A Statistical Machine Translation Model with Forest-to-Tree Algorithm for Semantic Parsing

Zhihua Liao
College of Teacher Education
Center for Faculty Development
Hunan Normal University
Changsha, China
cfd@hunnu.edu.cn

Yan Xie
English Department
Foreign Studies College
Hunan Normal University
Changsha, China
xieyanhnnu@163.com

Abstract

In this paper, we propose a novel supervised model for parsing natural language sentences into their formal semantic representations. This model treats sentence-to-λ-logical expression conversion within the framework of the statistical machine translation with forest-to-tree algorithm. To make this work, we transform the λ-logical expression structure into a form suitable for the mechanics of statistical machine translation and useful for modeling. We show that our model is able to yield new state-of-the-art results on both standard datasets with simple features.

1 Introduction

Semantic parsers convert natural language *(NL)* sentences to logical forms *(LFs)* through a meaning representation language *(MRL)*. Recent research has focused on learning such parsers directly from corpora made up of sentences paired with logical meaning representations (Artzi and Zettlemoyer, 2011, 2013; Liao and Zhang, 2013; Liao et al., 2015b,a; Lu et al., 2008; Lu and Ng, 2011; Krishnamurthy, 2016; Kwiatkowski et al., 2010, 2011; Zettlemoyer and Collins, 2005, 2007, 2009). And its goal is to learn a grammar that can map new, unseen sentences onto their corresponding meanings, or logical expressions.

While these algorithms usually work well on specific semantic formalisms, it is not clear how well they could be applied to a different semantic formalism. In this paper, we propose a novel supervised approach to learn semantic parsing task using the framework of the statistical machine translation with forest-to-tree algorithm. This method integrates both lexical acquisition and surface realization in a single framework. Inspired by the probabilistic forest-to-string generation algorithm (Lu and Ng, 2011) and the work of Wong and Mooney (2006; 2007a; 2007b) and Wong (2007) that learn for semantic parsing with statistical machine translation, our semantic parsing framework consists of two main components. Firstly it contains a lexical acquisition component, which is based on phrase alignments between natural language sentences and linearized semantic parses, given by an off-the-shelf phrase alignment model trained on a set of training examples. The extracted transformation rules form a synchronous context free grammar (SCFG), for which a probabilistic model is learned to resolve parse ambiguity. The second component is to estimate the parameters of a probabilistic model. The parametric models are based on maximum-entropy. The probabilistic model is trained on the same set of training examples in an unsupervised manner.

This paper is structured as follows. Section 2 describes how we build the framework of the statistical machine translation with forest-to-tree algorithm to develop a semantic parser, and Section 3 discusses the decoder. Then Section 4 presents our experiments and reports the results. Finally, we make the conclusion in Section 5.

2 The Semantic Parsing Model

Now we present the algorithm for semantic parsing, which translates NL sentences into LFs using a reduction-based λ-SCFG. It is based on an extended version of a reduction-based SCFG (Lu and Ng, 2011). Given a set of training sentences paired with their correct logical forms, the main learning task is to induce a set of reduction-based λ-SCFG rules, which we call a lexicon, a probabilistic model for derivations. A lexicon defines the set of derivations that are possible, so the induction of probabilistic model first requires a lex-

Proceedings of Recent Advances in Natural Language Processing, pages 446–451,
Varna, Bulgaria, Sep 4–6 2017.
https://doi.org/10.26615/978-954-452-049-6_059

icon. Therefore, the learning task can be separated into two sub-tasks:(1) the induction of a lexicon;(2) the induction of a probabilistic model - maximum-entropy model.

2.1 Lexical Acquistion

We introduce the grammar first. Next, we present the generative model for the grammar induction to acquire the grammar rules.

Grammar: We use a weighted λ-SCFG. The grammar is defined as follows: $\tau \rightarrow \langle h_\omega, p_\lambda, \sim \rangle$ where τ is the type associated with the sequence h_ω consisting of natural language words intermixed with types and the λ-production p_λ. The symbol $\sim$ denotes the one-to-one correspondence between nonterminal occurrences in both h_ω and p_λ. Specially, the symbol $\hat{\sim}$ denotes the one-to-one correspondence between terminal occurrence in both $\hat{h}_\omega$ and $\hat{p}_\lambda$, where $\hat{h}_\omega$ is an NL phrase and $\hat{p}_\lambda$ is the LF translation of $\hat{h}_\omega$. Then we allow a maximum of two nonterminal symbols in each synchronous rule (Lu and Ng, 2011). This makes the grammar a binary λ-SCFG.

