In: *Proceedings of CoNLL-2000 and LLL-2000*, pages 209–218, Lisbon, Portugal, 2000.

The Acquisition of Word Order by a Computational Learning System

Aline Villavicencio
Computer Laboratory, University of Cambridge
New Museums Site, Cambridge, CB2 3QG, England, UK
Aline.Villavicencio@cl.cam.ac.uk

Abstract

The purpose of this work is to investigate the process of grammatical acquisition from data. We are using a computational learning system that is composed of a Universal Grammar with associated parameters, and a learning algorithm, following the Principles and Parameters Theory. The Universal Grammar is implemented as a Unification-Based Generalised Categorial Grammar, embedded in a default inheritance network of lexical types. The learning algorithm receives input from a corpus annotated with logical forms and sets the parameters based on this input. This framework is used as basis to investigate several aspects of language acquisition. In this paper we are concentrating on the acquisition of word order for different learners. The results obtained show the different learners having a similar performance and converging towards the target grammar given the input data available, regardless of their starting points. It also shows how the amount of noise present in the input data affects the speed of convergence of the learners towards the target.

1 Introduction

In trying to solve the question of how to get a machine to automatically learn linguistic information from data, we can look at the way people do it. Gold (1967) when investigating language identification in the limit, obtained results that implied that natural languages could not be learned only on the basis of positive evidence. These results were used as a confirmation for the proposal that children must have some innate knowledge about language, the Universal Grammar (UG), to help them overcome the problem of the poverty of the stimulus and acquire a grammar on the basis of positive evidence only. According to Chomsky's Principles and Parameters Theory (Chomsky 1981), the UG is composed of principles and parameters, and the process of learning a language is regarded as the setting of values of a number of parameters, given exposure to this particular language. We employ this idea in the learning framework implemented.

In this work we are interested in investigating the acquisition of grammatical knowledge from data, focusing on the acquisition of word order, that reflects the underlying order in which constituents occur in different languages (e.g. SVO and SOV languages). The learning system is equipped with a UG and associated parameters, encoded as a Unification-Based Generalised Categorial Grammar, and a learning algorithm that fixes the values of the parameters to a particular language. The learning algorithm follows the Bayesian Incremental Parameter Setting (BIPS) algorithm (Briscoe 1999), and when setting the parameters it uses a Minimum Description Length (MDL) style bias to choose the most probable grammar that describes the data well, given the goal of converging to the target grammar. In section 2 we describe the components of the learning system. In section 3, we investigate the acquisition of word order within this framework and discuss the results obtained by different learners. Finally we present some conclusions and future work.

2 The Learning System

The learning system is composed of a language learner equipped with a UG and a learning algorithm that updates the initial parameter settings, based on exposure to a corpus of utterances. Each of these components is discussed in more detail in the following sections.

2.1 The Universal Grammar

The UG consists of **principles** and **parameters**, and the latter are set according to the linguistic environment (Chomsky 1981). This proposal suggests that human languages follow a common set of principles and differ among one another only in finitely many respects, represented by a finite number of parameters that can vary according to a finite number of values (which makes them learnable in Gold's paradigm). In this section, we discuss the UG and associated parameters, which are formalised in terms of a Unification-Based Generalised Categorial Grammar (UB-GCG), embedded in a default inheritance network of lexical types. We concentrate on the description of word order parameters, which reflect the basic order in which constituents occur in different languages.

UB-GCGs extend the basic Categorial Grammars ((Bar Hillel, 1964)) by including the use of attribute-value pairs associated with each category and by using a larger set of rules and operators. Words, categories and rules are represented in terms of typed default feature structures (TDFSs), that encode orthographic, syntactic and semantic information. There are two types of categories: atomic categories (**s - sentence-, np - noun phrase-, and n - noun**), that are saturated, and complex categories, that are unsaturated. Complex categories have a functor category (defined in RESULT), and a list of subcategorised elements (defined in ACTIVE), with each element in the list defined in terms of two features: SIGN, encoding the category, and DIRECTION, encoding the direction in which the category is to be combined (where VALUE can be either **forward** or **backward**). As an example, in English an intransitive verb (s\np) is encoded as shown in figure 1, where only the relevant attributes are shown. In this work, we employ the rules of (forward and backward) application, (forward and backward) composition and generalised weak permutation. A more detailed description of the UB-GCG used can be found in (Villavicencio 2000).

