Unsupervised Syntax Learning with Categorial Grammars using Inference Rules

Xuchen Yao, Jianqiang Ma, Sergio Duarte and Çağrı Çöltekin
Center for Language and Cognition, University of Groningen

Abstract

We propose a learning method with categorial grammars using inference rules. The proposed learning method has been tested on an artificial language fragment that contains both ambiguity and recursion. We demonstrate that our learner has successfully converged to the target grammar using a relatively small set of initial assumptions. We also show that our method is successful at one of the celebrated problems of language acquisition literature: learning the English auxiliary order.

1. Introduction

Unsupervised learning of natural language grammar is a challenging task. One of the challenges is learning a finite description, a grammar, of an infinite language using a finite amount of input. Besides, human languages are full of ambiguity, which contributes to the challenge of the learning experience. In this paper we present a computational language learner that successfully learns an artificial grammar exhibiting both challenges. The method is based on learning a categorial grammar in an unsupervised fashion.[1]

Categorial Grammar (CG) is a lexicalized grammar formalism with a high level of transparency between syntax and semantics. These features make CG an attractive formalism for computational studies of language acquisition. The lexicalized nature of the CG reduces learning syntax to learning a lexicon, while the close connection between syntax and semantics helps learning one using the other.

One of the earliest studies of CG learners was proposed by Buszkowski & Penn (1989). Their system used unification of type-schemes to determine categorial grammars from functor-argument structures. Kanazawa (1998) extended this algorithm to learn from strings of words. A number of applied studies (e.g. Waldron 1999, Villavicencio 2002, Buttery 2006) followed similar approaches to learn CG based grammars. Waldron (1999) used a rule-based method to infer a CG from input labeled with basic syntactic types. Villavicencio (2002) proposed a method that improves the performance of Waldron's system by describing an unconventional universal grammar based on CG, and using semantically annotated input. Watkinson & Manandhar (2000) presented an unsupervised stochastic learner which aims to learn a compact lexicon. They assumed that the set of possible categories are known, which maps the problem of grammar induction to categorization. The system achieved perfect accuracy in an artificial corpus. However, its performance dropped to 73.2% in lexicon accuracy and 28.5% in parsing accuracy when tested on the more realistic LLL corpus (Kazakov, et al. 1998).

This paper proposes an unsupervised method to learn categorial grammars. The learner is provided with a set of positive sentences generated by a target grammar. Unknown categories are learned by applying a set of inference rules incrementally. When

---

[1]We use the term 'unsupervised' in the sense that the learner is not provided with the information about the structure of the input sentences. Although non-standard, this use of the term is common in computational linguistics literature (e.g. Klein 2005, Watkinson & Manandhar 2000)

there are multiple choices, a *simple category preference* (SCP) principle that is inspired by the MDL principle (Rissanen 1989) is used to minimize the size of the grammar. We intend to develop this algorithm further to learn from real language corpora. However, in this paper we show that the learner is able to infer a recursive and ambiguous artificial grammar and learn the English auxiliary word order from a set of input sentences that are considered insufficient for the task.

The next section gives a short introduction to CG. Section 3 describes our learning architecture. Section 4 presents two experiments and discussion of the results together with limitations of our approach. In the last section we provide brief conclusions and address future directions.

2. Categorial Grammar

Categorial grammar (Ajdukiewicz 1935, Bar-Hillel 1953) is a lexicalized grammar formalism. CG describes all the language specific syntactic information inside the lexicon, leaving only a small number of universal rules outside the lexicon. We present a very brief introduction to CG here, more comprehensive description can be found in Wood (1993).

Every word in a CG lexicon is assigned to a syntactic category. A limited set of categories constitutes the *basic categories* of the grammar. For example, *S* (sentence), *NP* (noun phrase), *N* (noun) are commonly assumed to be the basic categories for English. *Complex categories*, such as *NP/N*, *S\NP*, *(S\NP)\(S\NP)*, are formed by combining any two CG categories with a forward (/) or backward (\) slash. Given the lexicon with categories of this form, the only rules of the CG are given in (1).

