaning Representation	Paraphrase
Kobe Hotel	$\lambda x\,(\text{hotel}(x) \wedge \text{in}(x, \text{kobe\_city}))$	Hotel in Kobe city
Kobe Flight	$\lambda x\,(\text{flight}(x) \wedge \text{to}(x, \text{kobe\_city}))$	Flight to Kobe city
Kobe Height	height(kobe_bryant)	Height of Kobe Bryant

Table 1: Example of a search query, MR, and its paraphrase.

get side of the grammar. Each non-terminal on the right side is indexed, with non-terminals with identical indices corresponding to each-other.

In SP with SCFGs, γ_s is used to construct a natural language string $\mathcal{S}$ and γ_t is used to construct the MR $\mathcal{T}$ [11]. Wong and Mooney [12] further extended this formalism to handle λ-SCFGs, which treat γ_s as the natural language query and γ_t as an MR based on λ-calculus. SCFG rules are automatically learned from pairs of sentences with input text and the corresponding MR, where the MR is expressed as a parse tree whose internal nodes are predicates, operators, or quantifiers.

In this paper, we build a SP framework using an approach from Li et al. that uses GHKM algorithm [5] to extract SCFGs [8].

3 Semantic Parsing of Keywords

When users input keyword queries, they will often ignore the grammatical structure and omit function words. We perform experiments on this particular variety of ambiguous input, both to examine the effect that it has on parsing accuracy under the baseline model, and to examine whether this sort of ambiguity can be reduced. In this work, we simulate the keyword query $\mathcal{K}$ by altering the original question $\mathcal{S}$ to make it more closely match the style of keyword queries by stop word deletion, and word order shuffling.

Stop word deletion, as its name implies, simply deletes all stop words from the input sentence. We use a stop word list, making a few subjective changes to make the simulated keyword output more realistic. Word order shuffling reorders the tokens randomly, then we had a human annotator fix the order of the keywords manually. This produced a single keyword query $\mathcal{K}$ for a particular question/MR pair in the Geoquery database, which will be used to train and verify our system. At the end we will have a tri-parallel corpus consisting of 880 pairs of keyword, question, and the meaning representation.

4 Joint Semantic Parsing and Paraphrasing using Tri-Synchronous Grammars

We adopt a generalization of SCFGs, the n-synchronous context free grammar (n-SCFG) [1, 10] to build our joint SP framework. In an n-SCFG, the elementary structures are rewrite rules of $n-1$ target sides:

$$X \rightarrow \langle \gamma_1, \gamma_2, ..., \gamma_n \rangle \tag{2}$$

where X is a non-terminal symbol, γ_1 is the source side string of terminal and non-terminal symbols, and $\gamma_2, ...\gamma_n$ are the target side strings. Therefore, at each derivation step, one non-terminal in γ_1 is chosen and all the corresponding non-terminals with the same index in $\{\gamma_2, ..., \gamma_n\}$ are rewritten using a single rule.

4.1 Tri-SCFGs for Semantic Parsing

In this work, we construct the tri-synchronous grammar by transforming the basic SCFG for semantic parsing into a 3-SCFG. Specifically, we first assume that the source question γ_s and target MR γ_t of the original SCFG become the two outputs γ_2 and γ_3 of the new 3-SCFG grammar. γ_1 is the newly added keyword query input.

During the process of model training, we first extract rules consisting of γ_2 and γ_3, then generate γ_1 from γ_2 by deleting the stop-words then rearranging the order of the words based on word alignments between the keyword query and the original question. This is done by rearranging each word in $\mathcal{K}$ based on their original position in $\mathcal{S}$.[2] Finally, we use the tuple $\langle \gamma_1, \gamma_2, \gamma_3 \rangle$ to form rules in our tri-synchronous grammar.

4.2 Integrating Language Models with Tri-SCFGs

LM plays important role in ensuring fluent output by assigning a probability to the target sentence. In case of n-gram language models, this probability is defined as:

$$p_{LM}(W) = \prod_{i=1}^{l} p(w_i | w_{i-1}, w_{i-2}, ...w_{i-n+1}) \tag{3}$$

where the probability of sentence W of length l is calculated as the product of the probability of its words, depending on the previous $n-1$ words.

We could also consider constructing a probabilistic LM over MR $\mathcal{T}$ for semantic parsing. However, constructing a language model for the MR is less straightforward for several reasons. First, the order of the words of MR in the same rooted logical tree will not make a difference in the final result (e.g. for a commutative operator node). Second, while language models for natural text benefit

[2] In practice, we use a statistical word alignment to determine the original position of $\mathcal{K}$ in $\mathcal{S}$.