Grammar Induction: We adopt a generative model for λ-hybrid tree models the mapping from λ-sub-expressions to word sequences with a joint generative process, which Lu and Ng (2011) developed. Figure 1 describes the generative process for a sentence together with its corresponding λ-meaning tree. It results in a λ-hybrid tree[1] (Lu et al., 2008). Figure 2 gives a part of the example λ-hybrid tree. Here, grammar rules are extracted from the λ-hybrid trees. We can use the same grammar for both parsing and generation. Since a SCFG is fully symmetric with respect to both generated strings, the same chart for parsing can be easily adapted for efficient parsing. Now we show how to use the generative model for mapping natural language sentence to λ-expressions. At first, this model finds the Viterbi λ-hybrid trees for all training instances, based on the learned parameters of the generative λ-hybrid tree model. Next, the model extracts grammar rules on the top of these λ-hybrid trees. Specifically, we extract the following tree types of synchronous grammar rules. They are λ-hybrid sequence rules, subtree

[1]The internal nodes of the λ-hybrid tree are called λ-productions, which are building blocks of a λ-forest. Each λ-production in turn has at most two child λ-productions. A λ-production has the form $\tau_a : \pi_a \triangleleft \overline{\tau_b}$, where τ_a is the expected type after type evaluation of the terms to its right, π_a is a λ-expression, and $\overline{\tau_b}$ are types of the child λ-productions.

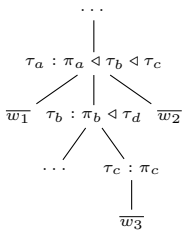


Figure 1: The joint generative process of both λ-meaning tree and its corresponding natural language sentence.

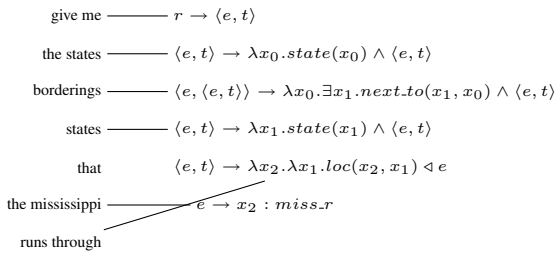


Figure 3: A phrase alignment based on a λ-hybrid tree.

rules and two-level λ-hybrid sequence rules. Here we give an example in Table 1.

1. λ-hybrid sequence rules: these conventional rules are constructed from one λ-production and its corresponding λ-hybrid sequence.

2. Subtree rules: these rules are constructed from a complete substree of the λ-hybrid tree. A mapping between a complete sub-expression and a contiguous sub-sentence can be acquired from each rule.

3. Two-level λ-hybrid sequence rules: these rules are constructed from a tree fragment with one of its grandchild subtrees being abstracted with its type only. These rules are constructed via substitution and reductions. We show how to construct two-level λ-hybrid sequence rules through substitution and reductions. Table 2 gives an example based on a tree fragment of the λ-hybrid tree in Figure 2.

To ground our discussion, we use the phrase alignment in Figure 2 as an example. To represent the logical form in Figure 3, we use its linearized parse — a list of MRL productions that generate the logical form in top-down and left-most order. Since the MRL grammar is unambiguous, every

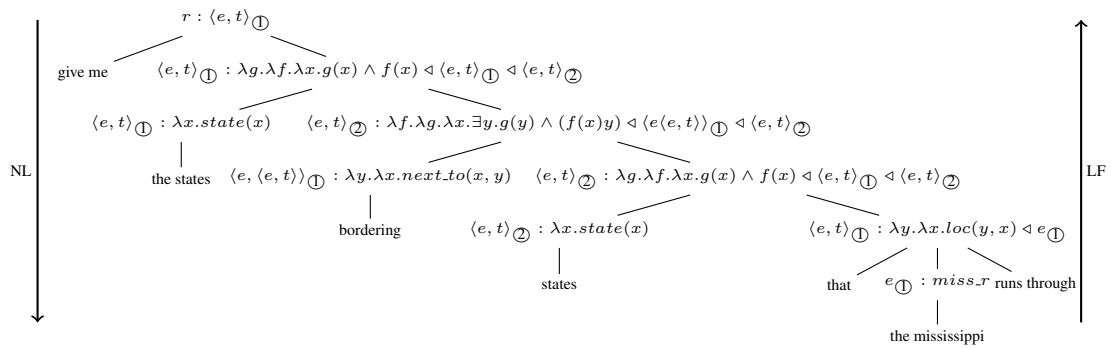


Figure 2: One example $\lambda-$hybrid tree for the sentence "give me the states bordering states that the mississippi runs through" together with its logical form "$\lambda x_0.state(x_0) \wedge \exists x_1.[loc(miss_r, x_1) \wedge state(x_1) \wedge next_to(x_1, x_0)]$".