The UG is implemented as a UB-GCG, embedded in a default inheritance network of lexical types (Villavicencio 1999), implemented in the YADU framework (Lascarides and Copestake 1999). The categories and rules in the grammar are defined as types in the hierarchy, represented in terms of TDFSs and the feature-structures associated with any given category or rule are defined by the inheritance chain. With different sub-networks used to encode different kinds of linguistic knowledge, linguistic regularities are encoded near the top of a network, while types further down the network are used to represent sub-regularities or exceptions. Thus, types are concisely defined, with only specific information being described, since more general information is inherited from the supertypes. The resulting UB-GCG is compact, since it avoids redundant specifications and the information is structured in a clear and concise way through the specification of linguistic regularities and sub-regularities and exceptions.

Regarding the categories of the UB-GCG, word order parameters are those that specify the direction of each element in the subcategorisation list of a complex category. In figure 1, **subjdir** is a parameter specifying that the np subject is to be combined backwards. As the categories are defined in terms of an inheritance hierarchy, the parameters (and their values) in these categories are propagated throughout the hierarchy, from supertypes to subtypes, which inherit this information by default. There are 28 parameters defined, and they are also in a hierarchical relationship, with the supertype being **gendir**, which specifies, by default, the general direction for a language, and from which all the other parameters inherit. Among the subtypes, we have **subjdir**, which specifies the direction of the subject, **vargdir**, which specifies the direction of the other verbal arguments and **ndir**, which specifies the direction of nominal categories. A fragment of the parameters hierarchy can be seen in figure 2. With these 28 binary-valued parameters the UG defines a space of almost 800 grammars.

The parameters are set based on exposure to a particular language, and while they are unset, they inherit their value by default, from their supertypes. Then, when they are set, they can either continue to inherit by default, in case they have the same value as the supertype, or they can override this default and specify their own value, breaking the inheritance chain. For instance, in the case of English, the value of

$$\begin{bmatrix}\textbf{intransitive}\\ \text{RESULT}: \begin{bmatrix}\text{SIGN}:\textbf{s}\end{bmatrix}\\ \text{ACTIVE}: \begin{bmatrix}\begin{bmatrix}\text{SIGN}:\textbf{np}\\ \text{DIRECTION}:\begin{bmatrix}\textbf{subjdir}\\ \text{VALUE}:\textbf{backward}\end{bmatrix}\end{bmatrix}\end{bmatrix}\end{bmatrix}$$

Figure 1: Intransitive Verb type

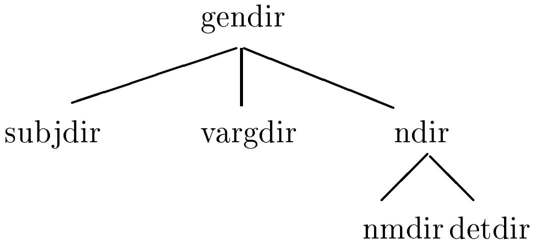

Figure 2: Fragment of The Parameters Hierarchy

gendir is defined, by default, as **forward**, capturing the fact that it is a predominantly right-branching language, and all its subtypes, like **subjdir** and **vargdir** inherit this default information. Then an intransitive verb, which has the direction of the subject specified by **subjdir**, will be defined as S/NP, with **subjdir** having default value **forward**. However, as in English, the subject NP occurs to the left of the verb, utterances with the subject to the left will trigger a change in **subjdir** to **backward**, which overrides the default value, breaking the inheritance chain, figure 3. As a result, intransitive verbs are defined as S\NP, figure 1, for the grammar to account for these sentences. In the syntactic dimension of this network, intransitive verbs can be considered the general case of verbs, and the information defined in this node is propagated through the hierarchy to its subtypes, such as the transitive verbs, figure 3. For the learner, the information about subjects (**subjdir** = **backward**) has already been acquired while learning intransitive verbs, and the learner does not need to learn it again for transitive verbs, which not only inherit this information, but also have the direction for the object defined by **vargdir** (**vargdir** = **forward**), as shown in figure 3. The use of a default inheritance schema reduces the pieces of information to be acquired by the learner, since the information is structured and what it learns is not a single isolated category, but a structure that represents this information in a general manner. This is a clear and concise way of defining the UG with the parameters being straightforwardly defined in the categories, in a way that takes advantage of the default inheritance mechanism, to propagate information about parameters, throughout the lexical inheritance network.

