Reducing Grounded Learning Tasks to Grammatical Inference

Benjamin Börschinger
Department of Computing
Macquarie University
Sydney, Australia
benjamin.borschinger@mq.edu.au

Bevan K. Jones
School of Informatics
University of Edinburgh
Edinburgh, UK
b.k.jones@sms.ed.ac.uk

Mark Johnson
Department of Computing
Macquarie University
Sydney, Australia
mark.johnson@mq.edu.au

Abstract

It is often assumed that 'grounded' learning tasks are beyond the scope of grammatical inference techniques. In this paper, we show that the grounded task of learning a semantic parser from ambiguous training data as discussed in Kim and Mooney (2010) can be reduced to a Probabilistic Context-Free Grammar learning task in a way that gives state of the art results. We further show that additionally letting our model learn the language's canonical word order improves its performance and leads to the highest semantic parsing f-scores previously reported in the literature.[1]

1 Introduction

One of the most fundamental ideas about language is that we use it to express our thoughts. Learning a natural language, then, amounts to (at least) learning a mapping between the things we utter and the things we think, and can therefore be seen as the task of learning a semantic parser, i.e. something that maps natural language expressions such as sentences into meaning representations such as logical forms. Obviously, this learning can neither take place in a fully supervised nor in a fully unsupervised fashion: the learner does not 'hear' the meanings of the sentences she observes, but she is also not treating them as merely meaningless strings. Rather, it seems plausible to assume that she uses extra-linguistic context to assign certain meanings to the linguistic input she is confronted with.

In this sense, learning a semantic parser seems to go beyond the well-studied task of unsupervised grammar induction. It involves not only learning a grammar for the form-side of language, i.e. language expressions such as sentences, but also the 'grounding' of this structure in meaning representations. It requires going beyond the mere linguistic input to incorporate, for example, perceptual information that provides a clue to the meaning of the observed forms. Essentially, it seems as if 'grounded' learning tasks like this require dealing with two different kinds of information, the purely formal (phonemic) and meaningful (semantic) aspects of language. Grammatical inference seems to be limited to dealing with one level of formal information (Chang and Maia, 2001). For this reason, probably, approaches to the task of learning a semantic parser employ a variety of sophisticated and task-specific techniques that go beyond (but often elaborate on) the techniques used for grammatical inference (Lu et al., 2008; Chen and Mooney, 2008; Liang et al., 2009; Kim and Mooney, 2010; Chen et al., 2010).

In this paper, we show that one can reduce the task of learning a semantic parser to a Probabilistic Context Free Grammar (PCFG) learning task, and more generally, that grounded learning tasks are not in principle beyond the scope of grammatical inference techniques. In particular, we show how to formulate the task of learning a semantic parser as discussed by Chen, Kim and Mooney (2008, 2010) as the task of learning a PCFG from strings. Our model does not only constitute a proof of concept that this

[1] The source code used for our experiments and the evaluation is available as supplementary material for this article.

Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 1416–1425,
Edinburgh, Scotland, UK, July 27–31, 2011. ©2011 Association for Computational Linguistics

reduction is possible for certain cases, it also yields highly competitive results.[2]

By reducing the problem to the well understood PCFG formalism, it also becomes easy to consider extensions, leading to our second contribution. We demonstrate that a slight modification to our model so that it also learns the language's canonical word order improves its performance even beyond the best results previously reported in the literature. This language-independent and linguistically well motivated elaboration allows the model to learn a global fact about the language's syntax, its canonical word order.

Our contribution is two-fold. We provide an illustration of how to reduce grounded learning tasks to grammatical inference. Secondly, we show that extending the model so that it can learn linguistically well motivated generalizations such as the canonical word order can lead to better results.

The structure of the paper is as follows. First we give a short overview of the previous work by Chen, Kim and Mooney and describe their dataset. Then, we show how to reduce the parsing task addressed by them to a PCFG-learning task. Finally, we explain how to let our model additionally learn the language's canonical word order.