(1) *Function application rules*
    Forward application $\quad A/B \quad B \quad\quad \rightarrow \quad A \quad (>)$
    Backward application $\quad B \quad\quad A\backslash B \quad \rightarrow \quad A \quad (<)$

CG as described above is weakly equivalent to Context Free Grammars, and cannot model the complexity of natural languages adequately. However, there are extensions such as *Combinatory Categorial Grammar* (CCG, Steedman 2000) that provide necessary descriptive and theoretical adequacy by introducing additional operations. In this work, we learn classical Categorial Grammars, while making use of some of the CCG operations in (2), namely *composition* and *type raising*, during the learning process.

(2) a. *Function composition rules:*
    Forward $\quad A/B \quad B/C \quad \rightarrow \quad A/C \quad (> \mathbf{B})$
    Backward $\quad B\backslash C \quad A\backslash B \quad \rightarrow \quad A\backslash C \quad (< \mathbf{B})$

b. *Type raising rules:*
    Forward $\quad A \quad \rightarrow \quad T/(T\backslash A) \quad (> \mathbf{T})$
    Backward $\quad A \quad \rightarrow \quad T\backslash(T/A) \quad (< \mathbf{T})$

3. Learning by Inference Rules

In this section we first introduce a series of inference rules used to perform grammar induction. Then we will present the complete learning architecture along with an example demonstrating the learning process.

3.1. Grammar Induction by Inference Rules

Our inference rules work when there is only one unknown category in the input. In the rule descriptions below, the letters *A, B, C* and *D* represent known categories, **X** represents the unknown category.

(3) *Level 0 inference rules:*
$$B/A \quad \mathbf{X} \quad \rightarrow \quad B \quad \Rightarrow \quad \mathbf{X} = A \quad if A \neq S$$
$$\mathbf{X} \quad B\backslash A \quad \rightarrow \quad B \quad \Rightarrow \quad \mathbf{X} = A \quad if A \neq S$$

(4) *Level 1 inference rules:*
$$A \quad \mathbf{X} \quad \rightarrow \quad B \quad \Rightarrow \quad \mathbf{X} = B\backslash A \quad if A \neq S$$
$$\mathbf{X} \quad A \quad \rightarrow \quad B \quad \Rightarrow \quad \mathbf{X} = B/A \quad if A \neq S$$

We define *level* as the number of *functioning* slash operators in a category. Functioning slash operators are functors that take an argument of one type and result in another during the derivation. Consequently, the basic categories are of level 0. The category $S\backslash NP$ belongs to level 1. Note that the category of adverbs $(S\backslash_f NP)\backslash_f(S\backslash NP)$ belongs to level 2. Although it has three slashes, only the slashes marked with subscript $_f$ are functioning, i.e. can be used in a derivation.

Level 0 and level 1 inference rules can be successfully used to learn the category of intransitive verbs, such as *slept* in *Peter slept*. The condition $if \quad A \neq S$ in (3) and (4), prevents learning a large number of incorrect categories.[2] For example, $S\backslash S$ for the word *well* from *Peter slept well*. As stated before, the category of adverbs belongs to level 2, so we need a level 2 inference rule to learn this category.

(5) a. *Level 2 side inference rules:*
$$\mathbf{X} \quad A \quad B \quad \rightarrow \quad C \quad \Rightarrow \quad \mathbf{X} = (C/B)/A$$
$$A \quad B \quad \mathbf{X} \quad \rightarrow \quad C \quad \Rightarrow \quad \mathbf{X} = (C\backslash A)\backslash B$$

   b. *Level 2 middle inference rule:*
$$A \quad \mathbf{X} \quad B \quad \rightarrow \quad C \quad \Rightarrow \quad \mathbf{X} = (C\backslash A)/B$$

Level 2 inference rules are divided into two parts: the *side rule* and the *middle rule*, depending on whether an unknown category is at the beginning/end of a sentence or in the middle.

Notice that in (5b) the category $(C/B)\backslash A$ is as viable as the inferred category $(C\backslash A)/B$. This can be shown by the following example of *left-combining* rule and *right-combining* rule.