2.2 The Corpus

The UG has to be general enough to capture the grammar for any language, and the parameters have to be set to account for a particular language, based on exposure to that language. This can be obtained by means of a corpus of utterances, annotated with logical forms, which is described in this section. Among these sentences, some will be triggers for certain parameters, in the sense that, to parse that sentence, some of the parameters will have to be set to a given value. We are using the Sachs corpus (Sachs 1983) from the CHILDES project (MacWhinney 1995), that contains interactions between only one child and her parents, from the age of 1 year and 1 month to 5 years and 1 month. From the resulting corpus, we extracted material for generating two different corpora: one containing only the child's sentences and the other containing the caretakers' sentences. The caretakers' corpus is given as input to the learner to mirror the input to which a child learning a language is exposed. And the child's corpus is used for comparative purposes.

In order to annotate the caretakers' corpus with the associated logical forms, a UB-GCG for English was built, that covers all the constructions in the corpus: several verbal con-

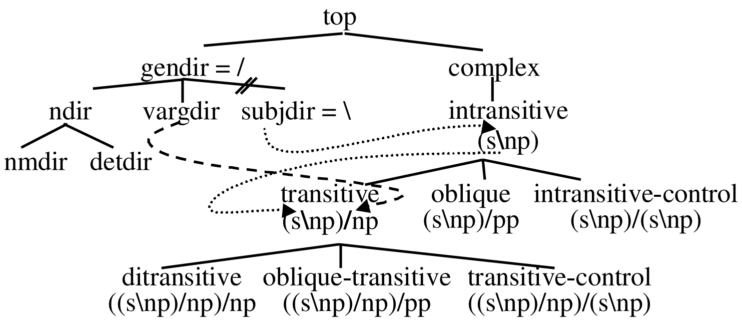

Figure 3: A Fragment of the Network of Types

structions (intransitives, transitives, ditransitives, obliques, control verbs, verbs with sentential complements, etc), declarative, imperative and interrogative sentences, and unbounded dependencies (wh-questions and relative clauses), among others. Thus the caretakers' corpus contains sentences annotated with logical forms, and an example can be seen in figure 4, for the sentence *I will take him*, where a simplified version of the relevant attributes is shown, for reasons of clarity. Each predicate in the semantics list is associated with a word in the sentence, and, among other things, it contains information about the identifier of the predicate (SIT), the required arguments (e.g. ACTOR and UNDERGOER for the verb *take*), as well as about the interaction with other predicates, specified by the boxed indices (e.g. *take*:ACTOR = $\boxed{3}$ = *i*:SIT). This grammar is not only used for annotating the corpus, but is also the target to which the learner has to converge. At the moment around 1,300 utterances were annotated with corresponding logical forms, with data ranging from when the child is 14 months old to 20 months old.

2.3 The Learning Algorithm

The learning algorithm implements the Bayesian Incremental Parameter Setting (BIPS) algorithm defined by Briscoe (1999). The parameters are binary-valued, where each possible value in a parameter is associated with a prior and a posterior probability. The value with highest posterior probability is used as the current value. Initially, in the learning process, the posterior probability associated with each parameter is initialised to the prior probability, and these values are going to define the parameter settings used. Then, as trigger sentences are successfully parsed, the posterior probabilities of the parameter settings that allowed the sentence to be parsed are reinforced. Otherwise, when a sentence cannot be parsed (with the correct logical form) the learning algorithm checks if a successful parse can be achieved by changing the values of some of the parameters, in constrained ways. If that is the case, the posterior probability of the values used are reinforced in each of the parameters, and if they achieve a certain threshold, they are retained as the current values, otherwise the previous values are kept. This constraint on the setting of the parameters ensures that a trigger does not cause an immediate change to a different grammar. The learner, instead, has to wait for enough evidence in the data before it can change the value of any parameter. As a consequence, the learner behaves in a more conservative way, being robust to noise present in the input data.

