Semantic Parsing as Machine Translation

Jacob Andreas
Computer Laboratory
University of Cambridge
jda33@cam.ac.uk

Andreas Vlachos
Computer Laboratory
University of Cambridge
av308@cam.ac.uk

Stephen Clark
Computer Laboratory
University of Cambridge
sc609@cam.ac.uk

Abstract

Semantic parsing is the problem of deriving a structured meaning representation from a natural language utterance. Here we approach it as a straightforward machine translation task, and demonstrate that standard machine translation components can be adapted into a semantic parser. In experiments on the multilingual GeoQuery corpus we find that our parser is competitive with the state of the art, and in some cases achieves higher accuracy than recently proposed purpose-built systems. These results support the use of machine translation methods as an informative baseline in semantic parsing evaluations, and suggest that research in semantic parsing could benefit from advances in machine translation.

1 Introduction

Semantic parsing (SP) is the problem of transforming a natural language (NL) utterance into a machine-interpretable meaning representation (MR). It is well-studied in NLP, and a wide variety of methods have been proposed to tackle it, e.g. rule-based (Popescu et al., 2003), supervised (Zelle, 1995), unsupervised (Goldwasser et al., 2011), and response-based (Liang et al., 2011).

At least superficially, SP is simply a machine translation (MT) task: we transform an NL utterance in one language into a statement of another (un-natural) meaning representation language (MRL). Indeed, successful semantic parsers often resemble MT systems in several important respects, including the use of word alignment models as a starting point for rule extraction (Wong and Mooney, 2006; Kwiatkowski et al., 2010) and the use of automata such as tree transducers (Jones et al., 2012) to encode the relationship between NL and MRL.

The key difference between the two tasks is that in SP, the target language (the MRL) has very different properties to an NL. In particular, MRs must conform strictly to a particular structure so that they are machine-interpretable. Contrast this with ordinary MT, where varying degrees of wrongness are tolerated by human readers (and evaluation metrics). To avoid producing malformed MRs, almost all of the existing research on SP has focused on developing models with richer structure than those commonly used for MT.

In this work we attempt to determine how accurate a semantic parser we can build by treating SP as a pure MT task, and describe pre- and post-processing steps which allow structure to be preserved in the MT process.

Our contributions are as follows: We develop a semantic parser using off-the-shelf MT components, exploring phrase-based as well as hierarchical models. Experiments with four languages on the popular GeoQuery corpus (Zelle, 1995) show that our parser is competitve with the state-of-the-art, in some cases achieving higher accuracy than recently introduced purpose-built semantic parsers. Our approach also appears to require substantially less time to train than the two best-performing semantic parsers. These results support the use of MT methods as an informative baseline in SP evaluations and show that research in SP could benefit from research advances in MT.

2 MT-based semantic parsing

The input is a corpus of NL utterances paired with MRs. In order to learn a semantic parser using MT we linearize the MRs, learn alignments between the MRL and the NL, extract translation rules, and learn a language model for the MRL. We also specify a decoding procedure that will return structured MRs for an utterance during prediction.

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 47–52,
Sofia, Bulgaria, August 4-9 2013.

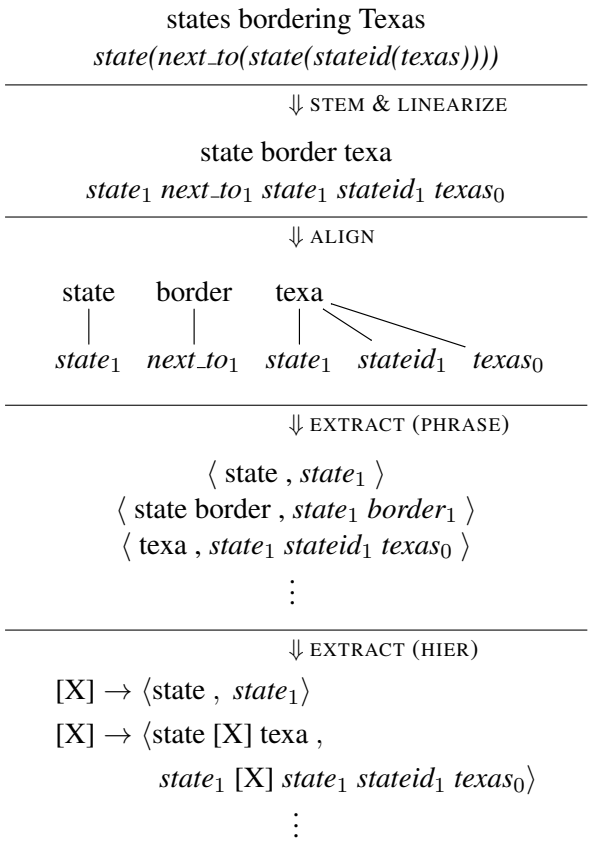


Figure 1: Illustration of preprocessing and rule extraction.