Copyright(C) 2016 The Association for Natural Language Processing.
All Rights Reserved.

from the large amounts of text data available on the web, obtaining correct MRs to train a model is less trivial.

On the other hand, in our tri-synchronous grammar framework, in addition to the MR itself, we are generating a paraphrase that nonetheless holds some disambiguating power over the MR. The naturalness of this paraphrase output, like the output of the MT system, can easily be judged by a language model, and might have some correlation with the naturalness of the MR itself. Thus, in this work we add a language model over the paraphrase output as a feature of the scoring model.

5 Experiment and Analysis

5.1 Setup

Data: We use the full Geoquery dataset using the standard 10 folds of 792 and 88 test data [12]. We created keyword queries according to the process described in Section 3. We follow standard procedure of removing punctuation and stemming for all natural language text.

Parsing: Parsing is done by performing decoding of the SCFG-based parsing model to translate the input query into an MR including λ-calculus expressions, then firing the query against the database. Before querying the database, we also apply Wong and Mooney's type-checking rules to ensure that all MRs are logically valid [12].

Language Model: For all 3-SCFG systems we use a 4-gram Kneser-Ney smoothed language model. Standard preprocessing such as lowercasing and tokenization is performed before training the models. We build language models with a corpus of question data [4]. In addition, we also use a 4-gram feed-forward neural network language model (NNLM) feature [2]. For the parsing with NNLM, we recalculate the score of the paraphrases by firstly adding the NNLM score as an additional feature and taking the parse with the best score.

Parameter Optimization: For learning the parameters of the scoring function we use 10-fold cross validation on the training data, using a model trained on 712 examples to parse the remaining 79 for each fold.

Evaluation: Following Wong and Mooney [12], we use question answering precision, recall, and F-measure as our evaluation measure. The query is correct if and only if the SCFGs can generate a syntactically correct parse tree, and retrieve the correct answers from the database.

5.2 Results & Analysis

In this section, we examine the effect of the proposed method on accuracy of parsing ambiguous keyword queries. Specifically, in Table 2, the baseline "Direct" method of training a standard SCFG-based semantic parser, the proposed method without language model verification "Tri-LM," and the proposed method using language model with NNLM reranking "Tri+LM."

Input	Method	P	R	F
Question	Direct	.880	.878	.879
Keyword	Direct	.792	.790	.791
	Tri-LM	.804	.790	.797
	Tri+LM	**.830**	**.820**	**.827**

Table 2: Parsing accuracy, where Keyword Direct is the baseline for semantic parsing on keyword queries, and the Tri with the language model (LM) for verification is our proposed method. Bold indicates a significant gain over both Direct and Tri-LM for keyword input according to bootstrap resampling [6] ($p < 0.05$).

Looking at the baseline accuracy over full questions (first row), the recall is slightly superior to 87.6% of Li et al. [8], showing our baseline is comparable to previous work. When we apply the same method to parse the keyword queries (second row), however, the recall drops almost 9%, showing that the ambiguity included in the keyword query input causes large decreases in accuracy of a semantic parser built according to the baseline method. Then, when adding the language model to the 3-SCFG system (fourth row) we can see a significant of 3-4% gain over the Direct and the Tri-LM systems, demonstrating that the proposed method is indeed able to resolve some of the ambiguity in the keyword queries.

To illustrate how the language model helps, we provide two examples in Table 3. The first example shows that considering the original question when parsing from keywords can help improve alignment with the MR for more plausible results. The second example shows the effect of adding the language model to disambiguate the keyword query. Here there are several interpretations for the keyword-query "largest capital state," which also can mean "state that has the largest capital," or "largest state in the capital." The system without the language model incorrectly chooses the latter interpretation, but the system with the language model correctly disambiguates the sentence as it considers the phrase "state in capital" is unlikely, showing the effectiveness of our method.

5.3 Human Evaluation

We performed an additional evaluation in which human annotators evaluate the paraphrases generated from the systems to validate our hypothesis that human users will be even better at judging whether or not a paraphrase makes sense. First, we took the 1-best parse and 7 random parses from the Tri+LM and Tri-LM systems where both systems produced a non-empty n-best. Then we show both the keyword queries and all the paraphrases to human evaluators to choose which paraphrase matches their interpretation. 3 annotators were asked to annotate 300 keyword queries and their paraphrases.