Following Briscoe (1999) the probabilities associated with the parameter values correspond to weights represented in terms of fractions, with the denominator storing the total evidence for a parameter and the numerator storing the evidence for a particular value of that parameter. For instance, if the value **backward** of the **subjdir** parameter has a weight of 9/10, it means that from 10 times that evidence was provided for **subjdir**, 9 times it was for the value **backward**, and only once for the other value, **forward**. Table 1 shows a possible initialisation for the **subjdir** parameter, where the prior has a weight of 1/10 for **forward**, corresponding to a probability of 0.1, and a weight of

$$\begin{bmatrix} \textbf{sign} \\ \text{ORTH} : \langle \textbf{i, will, take, him} \rangle \\ \text{CAT} : \textbf{s} \\ \text{SEM} : \left[\begin{bmatrix}\textbf{will}\\ \text{SIT}:\boxed{1} \\ \text{ARGUMENT}:\boxed{2}\end{bmatrix}, \begin{bmatrix}\textbf{take}\\ \text{SIT}:\boxed{2}\\ \text{ACTOR}:\boxed{3}\\ \text{UNDERGOER}:\boxed{4}\end{bmatrix}, \begin{bmatrix}\textbf{i}\\ \text{SIT}:\boxed{3}\end{bmatrix}, \begin{bmatrix}\textbf{him}\\ \text{SIT}:\boxed{4}\end{bmatrix} \right] \end{bmatrix}$$

Figure 4: Sentence: I will take him

9/10 for **backward**, corresponding to a probability of 0.9. The posterior is initialised with the same values as the prior, and as **backward** has a higher posterior probability it is used as the current value for the parameter. These initial parameter values determine the initial grammar for the learner. As triggers are processed, they provide evidence for certain parameters and these are represented as additions to the denominator and/or numerator of each of the posterior weights of the parameter values. Table 2 shows the status of the parameter after 5 triggers that provided evidence for the value **backward**. Initially, the learner uses the evidence provided by the triggers to choose certain parameter values, in order to be able to parse these triggers successfully while generating the appropriate logical form. After that, the triggers are used to reinforce these values, or to negate them.

Table 1: Initialisation of a Parameter

Value	Prior		Posterior	
	Prob.	Weight	Prob.	Weight
Forward	0.1	$\frac{1}{10}$	0.1	$\frac{1}{10}$
Backward	0.9	$\frac{9}{10}$	0.9	$\frac{9}{10}$

The 28 word order parameters are defined in a hierarchical relation, with the supertype parameters being set in accordance with the subtypes, to reflect the value of the majority of the subtypes. In this way, as the values of the subtypes are being set, they influence the value of the supertypes. If the value of a given subtype differs from the value of the supertype, the subtype overrides the inherited default value and specifies its own value, breaking the inheritance chain. For instance, in figure 3, **subjdir** overrides the default value specified by **gendir**, breaking the inheritance chain. Unset subtype parameters inherit, by default, the current value of their supertypes, and while they are unset they do not influence the values of their supertypes.

As the parameters are defined in a default inheritance hierarchy, each time the posterior probability of a given parameter is updated, it is necessary to update the posterior probabilities of its supertypes and examine the current parameter settings to determine what the most appropriate hierarchy for these settings is, given the goal of converging to the target. The learner has a preference for grammars (and thus hierarchies) that not only model the data (represented by the current settings) well, but are also compact, following the Minimum Description Length (MDL) Principle. In this case, the most probable grammar in the grammar space, among the ones consistent with the parameter settings, is the one where the default inheritance hierarchy is the more concise, having the minimum number of non-default parameter values specified, as described in (Villavicencio 2000).

Table 2: Status of the Parameter

Value	Prior		Posterior	
	Prob.	Weight	Prob.	Weight
Forward	0.1	$\frac{1}{10}$	0.07	$\frac{1}{15}$
Backward	0.9	$\frac{9}{10}$	0.93	$\frac{14}{15}$