(6) a. left-combining *rule:*
$$A \; \mathbf{X} \; B \rightarrow C \; \xrightarrow[divide\ A\mathbf{X}B]{left-combining} \; (A\mathbf{X}) \; B \rightarrow C \; \xrightarrow[rule(4)]{level\,1} \; A\mathbf{X} = C/B \; \xrightarrow{divide\ A\mathbf{X}} \; A \; \mathbf{X} \rightarrow C/B \; \xrightarrow[rule(4)]{level\,1} \; \mathbf{X} = (C/B)\backslash A$$

   b. right-combining *rule:*
$$A \; \mathbf{X} \; B \rightarrow C \; \xrightarrow[divide\ A\mathbf{X}B]{right-combining} \; A \; (\mathbf{X}B) \rightarrow C \; \xrightarrow[rule(4)]{level\,1} \; \mathbf{X}B = C\backslash A \; \xrightarrow{divide\ \mathbf{X}B} \; \mathbf{X} \; B \rightarrow C\backslash A \; \xrightarrow[rule(4)]{level\,1} \; \mathbf{X} = (C\backslash A)/B$$

---

[2]This is against the real world situation since the category $S\backslash S$ is a valid CG category. The problem can be possibly solved by putting S in a less favorite position in the simple category preference principle in later work.

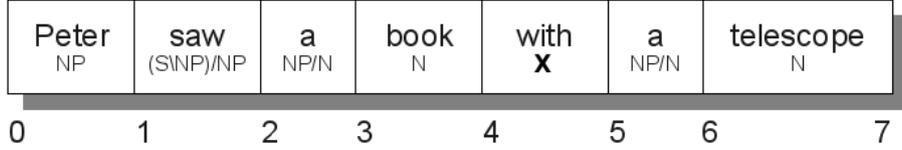| Peter	saw	a	book	with	a	telescope
NP	(S\NP)/NP	NP/N	N	**X**	NP/N	N

0   1   2   3   4   5   6   7

Figure 1: Index in the input string

As shown in the above example, left-combining and right-combining rules produce different but equivalent outputs. Our algorithm uses the right-combining rule when recursively dividing all the entities into two parts and whenever there are possibilities to combine an unknown category with either the left one or the right one, we always combine with the right one.[3]

It might seem that using (5b) we can learn the category of $(S\backslash S)/NP$ for the preposition *with* from the sentence *Peter slept with Mary*. But this will not happen: the level 2 inference rule is implemented by recursively calling level 0 and level 1 inference rules, all of which have the condition $if \quad A \neq S$ to prevent generating the category $S\backslash S$. As a matter of fact, none of the level 0-2 rules could help learning the category of *with* from the sentence *Peter slept with Mary*. So we need to use a level 3 inference rule.

(7) a. *Level 3 side inference rules:*
$$\mathbf{X} \quad A \quad B \quad C \quad \rightarrow \quad D \quad \Rightarrow \quad \mathbf{X} = ((D/C)/B)/A$$
$$A \quad B \quad C \quad \mathbf{X} \quad \rightarrow \quad D \quad \Rightarrow \quad \mathbf{X} = ((D\backslash A)\backslash B)\backslash C$$

  b. *Level 3 middle inference rules:*
$$A \quad \mathbf{X} \quad B \quad C \quad \rightarrow \quad D \quad \Rightarrow \quad \mathbf{X} = ((D\backslash A)/C)/B$$
$$A \quad B \quad \mathbf{X} \quad C \quad \rightarrow \quad D \quad \Rightarrow \quad \mathbf{X} = ((D\backslash A)\backslash B)/C$$

3.2.   The Learning Architecture

The learning framework consists of three parts: *the edge generator*, *the recursive learner* and *the output selector*. A schematic description of the learning process is provided in Figure 2. Below we provide a detailed description of the three parts, along with demonstration of learning the ambiguous and recursive categories of *with* in Figure 1.

The Edge Generator implements a variation of the CYK algorithm, which employs bottom-up chart parsing. Every known word in a sentence is an edge in the chart. The edge generator then tries to merge any consecutive edges into a single edge recursively. In order to produce as many edges as possible, besides function application rules ($>$,$<$), we have also used the composition ($> B,< B$) and the type raising ($> T,< T$) rules. Table 1 shows all possible edges generated for the example in Figure 1.

