Different Uses of Alignments in Extraction of Synchronous Grammars for Translation from Natural to Formal Languages

Natalia Silveira

1) Introduction

In this paper I explore an approach to the problem of finding a semantic representation in a formal language for an utterance in natural language. This problem can be formulated as a translation problem, but it can also be viewed as a deep parsing task; if the target formal language is suitable for expressing interesting semantic representations in a way that supports inference, the translation of a fragment of natural language into it can be viewed as deep semantic parsing (Mooney 2007).

I implemented an algorithm to perform this translation from natural to a formal language, as proposed by Wong and Mooney (2006). This approach is based on the extraction of a synchronous grammar from an aligned corpus of formal language expressions and their translations into natural language sentences. In our experiments, the synchronous grammar is used to parse unseen natural language sentences and find their translation into the formal language. One of the advantages of this approach is that the translation is guaranteed to be correct, since it is backed by a formal grammar. The method takes as input:

1) a parallel corpus composed of
    a. formal language expressions annotated with their derivation trees according to a context-free grammar, and
    b. the natural language translations of those expressions;
2) an alignment mapping objects from the derivation trees (which can be terminal nodes, any nodes, subtrees) to tokens of the natural language;

and produces as output:

3) a synchronous context-free grammar for a fragment of the two languages.

A synchronous context-free grammar is a grammar of the form GS$<L_1, L_2>$ = (N, Σ, Φ, R, S), where N is a finite set of nonterminal symbols and Σ and Φ are the two finite sets of terminal symbols corresponding to the alphabets of, respectively, languages L_1 and L_2. N and $\Sigma \cup \Phi$ are disjunct. R is a finite set of synchronous rewrite rules, that is, ordered triples of the form $<X, \alpha, \beta>$ (which I write $X \rightarrow <\alpha, \beta>$), where $X \in N$, $\alpha \in (N \cup \Sigma)^*$, and $\beta \in (N \cup \Delta)^*$. $X \rightarrow \alpha$ and $X \rightarrow \beta$ are context-free rules in grammars that generate L_1 and L_2, respectively. The nonterminals in β must be a permutation of the nonterminals in α.

The basic idea is that the alignment between derivation objects and tokens can guide the extension of the rules from the original formal language grammar into a synchronous grammar that also generates a fragment of the natural language. In every derivation tree obtained from the formal language side of the corpus, all the subtrees formed by a nonterminal node and its children can be used to extract a context-free rule of the form $X \rightarrow \alpha$. The alignment between objects in that subtree and natural language

tokens can then be used to create an annotation to the extracted rules that projects a second derivation, generating $X \rightarrow <\alpha, \beta>$. Every natural language token aligned to an object associated with an extracted rule will be taken to be derived by that rule; the ordering of symbols in β is given by the linear order in which the aligned natural language tokens appear in the example strings. This is a generalization of the algorithm proposed by Wong and Mooney (2006), in which the alignments are between natural language tokens and formal language derivation productions (which correspond to subtrees in our formulation).

2) Implementation

I now discuss in some more detail the implementation of this model which were tested. The following core steps, which I immediately discuss in more detail, were performed:

1) the formal language expressions were parsed;
2) filters were applied to words in both languages to allow for certain generalizations to be easily captured by the aligner;
3) an alignment was found, using the EM algorithm, between natural language tokens and formal language derivation objects;
4) for every pair of sentence and tree, a parse tree isomorphic to the formal language parse tree was projected for the natural language sentence, based on the alignments found in (3);
5) flattening operations were applied simultaneously to both trees to ensure that both parses could be produced with context-free rules;
6) the nodes in the natural language tree were permuted to preserve the original linear order;
7) synchronous rules were extracted from the pairs of trees;
8) k-fold evaluation was performed to verify whether or not the rules extracted in training sufficed to find correct translations for test sentences.

The corpus my experiments were performed on was the CLANG corpus, which contains 300 expressions in the CLANG language and manually produced translations into English. For the first step, a simple parser for CLANG was generated with the ANTLR tool, based on the grammar provided in Wong (2007) (with some extensions to account for expressions in the data set that were not described by that grammar). Parse trees for all expressions were obtained with this parser.