3 The Acquisition of Word Order

We are investigating the acquisition of word order, which reflects the underlying order in which constituents occur in different languages. In this section we describe one experiment, where we compare the performance of different learners under four conditions. Each learner is given as input the annotated corpus of sentences paired with logical forms, and they have to change the values of the parameters corresponding to the relevant constituents to account for the order in which these constituents appear in the input sentences. We defined five different learners corresponding to five different initialisations of the parameter settings of the UG, to investigate how the initialisations, or starting points, of the learners influence convergence to the target grammar. The first one, the unset learner, is initialised with all parameters unset, and the others, the default learners, are each initialised with default parameter values corresponding to one of four basic word orders, defined in terms of the canonical order of the verb (V), subject (S) and objects (O): SVO, SOV, VSO and OVS. We initialised the parameters **subjdir**, **vargdir** and **gendir** of the default learners according to each of the basic orders, with **gendir** having the same direction as **vargdir**, and all the other parameters having unset values. These parameters have the prior and posterior probabilities initialised with 0.1 for one value and 0.9 for the other. In this way, an SVO learner, for example, is initialised with **subjdir** having as current value backward (0.9), **vargdir** forward (0.9) and **gendir** forward (0.9).

The sentences in the input corpus are presented to a learner only once, sequentially, in the original order. The input to a learner is pre-processed by a system [Waldron, 2000] that assigns categories to each word in a sentence. The sentences with their putative category assignments are given as input to the learner. The learner then evaluates the category assignments for each sentence and only uses those that are valid according to the UG to set the parameters; the others are discarded. The corpus contains 1,041 English sentences (which follow the SVO order), but from these only a small proportion are triggers for the parameters, in the sense that, for the learner to process them, it has to select certain parameter values. As each triggering sentence is processed, the learner changes or reinforces its parameter values to reflect the order of constituents in these sentences.

We wanted to check how the different learners performed in a normal noisy environment, with a limited corpus as input, and also to check if there is an interaction between the different initialisations and the noise in the input data. To do that we tested how the learners performed under four conditions. Each condition was run 10 times for each learner, and we report here the average results obtained.

3.1 Condition 1: Learners-10 in a Noisy Environment

In the first condition, we initialised the parameters **subjdir**, **vargdir** and **gendir** of the default learners with the prior and posterior probabilities of 0.1 corresponding to a weight of 1/10, and probabilities of 0.9 to a weight of 9/10. Results from the first experiment can be seen in table 3, where the learners are specified in the first column, the number of input triggers in the second, the number of correct parameters in relation to the target is in the third, and the number of parameters that are set with these triggers is in the fourth column.

Table 3: Convergence of the different learners - Learners-10

Learners	Triggers	Parameters Correct	Parameters Set
Unset	179	22.3	10.5
SVO-10	211.4	22.5	11
SOV-10	205.4	22.2	10.2
OVS-10	271.5	22.5	11
VSO-10	198.7	22.1	10.2

The results show no significant variation in the performance of the different Learners. This is the case with the number of parameters that are correct in relation to the target, with an average of 22.3 parameters out of 28, and also with the number of parameters that are set given the triggers available, with an average of 10.5 parameters out of 28.

The only difference between the learners was

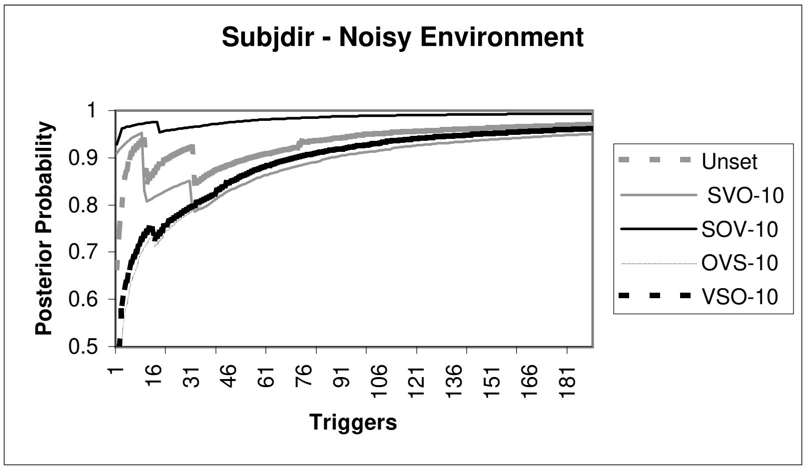

Figure 5: Convergence of Subjdir - Learners-10 - Noisy Environment

the time needed for each learner to converge: the closer the starting point of the learner was to the target, the faster it converged, as can be seen in figure 5, for the **subjdir** parameter. This figure shows all the learners converging to the target value, with high probability, and with a convergence pattern very similar to the one presented by the unset learner. Even those default learners that were initialised with values incompatible with the target soon overcame this initial bias and converged to the target. The same thing happens for **vargdir** and **gendir**. This figure also shows some sharp falls in the convergence to the target value, for these learners. For example, the unset learner had a sharp drop in probability, which fell from 0.94 to 0.85, around trigger 16. These declines were caused by noise in the category assignments of the input triggers, which provided incorrect evidence for the parameter values.