Linearization We assume that the MRL is variable-free (that is, the meaning representation for each utterance is tree-shaped), noting that formalisms with variables, like the λ-calculus, can be mapped onto variable-free logical forms with combinatory logics (Curry et al., 1980).

In order to learn a semantic parser using MT we begin by converting these MRs to a form more 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. After translation, recovery of the function is easy: if the arity of every function in the MRL is known, then every traversal uniquely specifies its corresponding tree. Using an example from GeoQuery, given an input function of the form

answer(population(city(cityid('seattle', 'wa'))))

we produce a "decorated" translation input of the form

$$answer_1\ population_1\ city_1\ cityid_2\ seattle_0\ wa_0$$

where each subscript indicates the symbol's arity (constants, including strings, are treated as zero-argument functions). Explicit argument number labeling serves two functions. Most importantly, it eliminates any possible ambiguity from the tree reconstruction which takes place during decoding: given any sequence of decorated MRL tokens, we can always reconstruct the corresponding tree structure (if one exists). Arity labeling additionally allows functions with variable numbers of arguments (e.g. *cityid*, which in some training examples is unary) to align with different natural language strings depending on context.

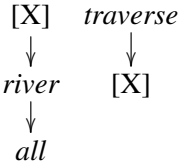
Alignment Following the linearization of the MRs, we find alignments between the MR tokens and the NL tokens using the IBM Model 4 (Brown et al., 1993). Once the alignment algorithm is run in both directions (NL to MRL, MRL to NL), we symmetrize the resulting alignments to obtain a consensus many-to-many alignment (Och and Ney, 2000; Koehn et al., 2005).

Rule extraction From the many-to-many alignment we need to extract a translation rule table, consisting of corresponding phrases in NL and MRL. We consider a phrase-based translation model (Koehn et al., 2003) and a hierarchical translation model (Chiang, 2005). Rules for the phrase-based model consist of pairs of aligned source and target sequences, while hierarchical rules are SCFG productions containing at most two instances of a single nonterminal symbol.

Note that both extraction algorithms can learn rules which a traditional tree-transducer-based approach cannot—for example the right hand side

$$[X]\ river_1\ all_0\ traverse_1\ [X]$$

corresponding to the pair of disconnected tree fragments:



(where each X indicates a gap in the rule).

Language modeling In addition to translation rules learned from a parallel corpus, MT systems also rely on an n-gram language model for the target language, estimated from a (typically larger) monolingual corpus. In the case of SP, such a monolingual corpus is rarely available, and we instead use the MRs available in the training data to learn a language model of the MRL. This information helps guide the decoder towards well-formed

structures; it encodes, for example, the preferences of predicates of the MRL for certain arguments.

Prediction Given a new NL utterance, we need to find the n best translations (i.e. sequences of decorated MRL tokens) that maximize the weighted sum of the translation score (the probabilities of the translations according to the rule translation table) and the language model score, a process usually referred to as decoding. Standard decoding procedures for MT produce an n-best list of all possible translations, but here we need to restrict ourselves to translations corresponding to well-formed MRs. In principle this could be done by re-writing the beam search algorithm used in decoding to immediately discard malformed MRs; for the experiments in this paper we simply filter the regular n-best list until we find a well-formed MR. This filtering can be done with time linear in the length of the example by exploiting the argument label numbers introduced during linearization. Finally, we insert the brackets according to the tree structure specified by the argument number labels.

3 Experimental setup

Dataset We conduct experiments on the GeoQuery data set. The corpus consists of a set of 880 natural-language questions about U.S. geography in four languages (English, German, Greek and Thai), and their representations in a variable-free MRL that can be executed against a Prolog database interface. Initial experimentation was done using 10 fold cross-validation on the 600-sentence development set and the final evaluation on a held-out test set of 280 sentences. All semantic parsers for GeoQuery we compare against also makes use of *NP lists* (Jones et al., 2012), which contain MRs for every noun phrase that appears in the NL utterances of each language. In our experiments, the NP list was included by appending all entries as extra training sentences to the end of the training corpus of each language with 50 times the weight of regular training examples, to ensure that they are learned as translation rules.