2 Previous Work by Chen, Kim and Mooney

In a series of recent papers, Chen, Kim and Mooney approach the task of learning a semantic parser from *ambiguous* training data (Chen and Mooney, 2008; Kim and Mooney, 2010; Chen et al., 2010). This goes beyond previous work on semantic parsing such as Lu et al. (2008) or Zettlemoyer and Collins (2005) which rely on *unambiguous* training data where every sentence is paired only with its meaning. In contrast, Chen, Kim and Mooney allow their training examples to exhibit the kind of uncertainty about sentence meanings human learners are likely to have to deal with by allowing for sentences to be associated with a *set* of candidate-meanings, and the correct meaning might not even be in this set. They create the training data by first collecting humanly generated written language comments on four different RoboCup games. The comments are recorded with a time-stamp and then associated with all game events automatically extracted from the games which occured up to five seconds before the comment was made. This leads to an ambiguous pairing of comments with candidate meanings that can be considered similar to the "linguistic input in the context of a rich, relevant, perceptual environment" to which real language learners probably have access (Chen and Mooney, 2008). For evaluation purposes, they manually create a gold-standard which contains unambiguous natural language comment / event pairs. Due to the fact that some comments refer to events not detected by their extraction-algorithm, not every natural language sentence has a gold matching meaning representation. In addition to the inherent ambiguity of the training examples, the learner therefore has to somehow deal with those examples which only have 'wrong' meanings associated with them.

Datasets exist for both Korean and English, each comprising training and gold data for four games.[3] Some details about this data are given in Table 1, such as the number of examples, their average ambiguity and the number of misleading examples.

For the following short discussion of previous approaches, we mainly focus on Kim and Mooney (2010). This is the most recent publication and reports the highest scores.

2.1 The parsing task

Learning a semantic parser from the ambiguous data is, in fact, just one of three tasks discussed by Kim and Mooney (2010), henceforth KM. In addition to parsing, they discuss matching and natural language generation. We are ignoring the generation task as we are currently only interested in the parsing problem, and we treat the matching task, picking the correct meaning from the set of candidates, merely as a byproduct of parsing, rather than as a completely separate task: parsing implicitly requires the model to disambiguate the data it is learning from.

[2] It has been pointed out to us by one reviewer that the task we address falls short of what is often called 'grounded learning'. We acknowledge that semantic parsing constitutes a very limited kind of grounded learning but want to point out that the task has been introduced as an instance of grounded learning in the previous literature such as Chen and Mooney (2008).

[3] The datasets are freely available at `http://www.cs.utexas.edu/~ml/clamp/sportscasting/`. We retrieved the data used here on March 29th, 2011.

	Number of comments				Ambiguity	
	# Training	# Training with Gold Match	# Training with correct MR	# Gold	Noise	Avg. # of MRs
English dataset						
total	1872	1492	1360	1539	0.2735	2.20
Korean dataset						
total	1914	1763	1733	1763	0.0946	2.39

Table 1: Statistics for the Korean and the English datasets. The numbers are basically identical to those reported in Chen et al. (2010) except for minimal differences in the number of training examples (we give one more for every English training set, and one more for the 2004 Korean training set). In addition, our calculation of the average sentential ambiguity (Avg. # of MRs) differs because we assume that mutiple occurences of the same event in a context do not add to the overall ambiguity, and our calculation of the noise (fraction of training examples without the correct meaning in their context) takes into account that there are training examples which do not have their gold meaning associated with them in the training data and is therefore slightly higher than the one reported in Chen et al. (2010).

KM's model builds on previous work by Lu et al. (2008) and is a generative model which defines a joint probability distribution over natural language sentences (NLs), meaning representations (MRs) and hybrid trees. The NLs are the natural language comments to the games, the MRs are simple logical formulae describing game events and playing the role of sentence meanings, and a hybrid tree is a tree structure that represents the correspondence between a sentence and its meaning. More specifically, if some NL W has as its meaning an MR m, and m has been generated by a meaning grammar (MG) G, the hybrid tree corresponding to the pair ⟨W,m⟩ has as its internal nodes those rules of G used in the derivation of m, and as its leaves the words making up W.[4] An example hybrid tree for the pair ⟨THE PINK GOALIE PASSES THE BALL TO PINK11,pass(pink1,pink11)⟩ is given in Figure 1. Their model is trained by a variant of the Inside-Outside algorithm which deals with the hybrid tree structure and takes into account the ambiguity of the training examples.

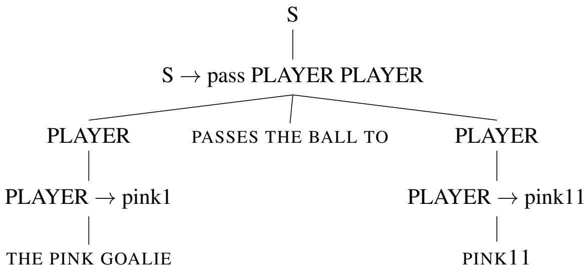

Figure 1: A hybrid tree for the sentence-meaning pair ⟨THE PINK GOALIE PASSES THE BALL TO PINK11,pass(pink1,pink11)⟩ . The internal nodes correspond to the rules used to derive pass(pink1,pink11) from a given Meaning Grammar, and the leaves correspond to the words or substrings that make up the sentence.

