The automatic construction of a symbolic parser via statistical techniques

Shyam Kapur
Department of Computer Science
James Cook University
Townsville QLD 4811 (Australia)
kapur@coral.cs.jcu.edu.au

Robin Clark
Department of Linguistics
University of Pennsylvania
Philadelphia PA 19104
rclark@babel.ling.upenn.edu

Abstract

We report on the development of a robust parsing device which aims to provide a partial explanation for child language acquisition and help in the construction of better natural language processing systems. The backbone of the new approach is the synthesis of statistical and symbolic approaches to natural language.

Motivation

We report on the progress we have made towards developing a robust 'self-constructing' parsing device that uses indirect negative evidence (Kapur, 1992) to set its *parameters*. Generally, by parameter, we mean any point of variation at which two languages may differ. Thus, the relative placement of an object with respect to the verb, a determiner with respect to a noun, the difference between prepositional and postpositional languages, and the presence of long distance anaphors like Japanese "zibun" and Icelandic "sig" are all parameters. The device would be exposed to an input text consisting of simple unpreprocessed sentences. On the basis of this text, the device would induce indirect negative evidence in support of some one parsing device located in the parameter space.

The development of a self-constructing parsing system would have a number of practical and theoretical benefits. First, such a parsing device would reduce the development costs of new parsers. At the moment, grammars must be developed by hand, a technique which requires a significant investment in money and man-hours. If a basic parser could be developed automatically, costs would be reduced significantly, even if the parser requires some fine-tuning after the initial automatic learning procedure. Second, a parser capable of self-modification is potentially more robust when confronted with novel or semi-grammatical input. This type of parser would have applications in information retrieval as well as language instruction and grammar correction. Finally, the development of a parser capable of self-modification would give us considerable insight into the formal properties of complex systems as well as the twin problems of language learnability and language acquisition.

Given a linguistic parameter space, the problem of locating a target language somewhere in the space on the basis of a text consisting of only grammatical sentences is far from trivial. Clark (1990, 1992) has shown that the complexity of the problem is potentially exponential because the relationship between the points of variation and the actual data can be quite indirect and tangled. Since, given n parameters, there are 2^n possible parsing devices, enumerative search through the space is clearly impossible. Because each datum may be successfully parsed by a number of different parsing devices within the space and because the surface properties of grammatical strings underdetermine the properties of the parsing device which must be fixed by the learning algorithm, standard deductive machine learning techniques are as complex as a brute enumerative search (Clark, 1992, 1994). In order to solve this problem, robust techniques which can rapidly eliminate inferior hypotheses must be developed.

We propose a learning procedure which unites symbolic computation with statistical tools. Historically, symbolic techniques have proven to be a versatile tool in natural language processing. These techniques have the disadvantage of being both brittle (easily broken by new input or by user error) and costly (as grammars are extended to handle new constructions, development becomes more difficult due to the complexity of rule interactions within the grammar). Statistical techniques have the advantage of robustness, although the resulting grammars may lack the intuitive clarity found in symbolic systems. We propose to fuse the symbolic and the statistical techniques, a development which we view as inevitable; the resulting system will use statistical

learning techniques to output a symbolic parsing device. We view this development to provide a nice middle ground between the problems of overtraining versus undertraining. That is, statistical approaches to learning often tend to overfit the training set of data. Symbolic approaches, on the other hand, tend to behave as though they were undertrained (breaking down on novel input) since the grammar tends to be compact. Combining statistical techniques with symbolic parsing would give the advantage of obtaining relatively compact descriptions (symbolic processing) with robustness (statistical learning) that is not overtuned to the training set.

Preliminaries

Naturally, a necessary preliminary for our work is to specify a set of parameters which will serve as a testing ground for the learning algorithm. This set of parameters must be embedded in a parsing system so that the learning algorithm can be tested against data sets that approximate the kind of input that parsing devices are likely to encounter in real world applications.In this section, we first list some parameters that gives some idea of the kinds of variations between languages that our system is hoped to be capable of handling. We then illustrate why parameter setting is difficult by standard methods. This provides some additional explanation for the failure so far in developing a truly universal parameterized parser.