The Recursive Learner performs grammar induction by the rules given in Subsection 3.1. The learning process first tries to learn from level 0 or level 1 inference rules. If the unknown word cannot be learned by level 0 or level 1 inference rules, higher level rules are tried by recursively dividing all the edges in a sentence into two parts and then calling level 0 or level 1 inference rules to learn (This process is also shown in (6)). Following the *simple category preference* (SCP) principle, if a category can be inferred with a lower level rule, we do not attempt to use higher level rules.

---

[3]We realize that this rule may lead to wrong choices for other languages, and plan to relax it in future work.

4

	A		B		X		C	
	span	rule used	category		span	rule used	category
1	(0, 1)	>T	S/(S\NP)	6	(0, 3)	>B	S/N
2	(0, 2)	>B	S/NP	7	(1, 4)	>	S\NP
3	(1, 3)	>B	(S\NP)/N	8	(2, 4)	<T	S/(S\NP)
4	(2, 4)	>	NP	9	(0, 4)	<	S
5	(5, 7)	>	NP	10	(0, 4)	>	S

Table 1: Generated edges in a chart

	A		B		X		C	
	cat	span	cat	span	cat	span	cat	span
1	NP	(0, 1)	S\NP	(1, 4)	**((S\NP)\(S\NP))/NP**	(4,5)	NP	(5, 7)
2	S/(S\NP)	(0, 1)	S\NP	(1, 4)	((S\NP)\(S\NP))/NP	(4,5)	NP	(5, 7)
3	S/NP	(0, 2)	NP	(2, 4)	**(NP\NP)/NP**	(4,5)	NP	(5, 7)
4	S/NP	(0, 2)	S/(S\NP)	(2, 4)	(NP\(S/(S\NP)))/NP	(4,5)	NP	(5, 7)
5	S/N	(0, 3)	N	(3, 4)	**(N\N)/NP**	(4,5)	NP	(5, 7)

Table 2: Categories learned from the rule $A \quad B \quad \mathbf{X} \quad C \rightarrow S$
for the sentence in Figure 1.

For the input in Figure 1, the level 0 and level 1 inference rules are not enough. Only the level 3 middle inference rules (7b) can be applied. Table 2 gives a list of all the possible categories using this inference rule.

The Output Selector tests the learned categories produced by the recursive learner and selects the ones that can be parsed using only function application rules. The categories that do not produce a valid parse with function application rules are discarded.

In Table 2, the sentence cannot be parsed using the category in row 4, so this category is discarded. Rows 1 (or equal category in row 2), 3 and 5 provide the learned categories.

4. Experiments and Results

We conducted two experiments with our learning system. In the first experiment, we tested the system's capabilities on an artificial language exhibiting a certain level of ambiguity and recursion. In the second experiment, we tried to learn the English auxiliary order, a well known problem in language acquisition literature.

4.1. Experiment 1: Learning an Artificial Grammar

For this experiment, we have created a small English-like artificial grammar. The lexicalized grammar that is used as the target grammar for this experiment is listed in Table 3. The input to the learner consists of 160 sentences (2 to 7 words in length) generated by the target grammar. Only correct sentences are used. The input sentences are unlabeled, except for nouns (N) and noun phrases (NP). Thus the learner first searches sentences with only one unknown word and tries to learn this word. Then it takes into account the learned category and searches for other sentences with unknown words. Using this
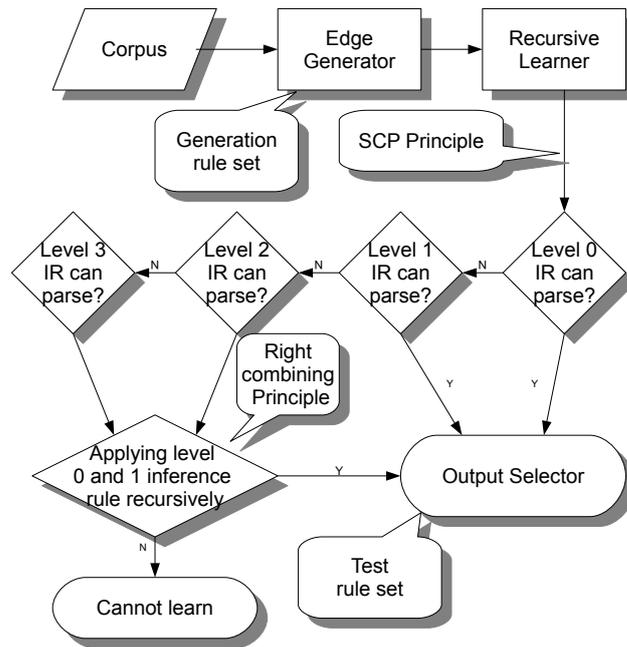
Figure 2: Learning process using inference rules