In addition to learning directly from the ambiguous training data, they also train a semantic parser in a supervised fashion on data that has been previously disambiguated by their matching model. This slightly improves their system's performance. Consequently, there are two scores for each of the two languages (English and Korean) with which we compare our own model: those of the parsers trained directly from the ambiguous data, and those of the 'supervised' parsers which constitute the current state of the art. The details of their evaluation method are disccused in Section 3.3, and their scores are given in Table 2, together with our own scores.

3 Learning a Semantic Parser as a PCFG-learning problem

Given that one can effectively represent both a sentence's form and its meaning in a hybrid tree, it is interesting to ask whether one can do with a structure that can be learned by grammatical inference tech-

[4] We use SMALL CAPS for words, sans serif for MRs and MR constituents (concepts), and *italics* for non-terminals and Grammars.

niques from strings which incorporate the contextual information. In this section, we show how to reduce hybrid trees to such 'standard' trees. In effect, we show via construction that 'grounded' learning tasks such as learning a semantic parser from semantically enriched and ambiguous data can be reduced to 'ungrounded' tasks such as grammatical inference.

Instead of taking the internal nodes of the trees generated by our model as corresponding to MG *production rules*, we take them to correspond to MR *constituents*. The MR pass(pink1,pink11), for example, has 4 constituents: the whole MR, the predicate pass, and the two arguments pink1 and pink11. Figure 2 gives the tree we assume instead of Figure 1 for the sentence-meaning pair ⟨THE PINK GOALIE PASSES THE BALL TO PINK11,pass(pink1,pink11)⟩. Its root is assumed to correspond to the whole MR and is labeled $S_{pass(pink1,pink11)}$. The remaining three MR constituents correspond to the root's daughters which we label $Phrase_{pink1}$, $Phrase_{pass}$ and $Phrase_{pink11}$. Generally speaking, we assume a special non-terminal S_m for every MR m generated by the MG, and a special non-terminal $Phrase_{con}$ for each of the terminals of the MG (which loosely correspond to concepts). This is only possible for MGs which create a finite set of MRs, but the MG used by Kim and Mooney (2010) obeys this restriction.[5]

The tree's terminals are the words that make up the sentence, and we assume them to be dominated by concept-specific pre-terminals $Word_{con}$ which correspond to concept-specific probability distributions over the language's vocabulary. Since each $Phrase_{con}$ may span multiple words, we give trees rooted in $Phrase_{con}$ a left-recursive structure that corresponds to a unigram Markov-process. This process generates an arbitrary sequence of words semantically related to con, dominated by the corresponding pre-terminal $Word_{con}$ in our model, and words not directly semantically related to con, dominated by a special word pre-terminal $Word_{\emptyset}$. The sole further restriction is that every $Phrase_{con}$ must contain at least one $Word_{con}$.

Trees like the one in Figure 2 can be generated by a Context-Free Grammar (CFG) which, in turn, can be trained on strings to yield a PCFG which embodies a semantic parser as will be discussed in Section 3.3. We now describe how to set up such a CFG in a systematic way and how to train it on the data used by KM.

3.1 Setting up the PCFG

The training data expresses information of two different kinds – form and meaning. Every training example consists of a natural language string (the formal information) and a set of candidate meanings for the string (the semantic information, its context), allowing for the possibility that none of the meanings in the context is the correct one. In order to learn from data like this within a grammatical inference framework, we have to encode the semantic information as part of the string. Assigning a specific MR m to a string corresponds, in our framework, to analyzing it as a tree with S_m as its root. A sentence's context constrains which of the many possible meanings might be expressed by the string. Thus the role played by the context is adequately modelled if we ensure that if a string w is associated with a context $\{m_1,...,m_n\}$, the model only considers the possibilities that this string might be analyzed as $S_{m_1},...,S_{m_n}$.

There are 959 different contexts, i.e. 959 different sets of MRs, in the English data set (984 for the Korean data), and we therefore introduce 959 new terminal symbols which play the role of context-identifiers, for example c_1 to c_{959}.[6] Formally speaking, a context-identifier is a terminal like any other word of the language and we can therefore prefix every comment in the training data with the context-identifier standing for the set of MRs associated with this comment, an idea taken from previous work such as Johnson et al. (2010). Thus having incorporated the contextual information into the string, we go on to show how our model makes use of this information, considering the MR pass(pink1,pink11) as an example. A formal description of the model is given Figure 3.