type 1:	$\langle e,\langle e,t\rangle \rightarrow \langle$ bordering, $\lambda y.\lambda x.next\_to(x,y)\rangle$ $\langle e,t\rangle \rightarrow \langle\langle e,t\rangle_{②} \langle e,t\rangle_{①}$, $\lambda g.\lambda y.\lambda x.g(x) \wedge f(x) \triangleleft \langle e,t\rangle_{①} \triangleleft \langle e,t\rangle_{②}\rangle$
type 2:	$\langle e,t\rangle \rightarrow \langle$ states that the mississippi runs through, $\lambda x.loc(miss\_r, x) \wedge state(x)\rangle$ $\langle e,t\rangle \rightarrow \langle$ that the mississippi runs through, $\lambda x.loc(miss\_r, x)\rangle$
type 3:	$\langle e,t\rangle \rightarrow \langle$ the states bordering $\langle e,t\rangle_{①}$, $\lambda f.\lambda x.state(x) \wedge \exists y.[f(y) \wedge next\_to(y,x)] \triangleleft \langle e,t\rangle_{①}\rangle$ $\langle e,t\rangle \rightarrow \langle$ states that $e_{①}$ runs through, $\lambda y.\lambda x.loc(y,x) \wedge state(x) \triangleleft e_{①}\rangle$

Table 1: Example synchronous rules that can be extracted from the $\lambda-$hybrid tree.

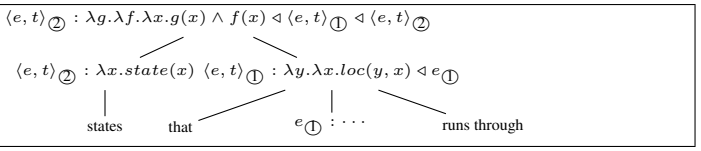
Tree fragment:	
Source: Target:	states that $e_{\square}$ runs through ($\alpha-$conversion)$\Rightarrow$ $\lambda y.\lambda x.loc(y,x) \wedge state(x) \triangleleft e_{①}$ ($\beta-$conversion)$\Rightarrow$ $\lambda y'.\lambda x.loc(y',x) \wedge state(x) \triangleleft e_{①}$ (two $\beta-$conversion)$\Rightarrow$ $\lambda y'.[\lambda f.\lambda x.loc(y',x) \wedge f(x) \triangleleft \lambda x.state(x)] \triangleleft e_{①}$ (substitution)$\lambda y'.[\lambda g.\lambda f.\lambda x.g(x) \wedge f(x) \triangleleft [\lambda y.\lambda x.loc(y,x) \triangleleft y'] \triangleleft \lambda x.state(x)] \triangleleft e_{①}$
Rule:	$\langle e,t\rangle \rightarrow \langle$ states that $e_{①}$ runs through, $\lambda y.\lambda x.loc(y,x) \wedge state(x) \triangleleft e_{①}\rangle$

Table 2: Construction of a two-level $\lambda-$hybrid sequence rule through substitution and reductions from a tree fragment. Note that the subtree rooted by $e_{\square} : miss\_r$ gets "abstracted" by its type e. The auxiliary variable y' of type e is thus introduced to facilitate the construction process.

logical form has a unique linearized parse. We assume the alignment to be n-to-1, where each word is linked to at most one MRL production. Basically, a reduction-based λ-SCFG grammar rule and a phrase alignment (Koehn et al., 2003) can be extracted from an λ-hybrid tree where logical variables are explicitly bound by λ-operators. And these grammar rules are extracted in a bottom-up manner, starting with MRL productions at the leaves of the λ-hybrid tree. Rule extraction continues in this manner until the root of the λ-hybrid tree is reached.

2.2 A Maximum-Entropy Model

Once a lexicon is acquired, the next task is to learn a probabilistic model for the semantic parser. We propose a maximum-entropy model that defines a conditional probability distribution over derivations d given the observed NL string w. Here, the maximum-entropy model is an exponential model:

$$P_\lambda(d|w) = \frac{1}{Z_\lambda(w)} \exp \sum_i \lambda_i f_i(d)$$

where the conditional probability, $P_\lambda(d|w)$, is proportional to the product of weights λ_i assigned to each feature f_i. A feature represents a certain characteristic of a derivation. In this case, the features are the number of times each transformation rule is used in a derivation. The function $Z_\lambda(w)$, called a partition function, is a normalizing factor such that the conditional probabilities sum to *one* over all derivations that yield w. A consequence is that feature weights, λ_i, can be any positive numbers. In a maximum-entropy model, generation of unseen words can be modeled using an extra feature, $f_*(d)$, whose value is the number of all words being skipped. Additional features that correspond to domain-specific word classes can be used for more fine-grained smoothing. The fact that these features may interact with each other is not a concern.