Peter	:= NP	*Mary*	:= NP	*with*	:= (N\N)/NP
book	:= N	*green*	:= N/N	*with*	:= ((S\NP)\(S\NP))/NP
colorless	:= N/N	*sleep*	:= S\NP	*furiously*	:= (S\NP)\(S\NP)
a	:= NP/N	*telescope*	:= N	*give*	:= ((S\NP)/NP)/NP
the	:= NP/N	*saw*	:= (S\NP)/NP	*read*	:= (S\NP)/NP
run	:= S\NP				

Table 3: Target grammar rules

"bootstrap"-like method the learner is expected to converge to the target grammar.

After only a single pass through input sentences, all categories in our target grammar presented in Table 3 are learned correctly. The learned grammar includes only one lexical item (*with* := $(NP\backslash NP)/NP$) that is not in the target grammar. This, however, is a useful generalization which allows deriving structures like *[Peter [saw [Mary [with [a telescope]]]]]*, while our original grammar does not.

4.2. Experiment 2: Learning Correct Word Order

The difficulty of learning English auxiliary order has also been used as a support for the poverty of the stimulus (POTS) argument, and hence for linguistic nativism. Introduced first by Kimball (1973), the problem can be summarized as follows: the English auxiliary verbs *should*, *have* and *be* occur exactly in this order and all of them are optional. The claim is that while sentences containing a single auxiliary (8a–8c) or two auxiliaries (8d–8f) are present in the input, sequences of three auxiliaries (8g) are not frequent enough. Hence, it is not possible to learn the correct three-auxiliary sequence from the input alone.

6

$$should := (S_s \backslash NP)/(S \backslash NP)$$
$$should := (S_s \backslash NP)/(S_h \backslash NP)$$
$$should := (S_s \backslash NP)/(S_b \backslash NP)$$
$$have \quad := (S_h \backslash NP)/(S \backslash NP)$$
$$have \quad := (S_h \backslash NP)/(S_b \backslash NP)$$
$$be \qquad := (S_b \backslash NP)/(S \backslash NP)$$

Table 4: Categories of some auxiliary verbs.

(8) a. *I should go.*
 b. *I have gone.*
 c. *I am going.*
 d. *I have been going.*
 e. *I should have gone.*
 f. *I should be going.*
 g. *I should have been going.*
 h. **I have should been going.*

The argument is controversial, and Pullum & Scholz (2002) have shown that there are more three-auxiliary sequences in children's input than claimed. In this study, we choose another approach: we present our learner with sentences containing only one or two auxiliaries (as in (8a-8f)), and we test if it can correctly recognize and generate sentences with three auxiliaries. The experiment setting is the same as in experiment 1. The only additional information provided is the type of sentences, i.e. every given input is marked with the "mood of the sentence". As well as simple declarative sentences (S), we used S_b, S_h and S_s for sentences with modal verbs *be*, *have* and *should* respectively.

Table 4 presents a fragment of the learned grammar. The derivation of the sentence (8g) using the learned grammar is given in Figure 3. As can be verified easily, the lexicalized grammar presented in Table 4 would not allow sequences as in (8h). The categories assigned to auxiliary verbs by the learner completely, and correctly derive the English auxiliary order.[4]

Success of the learner is again due to its assignment of words to syntactic categories. The categories induced from one- and two-auxiliary sequences in a logical way extend naturally to three-auxiliary sequences.