Evaluation for each utterance is performed by executing both the predicted and the gold standard MRs against the database and obtaining their respective answers. An MR is correct if it obtains the same answer as the gold standard MR, allowing for a fair comparison between systems using different learning paradigms. Following Jones et al. (2012) we report accuracy, i.e. the percentage of NL questions with correct answers, and F_1, i.e. the harmonic mean of precision (percentage of correct answers obtained).

Implementation In all experiments, we use the IBM Model 4 implementation from the GIZA++ toolkit (Och and Ney, 2000) for alignment, and the phrase-based and hierarchical models implemented in the Moses toolkit (Koehn et al., 2007) for rule extraction. The best symmetrization algorithm, translation and language model weights for each language are selected using cross-validation on the development set. In the case of English and German, we also found that stemming (Bird et al., 2009; Porter, 1980) was hepful in reducing data sparsity.

4 Results

We first compare the results for the two translation rule extraction models, phrase-based and hierarchical ("MT-phrase" and "MT-hier" respectively in Table 1). We find that the hierarchical model performs better in all languages apart from Greek, indicating that the long-range reorderings learned by a hierarchical translation system are useful for this task. These benefits are most pronounced in the case of Thai, likely due to the the language's comparatively different word order.

We also present results for both models without using the NP lists for training in Table 2. As expected, the performances are almost uniformly lower, but the parser still produces correct output for the majority of examples.

As discussed above, one important modification of the MT paradigm which allows us to produce structured output is the addition of structure-checking to the beam search. It is not evident, *a priori*, that this search procedure is guaranteed to find any well-formed outputs in reasonable time; to test the effect of this extra requirement on

	en	de	el	th
MT-phrase	75.3	68.8	70.4	53.0
MT-phrase (-NP)	63.4	65.8	64.0	39.8
MT-hier	80.5	68.9	69.1	70.4
MT-hier (-NP)	62.5	69.9	62.9	62.1

Table 2: GeoQuery accuracies with and without NPs. Rows with (-NP) did not use the NP list.

	English [en]		German [de]		Greek [el]		Thai [th]	
	Acc.	F_1	Acc.	F_1	Acc.	F_1	Acc.	F_1
WASP	71.1	77.7	65.7	74.9	70.7	78.6	71.4	75.0
UBL	82.1	82.1	75.0	75.0	73.6	73.7	66.4	66.4
tsVB	79.3	79.3	74.6	74.6	75.4	75.4	78.2	78.2
hybrid-tree	76.8	81.0	62.1	68.5	69.3	74.6	73.6	76.7
MT-phrase	75.3	75.8	68.8	70.8	70.4	73.0	53.0	54.4
MT-hier	80.5	81.8	68.9	71.8	69.1	72.3	70.4	70.7

Table 1: Accuracy and F_1 scores for the multilingual GeoQuery test set. Results for other systems as reported by Jones et al. (2012).

the speed of SP, we investigate how many MRs the decoder needs to generate before producing one which is well-formed. In practice, increasing search depth in the n-best list from 1 to 50 results in a gain of no more than a percentage point or two, and we conclude that our filtering method is appropriate for the task.

We also compare the MT-based semantic parsers to several recently published ones: WASP (Wong and Mooney, 2006), which like the hierarchical model described here learns a SCFG to translate between NL and MRL; tsVB (Jones et al., 2012), which uses variational Bayesian inference to learn weights for a tree transducer; UBL (Kwiatkowski et al., 2010), which learns a CCG lexicon with semantic annotations; and hybrid-tree (Lu et al., 2008), which learns a synchronous generative model over variable-free MRs and NL strings.

In the results shown in Table 1 we observe that on English GeoQuery data, the hierarchical translation model achieves scores competitive with the state of the art, and in every language one of the MT systems achieves accuracy at least as good as a purpose-built semantic parser.

We conclude with an informal test of training speeds. While differences in implementation and factors like programming language choice make a direct comparison of times necessarily imprecise, we note that the MT system takes less than three minutes to train on the GeoQuery corpus, while the publicly-available implementations of tsVB and UBL require roughly twenty minutes and five hours respectively on a 2.1 GHz CPU. So in addition to competitive performance, the MT-based parser also appears to be considerably more efficient at training time than other parsers in the literature.