As discussed in Silveira (2010), the use of filters for tokens in both the natural and formal languages can be useful in this task, as it allows for certain generalizations to surface that both the aligner and the synchronous grammar can profit from. By filters I mean the omission or substitution by some wildcard symbol of certain select words; for instance, numbers can be treated as *NUMBER*, and punctuation signs in both languages can be omitted in the alignment training. In this implementation I substituted numbers and rule identifiers with wildcards so as to facilitate the treatment of unseen tokens in these categories by the grammar, and omitted punctuation signs in both languages. This omission is of particular interest; in Wong (2007), the author cites the incorrect alignment of natural language tokens to meaningless formal language symbols such as parentheses as a reason to align English words to productions rather than CLANG terminals; however, in my implementation these symbols are not seen by the aligner, avoiding this problem. They are, however, seen by the synchronous rule grammar

extractor, which allows for extraction of rules that correctly derive punctuated expressions of the formal language.

In this paper the main contribution I make in relation to Wong and Mooney's (2006) model is the flexibilization of and experimentation with tree-to-sentence alignments in the parallel corpus. Whereas those authors use token-to-production, here I propose token-to-token and token-to-node alignments (the latter one, with and without parent information) and report results on these alternatives as well as on token-to-production alignments. These alignments were obtained with an implementation of IBM Model 1 (Brown et al., 1993) in which the formal language is treated as the target language, which ensures that no natural language token is aligned to more than one formal language object; this entails that it is only generated in one place in the grammar, which is desirable.

The reason for the choice of Model 1 was that it seems that, when comparing between tokens on the one hand and nodes on the other, the notions of distortion parameters used by the more sophisticated IBM models might not be equally applicable, in which case the comparison might not be fair. For example, it is not clear how the concept of distortion used in IBM Model 2 should be affected by the order of derivation tree nodes when these are linearized to be aligned to natural language tokens. By estimating only a parameter based on co-occurrence in pairs of sentences, we avoid this problem; the alignment model is, however, admittedly impoverished.

From the alignments I then obtained what will here be called "aligned trees"; every token of the natural language string is assumed to have been generated by the nonterminal node to which the alignment associates it. In the case of token-to-token alignments, the natural language token is assumed to be generated by the parent of the formal language token; in the case of token-to-production alignments, by the left-hand side of the production; and in the case of token-to-node alignments, by the node to which it is aligned, or its parent, if that node is a terminal. A tree is then obtained.

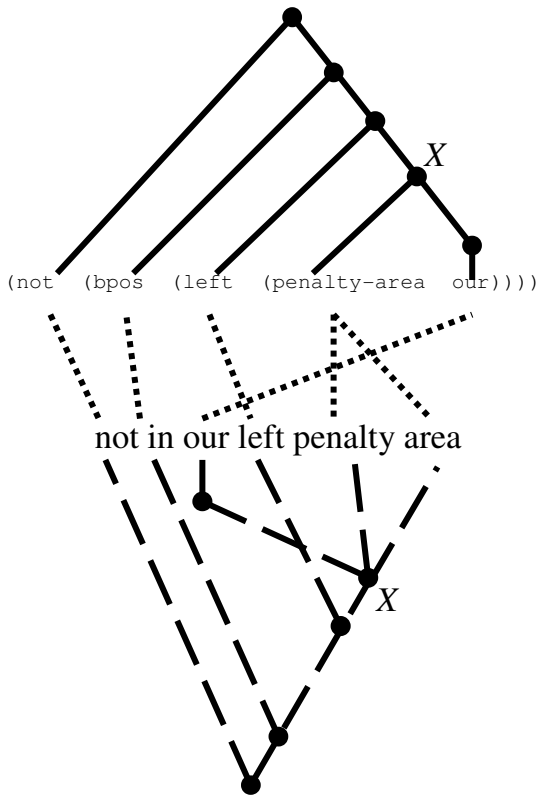

Figure 1. An aligned tree with a problem node, Y.

However, some nodes of the tree thus obtained, which I call problem nodes, may have noncontiguous alignment spans, meaning that their frontier cannot correctly be captured by a context-free rule (Figure 1). In such cases, following Wong and Mooney (2006), I resort to flattening of the tree; both the aligned tree and the tree from which it was generated are modified so that the problem node is destroyed and all its children, terminal or nonterminal, are linked to its parent node. This causes loss of generality because it leads to flatter and less usable rules being extracted, but it permits the extraction of synchronous context-free rules from all the training examples. After all flattening operations have been performed, the nodes of the aligned tree are permuted as necessary to preserve the linear order of the natural language string that they yield.

When a pair of isomorphic and context-free parse trees has been obtained for every sentence pair in the parallel corpus, synchronous rules can then be extracted by means of a simple top-down procedure of traversing each two trees in parallel and creating synchronous rules from the children of each of their nonterminal nodes. Since the nonterminal nodes in the trees are associated by common identifiers, rules of the form $X \rightarrow <\alpha, \beta>$ are easily created: for every X identified across both trees, the list of children in one tree forms α, and the list of children in the other tree forms β.