Assume that pass(pink1,pink11) is associated with only one training example and therefore occurs only in one specific context. If the context-identifier introduced for this context is c_1, we require the

[5] This grammar is given in the Appendix to Chen et al. (2010) and generates a total of 2048 MRs.

[6] If we were to consider every possible context, we would have to consider 2^{2048} contexts because the MG generates 2048 MRs.

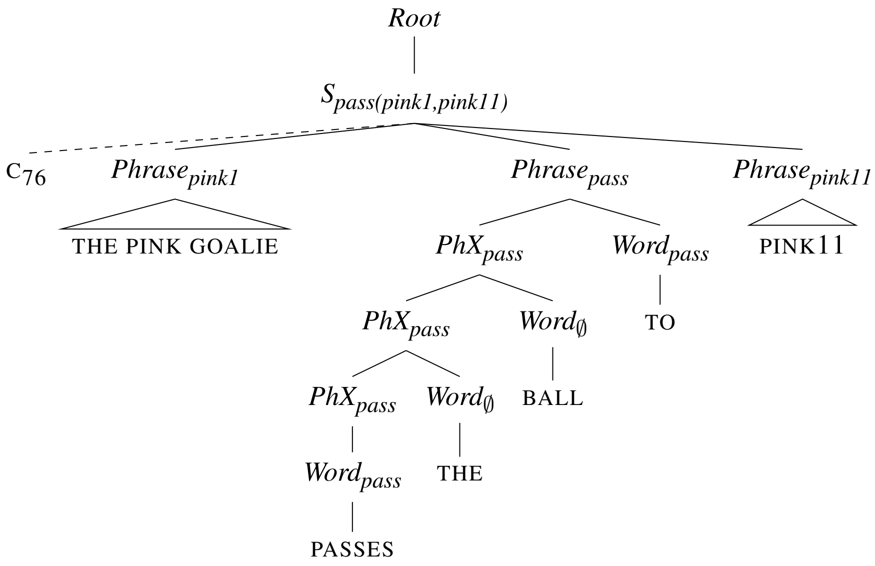

Figure 2: The tree-structure we propose instead of the Hybrid Tree structure used by (Kim and Mooney, 2010). The non-terminal nodes do not correspond to MG productions, but to MR constituents. The internal structure of the *Phrase*$_{con}$ constituents, shown in full detail for *Phrase*$_{pass}$, corresponds to a Markov process that generates the words that make up the sentence. The terminal C_{76} is a context-identifier that restricts the range of S_m non-terminals that might dominate the sentence and is only used during training, as described in Section 3.1. The grammar that generates this trees is described in Figure 3.

right-hand side of all rules with $S_{pass(pink1,pink11)}$ on their left-hand side to begin with C_1. More generally, if an MR m occurs in the contexts associated with the context-identifiers C_K,...,C_L, we require the right-hand side of all rules with S_m on their left-hand side to begin with exactly one of C_K,...,C_L.

In this sense, the *context-identifiers* can be seen as providing the model with a *top-down constraint* – if it encounters a context-identifier, it can only try analyses leading to MRs which are licensed by this context-identifier. On the other hand, the words have to be generated by concept-specific word-distributions, and the concepts that are present restrict the range of possible S_m non-terminals which might dominate the whole string. In this sense, the *words* the model observes provide it with a *bottom-up constraint* – if it sees words which are semantically related to certain concepts con_1,...,con_n, it has to arrive at an MR which licenses the presence of the corresponding *Phrase*$_{con_x}$ non-terminals. Of course, the model has to also learn which words are semantically related to which concepts. To enable it to do this, our grammar allows every *Word*$_x$ non-terminal to be rewritten as every word of the language.

Since there are sentences in the training data without the correct meaning in their context, we want to give our model the possibility of not assigning to a sentence any of the MRs licensed by its context-identifier. To do this, we employ another trick of previous work by Johnson et. al and assume a special null meaning $\emptyset$ to be present in every context. $S_\emptyset$ may only span words generated by *Word*$_\emptyset$, the language-specific distribution for words not directly related to any concept; this also has to be learned by the model.

As a last complication, we deal with the fact that syntactic constituents are linearized with respect to each other. For example, if an MR has 3 proper constituents (i.e. excluding the MR itself), our grammar allows the corresponding 3 syntactic constituents – which we might label *Phrase*$_{predicate}$, *Phrase*$_{arg1}$ and *Phrase*$_{arg2}$ – to occur in any of the 6 possible orders. Therefore, we have an S_m rule for every context in which m occurs and for every possible order of the proper constituents of m.