3 Decoding

Decoding of a maximum-entropy model can be done as following:

$$\begin{aligned} m^* &= m(\arg \max_{d \in D(G|w)} P_\lambda(d|w)) \\ &= m(\arg \max_{d \in D(G|w)} \exp \sum_i \lambda_i f_i(d)) \\ &= m(\arg \max_{d \in D(G|w)} \sum_i \lambda_i f_i(d)) \end{aligned}$$

It can be done in the cubic time with respect to sentence length using the *Viterbi* algorithm. An Earley chart is used for keeping track of all derivations that are consistent with the input. The maximum conditional likelihood criterion is used for estimating a maximum-entropy model parameters λ_i. This means that the conditional likelihood of f_i given w is maximized. This criterion is chosen because it is much easier to work with, and it allows for a form of discriminative learning that focuses on separating good parses from bad ones. A Gaussian prior ($\sigma^2 = 1$) is used for regularizing the model. Since the gold-standard derivations are not available in the training data, correct derivations must be treated as hidden variables. Here, to find a set of parameters λ^* that locally maximize the conditional likelihood, we use a version of improved iterative scaling (IIS) coupled with EM which has been used for estimating probabilistic unification-based grammars. Unlike the fully-supervised case, the conditional likelihood is not concave with respect to λ, so the estimation algorithm is sensitive to initial parameters. To assume as little as possible, λ is initialized to zero. The estimation algorithm requires for a statistics that depend on all possible derivations for a sentence or a sentence-MR pair. While it is not feasible to enumerate all derivations, a variant of the Inside-Outside algorithm can be used for efficiently collecting the required statistics. Only rules that are used in the best parses for the training set are retained in the final lexicon, and all other rules are discarded (Wong and Mooney, 2006, 2007b,a; Wong, 2007). This heuristic, commonly known as *Viterbi approximation*, is used to improve accuracy, when we assume that rules used in the best parses are the most accurate.

4 Experimental Setup

This section describes our experimental setup and comparisons of the result. We follow the setup of Zettlemoyer and Collins (2007) and Kwiatkowski et al. (2010; 2011), including datasets, and initialization as well as systems, as reviewed below. Finally, we report the experimental results.

Datasets: We evaluate on two benchmark closed-domain datasets. GeoQuery is made up of natural language queries to a database of geographical information, while ATIS contains natural language queries to a flight booking system (Zettlemoyer and Collins, 2007). The Geo880 dataset has been

(a) The Geo250 test set

system	Rec.	Pre.	F1
λ-WASP	75.6	91.8	82.9
UBL	81.8	83.5	82.6
FUBL	83.7	83.7	83.7
SMTFOREST2STRING	85.0	88.5	86.8

(b) The Geo880 test set

system	Rec.	Pre.	F1
ZC07	86.1	91.6	88.8
UBL	87.9	88.5	88.2
FUBL	88.6	88.6	88.6
SMTFOREST2STRING	89.6	91.8	90.7

Table 3: Performance of Exact Match between the different GeoQuery test sets.

system	Rec.	Pre.	F1
ZC07	74.4	87.3	80.4
UBL	65.6	67.1	66.3
FUBL	81.9	82.1	82.0
SMTFOREST2STRING	84.2	90.3	87.3

Table 4: Performance of Exact Match on the ATIS development set.

(a) Exact Match

system	Rec.	Pre.	F1
ZC07	84.6	85.8	85.2
UBL	71.4	72.1	71.7
FUBL	82.8	82.8	82.8
SMTFOREST2STRING	84.2	88.0	86.1

(b) Partial Match

system	Rec.	Pre.	F1
ZC07	96.7	95.1	95.9
UBL	78.2	98.2	87.1
FUBL	95.2	93.6	94.6
SMTFOREST2STRING	96.0	96.8	96.4

Table 5: Performance of Exact and Partial Matches on the ATIS test set.

split into a training set of 600 pairs and a test set of 280 ones. The Geo250 dataset is a subset of the Geo880, and is used 10-fold cross validation experiments with the same splits of this subset. The ATIS dataset is split into a 5000 example development set and a 450 example test set.

Initialization: For our algorithm learning, we use Och and Ney's (2003; 2004) GIZA++ implementation of IBM Model 5 for training word alignment models. IBM Models 1-4 are used for initializing the model parameters during training.