A synchronous implementation of an Earley parser is then used to find all possible derivations of each natural language string in the test set. A derivation is distinguished not only by its internal structure but also by the translation that it generates.

3) Related Work

As mentioned, the approach discussed in this paper is based on the Wong and Mooney (2006) model. The same authors present a modified model in Wong and Mooney (2007), where the formalism of synchronous grammars is extended to be able to account for languages which include variables. Kate and Mooney (2006) present a similar model in which the association between the natural language tokens and the formal language structures is found with a cascade of SVM classifiers rather than with an aligner. Zettlemoyer and Collins (2005, 2007) use probabilistic categorial grammars with manual seeds to find the translation of natural language sentences into a rich semantic representation in the lambda calculus.

4) Experiments and Discussion

We performed experiments with the three types of alignments proposed: token-to-token, token-to-node and token-to-production. The metric used to evaluate performance was number of test sentences for which a correct derivation was found among all possible ones; we are evaluating rule extraction, and leaving the problem of how to define a probability distribution over the rules aside for a moment. The data set was split for 10-fold evaluation; we used 270 sentences for training and 30 for test. The results reported are the average for ten different splits.

Token-to-token alignment:	**22.3% of correct translations found**
Token-to-node alignment:	**21.7% of correct translations found**
Token-to-production alignment:	**23.7% of correct translations found**

As we can see, the token-to-production alignment resulted in the best performance, but not by a large difference. It is unclear from these results whether there is much to be gained in the choice between token-to-token, token-to-production and token-to-node alignment in filtered inputs; but token-to-production alignments do yield the best results, confirming the findings of Wong and Mooney (2006), who prefer them.

However, the difference is very small, and the defeated alternatives seem worthy of more exploration, as there might be cases in which they are advantageous. The reasons for the similar performance, especially of token-to-token alignments, are probably related to the proposed filters, not adopted in Wong and Mooney (2006), which allow for certain generalizations to surface that would not be otherwise captured in tokens-tokens pairs, but might be more readily evident in tokens-nodes pairs.

We note that performance in these experiments was poor, but the reader should keep in mind that this was a very small data set to train on, and the type of evaluation chosen reflects this directly.

5) Conclusion and Future Work

An interesting next step would be to explore different alignment models; a promising option lies in the work of Yamada and Knight (2001), who propose an alignment model which is constrained by a requirement of isomorphism with a given parse tree. This seems to be suitable for this application, since it would allow us to find the best alignment that does not require any flattening of rules. However, from the theoretical point of view it is unclear whether this approach is correct, since it strengthens the assumption of isomorphism between the syntactic structures of the two languages. This assumption, while motivated and useful in the case of two natural languages, quickly reveals deeply flawed in the case of a natural and a formal language, which might make the adoption of such an alignment model result in a poorer grammar.

6) References

BROWN, P. F.; PIETRA, V. J.; PIETRA, S. A.; et al. (1993) "The mathematics of statistical machine translation: parameter estimation". Computational Linguistics, v. 19, n. 2, pp. 263-311.

KATE, R. J.; MOONEY, R. J. (2006) "Using string-kernels for learning semantic parsers". In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, pp. 913-920, Sydney.

MOONEY, R. J. (2007) "Learning for semantic parsing". Lecture Notes in Computer Science, v. 4394, pp. 311-324.

SILVEIRA, N. G. (2010). Extração de Gramáticas Sincrônicas para Tradução de Linguagens Naturais em Linguagens Formais. M.Sc. dissertation, Universidade Federal do Rio de Janeiro, Rio de Janeiro.

WONG, Y. W. (2007) Learning for Semantic Parsing and Natural Language Generation Using Statistical Machine Translation Techniques. Ph.D. dissertation, University of Texas at Austin, Austin, TX.

WONG, Y. W.; MOONEY, R. J. (2006) "Learning for semantic parsing with statistical machine translation". In: Human Language Technology Conference, pp.439-446, New York, NY.

YAMADA, K.; KNIGHT, K. (2001). "A Syntax-based Statistical Translation Model". In: Proceedings of the 39th Annual Meeting of the Association of Computational Linguistics, pp. 523-530,

ZETTLEMOYER, L. S.; COLLINS, M. (2005) "Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars".

In: Proceedings of 21st Conference on Uncertainty in Artificial Intelligence, pp. 658-666, Edinburgh.

ZETTLEMOYER, L. S.; COLLINS, M. (2007) "Online learning of relaxed CCG grammars for parsing to logical form". In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 678-687, Prague.