A formally explicit description of the rule schemata used to generate the CFG is given in Figure 3.[7] Instantiating all those schemata leads to a grammar with 33,101 rules for the English data and 30,731 rules for the Korean data. The difference in size is due to differences in the size of the vocabulary and the different number of contexts in the data sets.

These CFGs can now be trained on the training data using the Inside-Outside algorithm (Lari and Young, 1990). After training, the resulting PCFG embodies a semantic parser in the sense that, with a slight modification we describe in section 3.3, it can be used to parse a string into its meaning representation by determining the most likely syntactic analysis and reading off the meaning assigned by our model at the S_m-node.

3.2 Possible objections to our reduction

Before we go on to discuss the details of training and evaluation of our model, we want to address an objection that might seem tempting. Isn't our reduction impractical and unrealistic as even a highly abstract model of language learning – after all, setting up the huge CFG requires knowledge about the vocabulary, the MG and all the complicated rules discussed which, presumably, is more knowledge than we want to provide a language learner with, lest we trivialize the task. To this we reply firstly, that it is true that our reduction only works for *offline* or *batch grounded learning tasks* where all the data is available to the model before the actual learning begins so that it 'knows' the words, the meanings and the contexts present in the data. This offline constraint is, however, true of all models which are trained by iterating multiple times over training data such as KM's model. Secondly, the intimidating CFG can in principle be reduced to a hand-full of intuitive principles and is easy to generate automatically.

First of all, the many specific S_m-rewrite rules reduce to the heuristic that every semantic constituent should correspond to a syntactic constituent, and the fact that natural language expressions are linearly ordered. Note that our model does not contain knowledge about the specific word order of the language. It simply allows for the constituents of an MR to occur in every possible order which is a very unbiased and empiricist assumption. Of course, this leads to some limited kind of 'implicit learning' of word order in the sense that for every meaning and for every context, our model might (and in most cases will) assign different probabilities to the different rules for every word order; so it can learn that certain specific MRs such as pass(pink1,pink11) are more often linearized in one way than in any other. It cannot, however, generalize this to other (or even unseen) MRs, i.e. it does not learn a global fact about the language. In a way, it lacks the knowledge that there is such a thing as word order, a point which we will elaborate on in Section 4.

The many re-write rules for the pre-terminal $Word_x$s are nothing but an explicit version of the assumption that every word the model encounters might, in principle, be semantically related to every concept it knows. Again, this seems to us to be a reasonable assumption.

Finally, the complicated looking set of rules for the internal structure of $Phrase_x$s corresponds to a simple unigram Markov-process for generating strings. All in all, we do not see that we make any more assumptions than other approaches; our formulation may make explicit how rich those assumptions are but we have not qualitatively changed them.

3.3 Training and Evaluation

The CFG described in the previous section is trained on the same training data used by KM, except that we reduce it to strings (without changing the information present in the original data) by prefixing every sentence with a context-identifier. For training we run the Inside-Outside algorithm[8] with uniform initialization weights until convergence. For English, this results in an average number of 76 iterations for each fold, for Korean the average number of iterations is 50. To deal with the fact that the model might not observe certain meanings during training, we apply a simple smoothing technique by using a Dirichlet prior of α=0.1 on the rule probabilities. In effect, this provides our system with a small number of pseudo-observations for each rule which prevents

[7] In our description, we use context-identifiers such as c_1 with a systematic ambiguity, letting them stand for the terminal symbol representing a context and, in contexts such as $m \in c_1$, for the represented context itself.

[8] We use Mark Johnson's freely available implementation, available at `http://web.science.mq.edu.au/~mjohnson/Software.htm`.

$\mathrm{Root} \rightarrow S_m$	$m \in M \cup \{\emptyset\}$
$S_m \rightarrow c\, Phrase_{p(m)}$	$c \in C, m \in c, m \in Pred0(M)$
$S_m \rightarrow c\, \{Phrase_{p(m)}, Phrase_{a1(m)}\}$	$c \in C, m \in c, m \in Pred1(M)$
$S_m \rightarrow c\, \{Phrase_{p(m)}, Phrase_{a1(m)}, Phrase_{a2(m)}\}$	$c \in C, m \in c, m \in Pred2(M)$
$S_\emptyset \rightarrow c\, Phrase_\emptyset$	$c \in C$
$Phrase_\emptyset \rightarrow Word_\emptyset$	
$Phrase_\emptyset \rightarrow Phrase_\emptyset\, Word_\emptyset$	
$Phrase_x \rightarrow Word_x$	$x \in T$
$Phrase_x \rightarrow PhX_x\, Word_x$	$x \in T$
$Phrase_x \rightarrow Ph_x\, Word_\emptyset$	$x \in T$
$PhX_x \rightarrow Word_r$	$x \in T, r \in \{x, \emptyset\}$
$PhX_x \rightarrow PhX_x\, Word_r$	$x \in T, r \in \{x, \emptyset\}$
$Ph_x \rightarrow PhX_x\, Word_x$	$x \in T$
$Ph_x \rightarrow Ph_x\, Word_\emptyset$	$x \in T$
$Ph_x \rightarrow Word_x$	$x \in T$
$Word_x \rightarrow v$	$x \in T \cup \{\emptyset\}, v \in V$