5 Related Work

WASP, an early automatically-learned SP system, was strongly influenced by MT techniques. Like the present work, it uses GIZA++ alignments as a starting point for the rule extraction procedure, and algorithms reminiscent of those used in syntactic MT to extract rules.

tsVB also uses a piece of standard MT machinery, specifically tree transducers, which have been profitably employed for syntax-based machine translation (Maletti, 2010). In that work, however, the usual MT parameter-estimation technique of simply counting the number of rule occurrences does not improve scores, and the authors instead resort to a variational inference procedure to acquire rule weights. The present work is also the first we are aware of which uses phrase-based rather than tree-based machine translation techniques to learn a semantic parser. hybrid-tree (Lu et al., 2008) similarly describes a generative model over derivations of MRL trees.

The remaining system discussed in this paper, UBL (Kwiatkowski et al., 2010), leverages the fact that the MRL does not simply encode trees, but rather λ-calculus expressions. It employs resolution procedures specific to the λ-calculus such as splitting and unification in order to generate rule templates. Like other systems described, it uses GIZA alignments for initialization. Other work which generalizes from variable-free meaning representations to λ-calculus expressions includes the natural language generation procedure described by Lu and Ng (2011).

UBL, like an MT system (and unlike most of the other systems discussed in this section), extracts rules at multiple levels of granularity by means of this splitting and unification procedure. hybrid-tree similarly benefits from the introduction of

multi-level rules composed from smaller rules, a process similar to the one used for creating phrase tables in a phrase-based MT system.

6 Discussion

Our results validate the hypothesis that it is possible to adapt an ordinary MT system into a working semantic parser. In spite of the comparative simplicity of the approach, it achieves scores comparable to (and sometimes better than) many state-of-the-art systems. For this reason, we argue for the use of a machine translation baseline as a point of comparison for new methods. The results also demonstrate the usefulness of two techniques which are crucial for successful MT, but which are not widely used in semantic parsing. The first is the incorporation of a language model (or comparable long-distance structure-scoring model) to assign scores to predicted parses independent of the transformation model. The second is the use of large, composed rules (rather than rules which trigger on only one lexical item, or on tree portions of limited depth (Lu et al., 2008)) in order to "memorize" frequently-occurring large-scale structures.

7 Conclusions

We have presented a semantic parser which uses techniques from machine translation to learn mappings from natural language to variable-free meaning representations. The parser performs comparably to several recent purpose-built semantic parsers on the GeoQuery dataset, while training considerably faster than state-of-the-art systems. Our experiments demonstrate the usefulness of several techniques which might be broadly applied to other semantic parsers, and provides an informative basis for future work.

Acknowledgments

Jacob Andreas is supported by a Churchill Scholarship. Andreas Vlachos is funded by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 270019 (SPACEBOOK project `www.spacebook-project.eu`).

References

Steven Bird, Edward Loper, and Edward Klein. 2009. *Natural Language Processing with Python*. O'Reilly Media, Inc.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19(2):263–311.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 263–270, Ann Arbor, Michigan.

H.B. Curry, J.R. Hindley, and J.P. Seldin. 1980. *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus, and Formalism*. Academic Press.

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.

Bevan K. Jones, Mark Johnson, and Sharon Goldwater. 2012. Semantic parsing with bayesian tree transducers. In *Proceedings of the 50th Annual Meeting of the Association of Computational Linguistics*, pages 488–496, Jeju, Korea.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 48–54, Edmonton, Canada.

Philipp Koehn, Amittai Axelrod, Alexandra Birch-Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation. In *Proceedings of the International Workshop on Spoken Language Translation*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic.

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, Massachusetts.

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.

Wei Lu and Hwee Tou Ng. 2011. A probabilistic forest-to-string model for language generation from typed lambda calculus expressions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1611–1622. Association for Computational Linguistics.

Wei Lu, Hwee Tou Ng, Wee Sun Lee, and Luke 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, Edinburgh, UK.

Andreas Maletti. 2010. Survey: Tree transducers in machine translation. In *Proceedings of the 2nd Workshop on Non-Classical Models for Automata and Applications*, Jena, Germany.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 440–447, Hong Kong, China.

Ana-Maria Popescu, Oren Etzioni, and Henry Kautz. 2003. Towards a theory of natural language interfaces to databases. In *Proceedings of the 8th International Conference on Intelligent User Interfaces*, pages 149–157, Santa Monica, CA.

M. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.

Yuk Wah Wong and Raymond Mooney. 2006. Learning for semantic parsing with statistical machine translation. In *Proceedings of the 2006 Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 439–446, New York.

John M. Zelle. 1995. *Using Inductive Logic Programming to Automate the Construction of Natural Language Parsers*. Ph.D. thesis, Department of Computer Sciences, The University of Texas at Austin.