3.2 Condition 2: Learners-10 in a Noise-free Environment

In order to test if and how much of the learners' performance was affected by the presence of noisy triggers, using the same initialisations as the ones in condition 1, we tested how the learners performed in a noise-free environment. To obtain such an environment, as each trigger was processed, a module was used for correcting the category assignment, if noise was detected. The results are shown in table 4.

Table 4: Convergence of the different learners - Learners-10 - Noise-free

Learners	Triggers	Parameters Correct	Parameters Set
Unset	235.1	22.3	10.6
SVO-10	227.9	22.3	10.6
SOV-10	213.9	22.6	11.2
OVS-10	212.2	22.3	10.6
VSO-10	172.4	22	10

These learners have performances similar to those in condition 1 (section 3.1), with an average of 22.3 of the 28 parameters correct in relation to the target, and an average of 10.6 parameters that can be set with the triggers available. But, in this condition the convergence was slightly faster for all learners, as can be seen in figure 6. These results show that, indeed, the presence of noise slows down the convergence of the learners, because they need more triggers to compensate for the effect produced by the noisy triggers.

3.3 Condition 3: Learners-50 in a Noisy Environment

We then tested if the use of stronger weights to initialise the learners would affect the learners performance. The parameters **subjdir**, **vargdir** and **gendir** were initialised with a weight of 5/50 for the probability of 0.1 and a

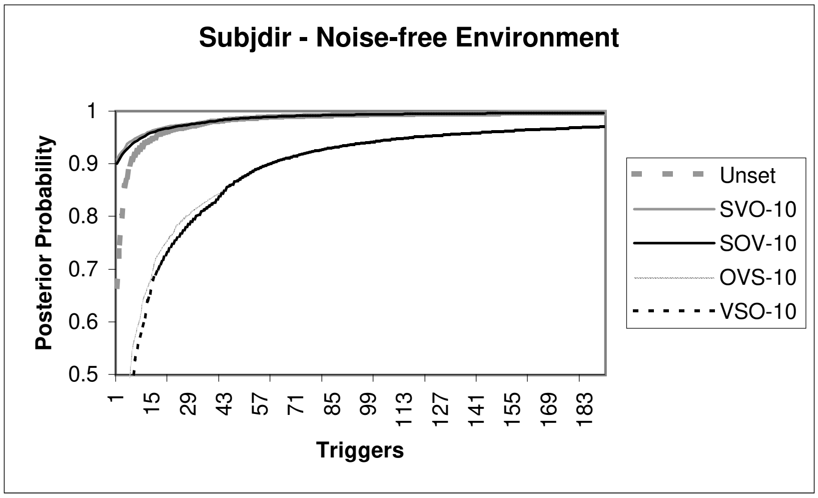

Figure 6: Convergence of Subjdir - Learners-10 - Noise-free Environment

weight of 45/50 for the probability of 0.9. These weights provide an extreme bias for each of the learners. In this condition, the learners were tested again in a normal noisy environment.

Figure 7 shows the convergence patterns presented by these learners for the **subjdir** parameter. The effect produced by the noise was increased with these stronger weights, such that all the learners had a slower convergence to the target. Even those default learners initialised with values compatible with the target had a slightly slower convergence when compared to those in condition 1, with weaker weights, because they had to overcome the stronger initial bias before converging to the target values. But, in spite of that, the performance of the learners is only slightly affected by the stronger weights, as shown in table 5. They had a performance similar to the ones obtained by the learners in the previous conditions, as shown in figure 8, comparing these learners with those in condition 1.