Figure 3: The rule-schemata used to generate the NoWo-PCFG. Root is the unique start-symbol, M is the set of all MRs present in the corpus, C is set the of all context-identifiers present in the corpus, T is the set of terminals of the MG, V is the vocabulary of the corpus. $Pred0(M)$ is the subset of all MRs in M of the form predicate, $Pred1(M)$ is the subset of all MRs in M of the form predicate(arg1) and $Pred2(M)$ is the subset of all MRs in M of the form predicate(arg1,arg2). $p(m)$ is the predicate of the MR m, $a1(m)$ is the first argument of the MR m, $a2(m)$ is the second argument of the MR m. The rules expanding $Phrase_x$ ensure that it contains at least one $Word_x$. A set on the right-hand side of a rule is shorthand for all possible orderings of the elements of the set.

the automatic assignment of zero probability to rules not used during training.[9]

For parsing, the resulting PCFG is slightly modified by removing the context-identifiers. This is done because the task of a semantic parser is to establish a mapping between NLs and MRs, irrespective of contexts which were only used for learning the parser and should not play a role in its final performance. To do this, we add up the probability of all rules which differ only in the context-identifier which can be thought of as marginalizing out the different contexts, giving our first model which we call NoWo-PCFG.[10]

Note that the context-deletion (and the simple smoothing) enables NoWo-PCFG to parse sentences into meanings not present in the data it was trained on which, in fact, happens. For example, there are 81 meanings in the training data for the first English match that are not present in any of the other games' training data. The PCFG trained on games 2, 3 and 4 is still able to correctly assign 12 of those 81 meanings which it has not seen during the training phase which shows the effectiveness of the bottom-up constraint.

For evaluation, we employ 4-fold cross validation as described in detail in Chen and Mooney (2008) and used by KM: the model is trained on all possible combinations of 3 of the 4 games and is then used to produce an MR for all sentences of the held-out game *for which there is a matching gold-standard meaning*. For an NL w, our model produces an MR m by finding the *most probable* parse of w with the CKY algorithm and reading off m at the S_m-node.[11] An MR is considered correct if and only if it matches the gold-standard MR *exactly*; the final evaluation result is averaged over all 4 folds. Our evaluation results for NoWo-PCFG are given in Table 2. All scores are reported in F-measure which is the harmonic mean of Precision and Recall. In this specific case, precision is the fraction of correct parses out

[9] We experimented with α=0.1, α=0.5 and α=1.0 and found that overall, 0.1 yields the best results. We also tried jittering the initial rule weights during training and found that our results are very robust and seem to be independent of a specific initialization.

[10] NoWo because this model, unlike the one described in Section 4, does **no**t make explicit use of **w**ord **o**rder generalisations.

[11] For parsing, we use Mark Johnson's freely available CKY implementation which can be downloaded at `http://web.science.mq.edu.au/~mjohnson/Software.htm`.

	English	Korean
KM	0.742	0.764
KM ‘supervised’	0.810	0.808
Chen et al. (2010)	0.801	0.812
NoWo-PCFG	0.742	0.718
WO-PCFG	**0.860**	**0.829**

Table 2: A summary of results for the parsing task, in F-measure. We also show the results of Chen et al. (2010), as given in Kim and Mooney (2010), which to our knowledge are the highest previously reported scores for Korean. WO-PCFG, described in Section 4 performs better than all previously reported models, but only slightly so for Korean.

of the total number of parses the model returns. Recall is the fraction of correct parses out of the total number of test sentences.[12]

NoWo-PCFG performs a little worse than KM’s model. Its scores are virtually identical for English (0.742) and worse for Korean (0.718 vs 0.764). We are not sure as to why our model performs worse on the Korean data, but it might have to do with the fact that the Korean average ambiguity is higher than for the English data.

This shows that it is not only possible to reduce the task of learning a semantic parser to standard grammatical inference, but that this way of approaching the problem yields comparable results.