4.3. Discussion

We have presented a learner that learns syntax using CG. One of the characteristics of our method is that it learns from input without any structure, semantic annotation or negative evidence. Although there are theoretical results about learnability on only strings (Kanazawa 1998), and more applicable research about learning from sentences annotated with structures (Villavicencio 2002, Buttery 2006), applied work on learning from strings

---

[4]An alternative approach would be assuming that the sequences like '*should have been*' are learned as single units, at least at the beginning of the learning process. Lexical items spanning multiple input units are considered by some of the related learners (e.g. Zettlemoyer & Collins 2005, Çöltekin & Bozsahin 2007). However, to be compatible with the original claim, the learner presented in this paper assigns categories to single input units.

$$\frac{\text{I} \quad \text{should} \qquad \text{have} \qquad\qquad \text{been} \qquad \text{going.}}{\overline{NP}\ \overline{(S_s\backslash NP)/(S_h\backslash NP)}\ \overline{(S_h\backslash NP)/(S_b\backslash NP)}\ \overline{(S_b\backslash NP)/(S\backslash NP)}\ \overline{S\backslash NP}}$$
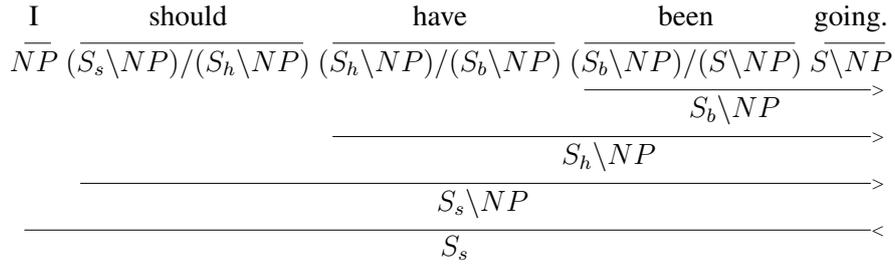
Figure 3: Derivation of the correct word order.

is rather limited. This study is our first attempt to fill this gap. Although the experiments are based on artificial data, our aim is to further develop the method and apply it on real-world linguistic input.

The simple inference rules used here are admittedly ad-hoc and we have not yet attempted to provide the guarantee of convergence. Our main goal with this method is to experiment with the possibilities of exploiting the information in the linguistic input, rather than to find a learning algorithm that is guaranteed to learn in a wide range of input distributions. For the fragment of the English grammar captured by our artificial language learning experiments, the results are promising.

The method is in essence similar to unification based learner of Buszkowski & Penn (1989), which learns from structurally annotated input. Unfortunately, Kanazawa's extension of the algorithm to learn from strings is computationally intractable. The use of ad-hoc rules and constraints instead of standard learning framework is motivated by the aim of using of a reasonable amount of input and computational resources.

The input to our learner is partially annotated. This approach carries an affinity to the *partial learning* system described by Moreau (2004). However, crucially, the annotation provided to our learner does not contain any structure. Moreau (2004) especially makes use of high-frequency closed-class words with categories that give hints about the structure of the input sentences. This is useful in practical computational linguistic applications, as it helps inducing a grammar with relatively small amount of annotation. However, our approach is more plausible for language acquisition, as children are known to learn nouns earlier than other word classes (Gentner 1982).

Another apparent limitation of our learner is that it only learns from the input that contains only one unknown word. This avoids the combinatorial expansion of hypothesized categories and keeps the required computational resources low, and this worked fine for our artificial grammar learning experiments.[5] For language acquisition, this is not a wildly wrong assumption. We assume that the children do not understand and, hence, make use of complicated input at first sight. However, the category of the word can still be inferred, when the same word later appears in an understandable context.

The output selector only selects the categories that can be parsed by the AB grammars. This could lead to inconsistency with the target grammar: if the target grammar can only be parsed by more complicated CCG rules while the output selector only uses AB rules, or if the target grammar can be parsed by AB rules while the output selector uses CCG

---

[5]It should also be noted that, the algorithm can be adapted to use the input with 'k unknown words' with the expense of additional computational resources.

rules. Here we do not think there will be big problems: the initial setting is to use AB rules, when there are no candidates under AB rules, the output selector adjusts to more rules to produce an output. The simple category preference guarantees that no matter how complex rules are used, the output category will stay as simple as possible.

The inference rules are simple and even intuitive. Although there is no solid psychological evidence that children learn a grammar in an inference-rule-based way, the 100% correct results in parsing and generation by our model suggests that it is sufficient to assume that children use a small set of rules together with a plausible inference procedure for learning the categories of unknown words. The only other additional piece in our learning algorithm is a preference towards simpler categories.