3.4 Condition 4: Learners-50 in a Noise-free Environment

When the noise-free environment was used with these stronger weights, the convergence pattern was slightly faster for all learners, when compared to condition 3 (which used a noisy environment), but still slower than conditions 1 and 2, as shown in figure 9. These learners had a similar performance to those obtained in all the previous conditions, as can be seen in table 6,

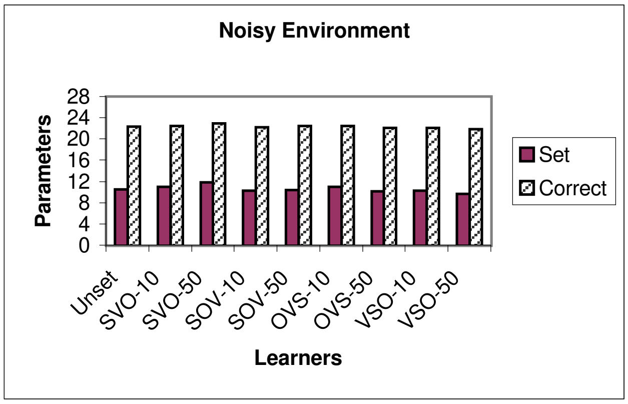

Figure 8: Learners in Noisy Environment

and in figure 10, which also shows the results obtained by the learners in condition 2, which

Table 5: Convergence of the different learners - Learners-50 - Noise

Learners	Triggers	Parameters Correct	Parameters Set
SVO-50	230.3	22.9	11.8
SOV-50	168.1	22.4	10.4
OVS-50	221.4	22.1	10.1
VSO-50	154.6	21.9	9.7

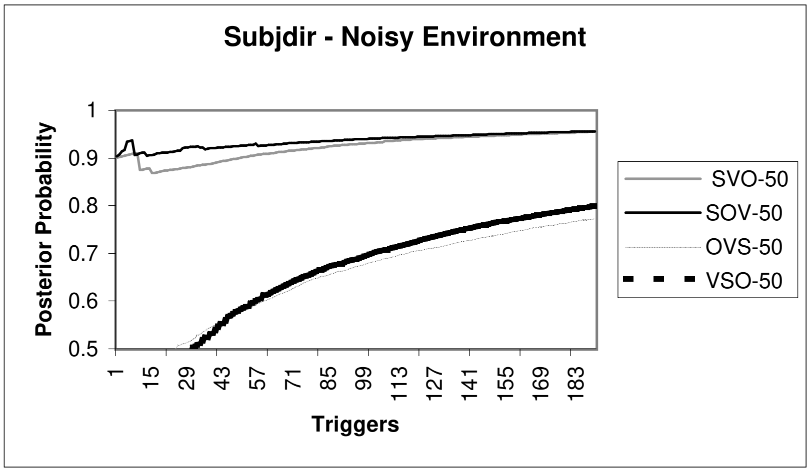

Figure 7: Convergence of Subjdir - Learners-50 - Noisy Environment

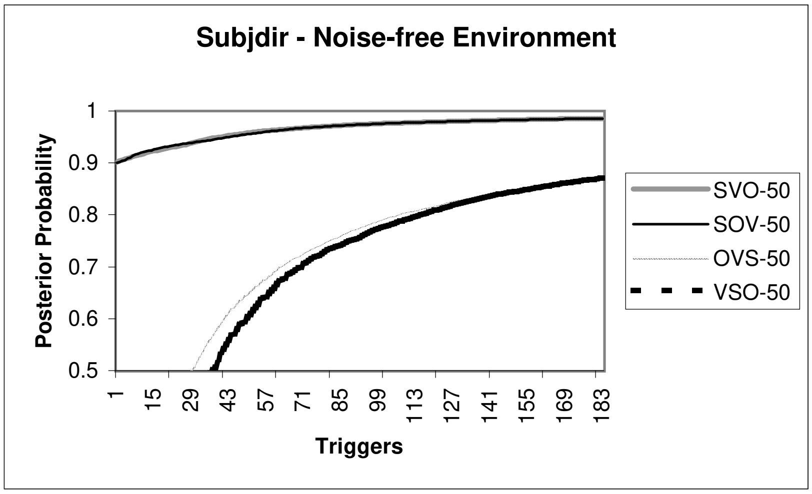

Figure 9: Convergence of Subjdir - Learners-50 - Noise-free Environment

used weaker weights.