The remainder of the paper focuses on our second main point: that letting the model learn additional kinds of information, such as the language’s canonical word order, can further improve its performance. In order to do this we propose a model that learns the word order as well as the mapping from NLs to MRs, and compare its performance to that of the other models.

4 Extending NoWo-PCFG to WO-PCFG

We already pointed out that our model considers every possible linear order of syntactic constituents. Our NoWo-PCFG model considers each of the possible word orders for every meaning and context in isolation: it is unable to infer from the fact that most meanings it has observed are most likely to be expressed with a certain word order that new meanings it will encounter are also more likely to be expressed with this word order. It seems, however, to be at least a soft fact about languages that they *do* have a canonical word order that is more likely to be realized in its sentences than any other possible word order. In order to test whether trying to learn this order helps our model, we modify the CFG used for NoWo-PCFG so it can learn word order generalizations, and train it in the same way to yield another semantic parser, WO-PCFG.

4.1 Setting up WO-PCFG

For every possible ordering of the constituents corresponding to an MR, our grammar contains a rule. In NoWo-PCFG, these different rules all share the same parent which prevents the model from learning the probability of the different word orders corresponding to the many rules. A straight-forward way to overcome this is to annotate every S_m node with the word order of its daughter. We split every S_m non-terminal in multiple S_{wo_m} non-terminals, where $wo \in \{$v,sv,vs,svo,sov,osv,ovs,vso,vos$\}$ indicates the linear order of the constituents the non-terminal rewrites as.[13]

This in itself does not yet allow our model to *use* word order as a means of generalization. To model that whenever it *encounters a specific example* that is indicative of a certain word order, this word order becomes slightly more probable *for every other example as well*, we have to make a further slight change to the CFG which we now describe. A formally explicit description of the necessary changes which we go on to describe is given in Figure 4.

We introduce six new non-terminals, corresponding to the six possible word orders SVO, SOV, VSO, VOS, OSV and OVS and require every S_{wo_m} non-terminal to be dominated by the non-terminal compatible with its daughters linear order. As an example, consider the two syntactic non-terminals corresponding to the MR kick(pink1), $S_{vs\_kick(pink11)}$ and $S_{sv\_kick(pink11)}$. Whenever an example is successfully analyzed as $S_{vs\_kick(pink11)}$, this should strengthen our model’s expectation of encountering

[12] Because our model parses every sentence, for it Recall and Precision are identical and F-measure is identical to Accuracy.

[13] We assume, somewhat simplifying, that an MR’s predicate corresponds to a V(erb), its first argument corresponds to the S(ubject) and its second argument corresponds to the O(bject). These are purely formal categories that are not constrained to correspond to specific linguistic categories.

$$\begin{array}{ll}
\text{Root} \rightarrow wo & wo \in WO \\
wo \rightarrow S_{x\_m} & wo \in WO, x \in WOS, x \subset wo, m \in M \\
S_{v\_m} \rightarrow c\, Phrase_{p(m)} & c \in C, m \in c, m \in Pred0(M) \\
S_{x\_m} \rightarrow c\, \{Phrase_{p(m)}, Phrase_{a1(m)}\} & c \in C, m \in c, m \in Pred1(M), x \in \{sv, vs\} \\
S_{x\_m} \rightarrow c\, \{Phrase_{p(m)}, Phrase_{a1(m)}, Phrase_{a2(m)}\} & c \in C, m \in c, m \in Pred2(M), x \in WOS \\
S_{v\_\emptyset} \rightarrow c\, Phrase_{\emptyset} & c \in C
\end{array}$$

Figure 4: In order to turn NoWo-PCFG described in Figure 3 into the WO-PCFG described in the text, substitute the first five rule-schemata with the schemata given here. WO is the set of word order non-terminals $\{SVO, SOV, OSV, OVS, VSO, VOS\}$, WOS is the set of word order annotations $\{v, sv, vs, svo, svo, ovs, osv, vso, vos\}$. We take $x \subset wo$ to mean that x is compatible with wo, where v is compatible with all word orders, sv is compatible with SVO,SOV and OSV, and so on. For rule-schemata 4 and 5, the choice of x determines the order of the elements of the set on the right-hand side. All other symbols have the same meaning as explained in Figure 3.

more examples where the verb precedes the subject, i.e. of the language being pre-dominantly VSO, VOS or OVS. Therefore, we allow *VSO*, *VOS* and *OVS* to be rewritten as $S_{vs\_kick(pink11)}$. More generally, every word order non-terminal can rewrite as any of the S_{wo_m} non-terminals that are compatible with it. Adding this additional layer of word order abstraction leads to a grammar with 36,019 rules for English and a grammar with 33,715 rules for Korean.