Table 4 shows the improvement of the system with human help. We take all the answers from the annotators and replaced the answer of the Tri+LM system. Overall,

Copyright(C) 2016 The Association for Natural Language Processing.
All Rights Reserved.

Ex.	LM	Paraphrase/MR	Correct
1	Direct	answer(A,(capital(A),loc(A,B),largest(C,population(B,C))))	no
	Tri-LM Tri+LM	answer(A,largest(B,(capital(A),population(A,B)))) what capital has the largest population	yes
	Original Question: what capital has the largest population		
	Original MR: answer(A,largest(B,(capital(A),population(A,B))))		
	Keyword: largest population capital		
2	Direct(-)	answer(A,largest(A,(capital(A),city(A),loc(A,B),state(B))))	no
	Tri-LM	answer(A,largest(A,(state(A),loc(A,B),capital(B)))) what is the largest state in capital	no
	Tri+LM	answer(A,(state(A),loc(B,A),largest(B,capital(B)))) what state has the largest capital	yes
	Original Question: what state has the largest capital		
	Original MR: answer(A,(state(A),loc(B,A),largest(B,capital(B))))		
	Keyword: largest capital state		

Table 3: Examples of paraphrase outputs produced by the direct keyword-MR system, and the proposed systems without and with a language model.

System	Precision
Tri-LM	.803
Tri+LM	.834
Tri+LM+Human	.846

Table 4: System precision with additional human help.

there were 35 questions that changed between the 1-best and human choices, with 23 improving and 12 degrading accuracy. This experiment suggests that it is possible to show the generated paraphrases to human users to improve the accuracy of the semantic parser.

6 Conclusion and Future Work

We introduced a method for constructing a semantic parser for ambiguous input that paraphrases the ambiguous input into a more explicit form, and verifies the correctness using a language model. An evaluation showed that our method is effective in recovering the 9% loss of system accuracies, providing a 3% improvement. Human evaluation also confirmed that manually evaluating the paraphrases generated by our framework can improve the accuracy of the semantic parser further.

There are a number of future directions for this study. First, we plan to scale the proposed method to open domain semantic parsing of search queries over extensive knowledge bases. In addition, we also plan to incorporate a learning framework that learns from unannotated data.

Acknowledgments

This project is supported by grants from the Ministry of Education, Culture, Sport, Science, and Technology of Japan and from the Microsoft CORE program.

References

[1] Philip Arthur, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. Semantic parsing of ambiguous input through paraphrasing and verification. *TACL*, Vol. 3, pp. 571–584, December 2015.

[2] Yoshua Bengio, Ducharme Réjean, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *JMLR*, Vol. 3, pp. 1137–1155, 2003.

[3] David Chiang. Hierarchical phrase-based translation. *CL*, No. 2, pp. 201–228, 2007.

[4] Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. Paraphrase-driven learning for open question answering. In *ACL*, pp. 1608–1618, 2013.

[5] Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. What's in a translation rule? In *NAACL*, pp. 273–280, 2004.

[6] Philipp Koehn. Statistical significance tests for machine translation evaluation. In *EMNLP*, 2004.

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

[8] Peng Li, Yang Liu, and Maosong Sun. An extended GHKM algorithm for inducing lambda-SCFG. In *AAAI*, pp. 605–611, 2013.

[9] Percy Liang, Michael I. Jordan, and Dan Klein. Learning dependency-based compositional semantics. In *ACL*, pp. 590–599, 2011.

[10] Graham Neubig, Philip Arthur, and Kevin Duh. Multi-target machine translation with multi-synchronous context-free grammars. In *NAACL*, Denver, USA, May 2015.

[11] Yuk Wah Wong and Raymond J Mooney. Learning for semantic parsing with statistical machine translation. In *NAACL*, pp. 439–446, 2006.

[12] Yuk Wah Wong and Raymond J Mooney. Learning synchronous grammars for semantic parsing with lambda calculus. In *ACL*, No. 1, pp. 960–967, 2007.

[13] John M. Zelle and Raymond J. Mooney. Learning to parse database queries using inductive logic programming. In *Proceedings of the 13th AAAI*, pp. 1050–1055, 1996.

[14] Luke S. Zettlemoyer and Michael Collins. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. *Uncertainty in Artificial Intelligence (UAI)*, pp. 658–666, 2005.

Copyright(C) 2016 The Association for Natural Language Processing.
All Rights Reserved.