This method performs well on learning a typical language phenomenon such as learning English auxiliary order. We have shown that only being exposed to one- and two-auxiliary sequences, our simple algorithm generalized correctly to sentences containing three auxiliary verbs. Even if the POTS claims are correct, children can still learn correct forms with simple inference mechanisms.

5. Conclusion

We described a method to learn categorial grammars using inference rules. Our method has learned all the categories of the target grammar. We use simple logical and intuitive inference rules to solve the problem of unknown categories in the input. The only additional aid provided to our learner is the simple category preference. Using only this set of initial assumptions, our system is also able to learn a phenomenon that has been considered difficult. The learner is able to infer English auxiliary order correctly without being presented with all possible sequences.

However, it is necessary to note that our system has a number of limitations. First, these results were obtained using data that was generated artificially. Second, since we do not use any statistical inference mechanism, our system is not robust against noise. Using statistical patterns in the input language, it may also be possible to relax some of the assumptions presented here. This is a limitation when the amount of data is not large enough to "bootstrap" the learner.

Future work includes developing the algorithm further and evaluating it on real data, such as child-directed speech from CHILDES database (MacWhinney 2000).

References

K. Ajdukiewicz (1935). 'Die syntaktische Konnexität'. *Studia Philosophica* **1**:1–27.

Y. Bar-Hillel (1953). 'A Quasi-Arithmetical Notation for Syntactic Description'. *Language* **1**:47–58.

W. Buszkowski & G. Penn (1989). 'Categorial grammars determined from linguistic data by unification'. Tech. rep., Chicago, IL, USA.

P. J. Buttery (2006). 'Computational models for first language acquisition'. Tech. rep., University of Cambridge, Churchill College.

Ç. Çöltekin & C. Bozsahin (2007). 'Syllables, Morphemes and Bayesian Computational Models of Acquiring a Word Grammar'. In *Proceedings of 29th Annual Meeting of Cognitive Science Society*, Nashville.

D. Gentner (1982). 'Why nouns are learned before verbs: Linguistic relativity versus natural partitioning.'. In S. Kuczaj (ed.), *Language development*, vol. 2. Erlbaum, Hillsdale, NJ.

M. Kanazawa (1998). *Learnable classes of categorial grammars*. Cambridge University Press, New York, NY, USA.

D. Kazakov, et al. (1998). 'The FraCas dataset and the LLL challenge'. Tech. rep., SRI International.

J. P. Kimball (1973). *The Formal Theory of Grammar*. Englewood Cliffs, NJ: Prentice-Hall.

D. Klein (2005). *The Unsupervised Learning of Natural Language Structure*. Ph.D. thesis, Stanford University.

B. MacWhinney (2000). *The CHILDES project: tools for analyzing talk*. Lawrence Erlbaum, Mahwah, NJ u.a. EN.

E. Moreau (2004). 'Partial Learning Using Link Grammars Data'. In G. Paliouras & Y. Sakakibara (eds.), *Grammatical Inference: Algorithms and applications. 7th International Colloquium: ICGI 2004*, vol. 3264 of *Lectures Notes in Artificial Intelligence*, pp. 211–222, Athens, Greece. Springer.

G. K. Pullum & B. C. Scholz (2002). 'Empirical assessment of stimulus poverty arguments'. *The Linguistic Review* **19**:9–50.

J. Rissanen (1989). *Stochastic Complexity in Statistical Inquiry Theory*. World Scientific Publishing Co., Inc., River Edge, NJ, USA.

M. Steedman (2000). *The syntactic process*. MIT Press, Cambridge, MA, USA.

A. Villavicencio (2002). *The acquisition of a unification-based generalised categorial grammar*. Ph.D. thesis, University of Cambridge.

B. Waldron (1999). 'Learning grammar from corpora'. Master's thesis, Cambridge University.

S. Watkinson & S. Manandhar (2000). 'Unsupervised lexical learning with Categorial Grammars using the LLL corpus'. In *Learning language in logic*, pp. 218–233. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

M. M. Wood (1993). *Categorial Grammars*. Routledge, London.

L. S. Zettlemoyer & M. Collins (2005). 'Learning to Map Sentences to Logical Form: Structured Classification with Probabilistic Categorial Grammars'. In *Proceedings of the Twenty First Conference on Uncertainty in Artificial Intelligence (UAI-05)*.