Systems: We compare this performance to those recently-published and directly-comparable results. For GeoQuery, they include the ZC07 (Zettlemoyer and Collins, 2007), λ-WASP (Wong and Mooney, 2007b; Wong, 2007), UBL (Kwiatkowski et al., 2010) and FUBL (Kwiatkowski et al., 2011). For ATIS, we report results from ZC07, UBL and FUBL.

Results: Tables 3-5 present all the results on the GeoQuery and ATIS domains. In all cases, our system achieves at state-of-the-art recall and precision when compared to directly comparable systems and it significantly outperforms ZC07, λ-WASP, UBL and FUBL. The major advantage of our algorithm over other three systems is that it does not require any prior knowledge of the NL syntax. Hence it is straightforward to apply this algorithm to other NL sentences for which training data is available.

5 Conclusion

This paper presents a novel supervised method for semantic parsing which adopts the framework of the statistical machine translation with forest-to-tree algorithm. The experiments on both benchmark datasets (i.e., GeoQuery and ATIS) show that our method achieves suitable performances.

Acknowledgments

We are grateful to the anonymous reviewers for their valuable feedback on an earlier version of this paper. This research was supported in part by the Foreign Language Teaching Research Project of National Universities (grant no.2015HN0009B) and the Social Science Foundation of Hunan Province for Youth Program (grant no.14YBA260).

References

Yoav Artzi and Luke Zettlemoyer. 2011. Bootstrapping semantic parsers from conversations. In *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Yoav Artzi and Luke Zettlemoyer. 2013. Weakly supervised learning of semantic parsers for mapping instructions to actions. *Transactions of the Association for Computational Linguistics (TACL)* .

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*.

Jayant Krishnamurthy. 2016. Probabilistic models for learning a semantic parser lexicon. In *the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*.

Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2010. Inducing probabilistic ccg grammars from logical form with higher-order unification. In *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Cambridge, MA.

Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2011. Lexical generalization in ccg grammar induction for semantic parsing. In *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Edinburgh, UK.

Zhihua Liao, Qixian Zeng, and Qiyun Wang. 2015a. Semantic parsing via ℓ_0-norm-based alignment. In *Recent Advances in Natural Language Processing(RANLP)*. pages 355–361.

Zhihua Liao, Qixian Zeng, and Qiyun Wang. 2015b. A supervised semantic parsing with lexical extension and syntactic constraint. In *Recent Advances in Natural Language Processing(RANLP)*. pages 362–370.

Zhihua Liao and Zili Zhang. 2013. Learning to map chinese sentences to logical forms. In *the 7th International Conference on Knowledge Science, Engineering and Management (KSEM)*. pages 463–472.

Wei Lu and Hwee Tou Ng. 2011. A probabilistic forest-to-string model for language generation from typed lambda calculus expressions. In *the Conference on Empirical Methods in Natural Language Processing(EMNLP)*.

Wei Lu, Hwee Tou Ng, Wee Sun Lee, and Luke S. Zettlemoyer. 2008. A generative model for parsing natural language to meaning representations. In *the Conference on Empirical Methods in Natural Language Processing(EMNLP)*.

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

Franz Joseph Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics* 30:417–449.

Yuk Wah Wong. 2007. *Learning for Semantic Parsing and Natural Language Generation Using Statistical Machine Translation Techniques*. Ph.D. thesis, Department of Computer Sciences, University of Texas at Austin, Austin, TX.

Yuk Wah Wong and Raymond J. Mooney. 2006. Learning for semantic parsing with statistical machine translation. In *the Human Language Technology Conference of the North American Association for Computational Linguistics (NAACL)*.

Yuk Wah Wong and Raymond J. Mooney. 2007a. Generation by inverting a semantic parser that uses statistical machine translation. In *the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT-07)*.

Yuk Wah Wong and Raymond J. Mooney. 2007b. Learning synchronous grammars for semantic parsing with lambda calculus. In *the Conference of the Association for Computational Linguistics (ACL)*.

Luke S. Zettlemoyer and Michael Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *the 21st Conference on Uncertainty in Artificial Intelligence (UAI)*. pages 658–666.

Luke S. Zettlemoyer and Michael Collins. 2007. Online learning of relaxed ccg grammars for parsing to logical form. In *the Conference on Empirical Methods in Natural Language Processing and the Conference on Computational Natural Language Learning (EMNLP-CoNLL)*. pages 678–687.

Luke S. Zettlemoyer and Michael Collins. 2009. Learning context-dependent mappings from sentences to logical form. In *Joint conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP)*. pages 976–984.