4.2 Evaluation of WO-PCFG

Training and evaluating WO-PCFG in exactly the same way as the previous grammar gives an F-measure of 0.860 for English and an F-measure of 0.829 for Korean. Those scores are, to our knowledge, the highest scores previously reported for this parsing task and establish our second main point: letting the model learn the language's word order in addition to learning the mapping from sentences to MR increases semantic parsing accuracy.[14]

An intuitive explanation for the increase in performance is that by allowing the model to learn word order, we are providing it with a new dimension along which it can generalize.

In this sense, we can look at our refinement as providing the model with abstract linguistic knowledge, namely that languages tend to have a canonical word order. The usefulness of this kind of information is impressive – for English, it improves the accuracy of semantic parsing by almost 12% in F-measure and for Korean by 11.1%. In addition, our model correctly learns that English's predominant word order is SVO and that Korean is predominantly SOV, assigning by far the highest probability to the corresponding *Root* rewrite rule (0.91 for English and 0.98 for Korean). This kind of information is useful in its own right and could, for example, be exploited by coupling word order with other linguistic properties, perhaps following Greenberg (1966)'s implicational universals.

In this sense, the reduction of grounded learning problems to grammatical inference does not only make possible the application of a wide variety of tools and insights developed over years of research, it might also make it easier to bring abstract (and not so abstract) linguistic knowledge to bear on those tasks.

The overall slightly worse performance of our system on Korean data might stem from the fact that Korean, unlike English, has a rich morphology, and that our model does not learn anything about morphology at all. We plan on further investigating effects like this in the future, as well as applying more advanced grammatical inference algorithms.

5 Conclusion and Future Work

We have shown that certain grounded learning tasks such as learning a semantic parser from semantically enriched training data can be reduced to a grammatical inference problem over strings. This allows

[14] Liang et al. (2009)'s model can be seen as capturing something similar to our word order generalization with the help of a Field Choice Model which primarily captures discourse coherence and salience properties. It differs, however, in that it can only learn one generalization for each predicate type and no language wide generalization.

for the application of techniques and insights developed for grammatical inference to grounded learning tasks. In addition, we have shown that letting the model learn the language's canonical word order improves parsing performance, beyond the top scores previously reported, thus illustrating the usefullnes of linguistic knowledge for tasks like this.

In future research, we plan to address the limitation of our model to a finite set of meaning representations, in particular through the use of nonparametric Bayesian models such as the Infinite PCFG model of Liang et al. (2007) and the Infinite Tree model of Finkel et al. (2007); both allow for a potentially infinite set of non-terminals, hence directly addressing this problem. In addition, we are thinking about using an extension of the PCFG formalism that allows for some kind of 'feature-passing' which could lead to much smaller and more general grammars.

References

N. C. Chang and T. V. Maia. 2001. Grounded learning of grammatical constructions. In *2001 AAAI Spring Symposium on Learning Grounded Representations*.

David L. Chen and Raymond J. Mooney. 2008. Learning to sportscast: A test of grounded language acquisition. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*.

David L. Chen, Joohyun Kim, and Raymond J. Mooney. 2010. Training a multilingual sportscaster: Using perceptual context to learn language. *Journal of Artificial Intelligence Research*, 37:397–435.

Jenny R. Finkel, Trond Grenager, and Christopher D. Manning. 2007. The infinite tree. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 272–279.

Joseph H. Greenberg. 1966. Some universals of grammar with particular reference to the order of meaningful elements. In Joseph H. Greenberg, editor, *Universals of Language*, chapter 5, pages 73–113. The MIT Press, Cambridge, Massachusetts.

Mark Johnson, Katherine Demuth, Michael Frank, and Bevan Jones. 2010. Synergies in learning words and their referents. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1018–1026.

Joohyun Kim and Raymond J. Mooney. 2010. Generative alignment and semantic parsing for learning from ambiguous supervision. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)*.

K. Lari and S.J. Young. 1990. The estimation of Stochastic Context-Free Grammars using the Inside-Outside algorithm. *Computer Speech and Language*, 4(35-56).

Percy Liang, Slav Petrov, Michael Jordan, and Dan Klein. 2007. The infinite PCFG using hierarchical Dirichlet processes. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 688–697.

Percy Liang, Michael I. Jordan, and Dan Klein. 2009. Learning semantic correspondences with less supervision. In *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*.

Wei Lu, Hwee Tou Ng, Wee Sun Lee, and Luke S. Zettlemoyer. 2008. A generative model for parsing natural language to meaning representations. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 783–792.

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*.