Table 6: Convergence of the different learners - Learners-50 - Noise Free

Learners	Triggers	Parameters Correct	Parameters Set
SVO-50	221.7	23.2	11.5
SOV-50	195.4	23.2	11.8
OVS-50	223.2	22.1	9.9
VSO-50	223.4	21.8	9.8

3.5 Discussion

As confirmed by these results, there is a strong interaction between the different starting points and the presence of noise. The noise has a strong influence on the convergence of the learners, slowing down the learning process, since the learners need more triggers to compensate for the effect caused by the noisy ones. The different initialisations caused little impact in the learners' performance, in spite of noticeably delaying the convergence to the target of those learners that have values incompatible with the target. Thus, when combining the presence of noise with the use of stronger weights, there was a significant delay in convergence, where the final posterior probability was up to 10% lower than in the noise-free case (e.g. for the OVS learner), as can be seen in figures 7 and 9. Nonetheless, these learners were robust to the presence of noise in the input data, only selecting or changing a value for a given parameter when there was enough evidence for that. As a consequence, all the learners were converging towards the target, even with the small amount of available triggers, regardless of the initialisations and the presence of noise. This is the case even with an extreme bias in the initial values. Moreover, the learners make effective use of the inheritance mechanism to propagate default values, with an average of around 4.2 non-default specifications for these learners.

4 Conclusion and Future Work

The purpose of this work is to investigate the process of grammatical acquisition from a computational perspective, focusing on the acquisition of word order from data. Five different learners were implemented in this framework and we investigated how the starting point for the learners affects their performance in converging to the target and its interaction with noise. The learners were all converging towards the target grammar, where the different starting points and the presence of noise affected only convergence times, with learners more far away from the target having a slower convergence pattern. Future works include annotating more data to have a bigger corpus, and running more experiments with this corpus, testing how much data is required for all the triggers to converge, with high probability to the target grammar. After that, we will concentrate on investigating the acquisition of subcategorisation frames and argument structure, using the same framework for learning. Although this is primarily a cognitive computational model, it is potentially relevant to the development of more adaptive NLP technology.

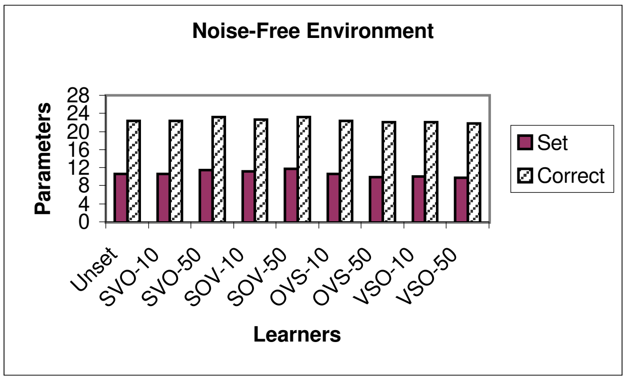

Figure 10: Learners in Noise Free Environment

5 Acknowledgements

I would like to thank Ted Briscoe for his comments and advice on this paper, and Fabio Nemetz for his support. Thanks also to the anonymous reviewers for their comments. The research reported on this paper is supported by doctoral studentship from CAPES/Brazil.

References

Bar Hillel, Y. *Language and Information.* Wesley, Reading, Mass. 1964.

Briscoe, T. *The Acquisition of Grammar in an Evolving Population of Language Agents.* Linkoping Electronic Articles in Computer and Information Science, http://www.ep.liu.se/ea/cis/1999.

Chomsky, N. *Lectures on Government and Binding.* Foris Publications, 1981.

Gold, E.M. *Language Identification in the Limit.* Information and Control, v.10, p.447-474, 1967.

Lascarides, A. and Copestake, A. *Default Representation in Constraint-based Frameworks.* Computational Linguistics, v.25 n.1, p.55-105, 1999.

MacWhinney, B. *The CHILDES Project: Tools for Analyzing Talk.* Second Edition, 1995.

Sachs, J. *Talking about the there and then: the emergence of displaced reference in parent-child discourse.* In K. E. Nelson editor, Children's language, v.4, 1983.

Villavicencio, A. *Representing a System of Lexical Types Using Default Unification.* Proceedings of EACL, 1999.

Villavicencio, A. *The Acquisition of a Unification-Based Generalised Categorial Grammar.* Proceedings of the Third CLUK Colloquium, 2000.

Waldron, B. *Learning Natural Language within the framework of categorial grammar.* Proceedings of the Third CLUK Colloquium, 2000.