Linguistic Parameters

Our goal will be to first develop a prototype. We do not require that the prototype accept any arbitrarily selected language nor that the coverage of the prototype parser be complete in any given language. Instead, we will develop a prototype with coverage that extends to some basic structures that any language learning device must account for, plus some structures that have proven difficult for various learning theories. In particular, given an already existing parser, we will extend its coverage by parameterizing it, as described below.

Our initial set of parameters will include the following other points of variation:

1. **Relative order of specifiers and heads**: This parameter covers the placement of determiners relative to nouns, relative position of the subject and the placement of certain VP-modifying adverbs.
2. **Relative order of heads and complements**: This parameter deals with the position of objects relative to the verb (VO or OV orders), placement of nominal and adjectival complements as well as the choice between prepositions and postpositions.
3. **Scrambling**: Some languages allow (relatively) free word order. For example, German has rules for displacing definite NPs and clauses out of their canonical positions. Japanese allows relatively free ordering of NPs and postpositional phrases so long as the verbal complex remains clause final. Other languages allow even freer word orders. We will focus on German and Japanese scrambling, bearing in mind that the model should be extendible to other types of scrambling.
4. **Relative placement of negative markers and verbs**: Languages vary as to where they place negative markers like English *not*. English places its negative marker after the first tensed auxiliary, thus forcing *do* insertion when there is no other auxiliary, while Italian places negation after the tensed verb. French uses discontinuous elements like *ne...pas...* or *ne...plus...* which are wrapped around the tensed verb or which occur as continuous elements in infinitivals. Italian differs from both English and French in placing its negative marker before the first verb, whether tensed or infinitive. The proper treatment of negation will require several parameters, given the range of variation.
5. **Root word order changes**: In general, languages allow for certain word order changes in root clauses but not in embedded clauses. An example of a root word order change is subject-auxiliary inversion in English which occurs in root questions (*Did John leave?* vs. **I wonder did John leave?*). Another example would be inversion of the subject clitic with the tensed verb in French (*Quelle pomme a-t-il mangée* ["which apple did he eat?"]) and process of subject postposition and PP preposition in English (*A man walked into the room* vs. *Into the room walked a man*).
6. **Rightward dislocation**: This includes extraposition structures in English (*That John is late amazes me.* vs. *It amazes me that John is late.*), presentational *there* structures (*A man was in the park.* vs. *There was a man in the park.*), and stylistic inversion in French (*Quelle piste Marie a-t-elle choisie?* ["What path has Marie chosen?"]). Each of these constructions present unique problems so that the entire data set is best handled by a system of interacting parameters.
7. **Wh-movement versus wh-in situ**: Languages vary in the way they encode *wh*-questions. English obligatorily places one and only one *wh*-phrase (for example, *who* or *which picture*) in first position. In French the wh-phrase may remain in place (*in situ*) although it may also form *wh*-questions as in English.

Polish allows *wh*-phrases to be stacked at the beginning of the question.

8. **Exceptional Case Marking, Structural Case Marking**: These parameters have little obvious effect on word order, but involve the treatment of infinitival complements. Thus, exceptional case marking and structural case marking allow for the generation of the order $V_{[+tense]}$ NP $VP_{[-tense]}$, where "$V_{[+tense]}$" is a tensed verb and "$VP_{[-tense]}$" is a VP headed by a verb in the infinitive. Both parameters involve the semantic relations between the NP and the infinitival VP as well as the treatment of case marking. These relations are reflected in constituent structure rather than word order and thus pose an interesting problem for the learning algorithm.

9. **Raising and control**: In the case of raising verbs and control verbs, the learner must correctly categorize verbs which occur in the same syntactic frame into two distinct groups based on semantic relations as reflected in the distribution of elements (for example, idiom chunks) around the verbs.

10. **Long and short distance anaphora**: Short distance anaphors, like "himself" in English must be related to a coreferential NP within a constrained local domain. Long distance anaphors (Japanese "zibun", Korean "caki") must also be related to a coreferential NP, but this NP need not be contained within the same type of local domain as in the short distance case.

The above sampling of parameters has the virtue of being both small (and, therefore, possible to implement relatively quickly) and posing interesting learnability problems which will appropriately test our learning algorithm. Although the above list can be described succinctly, the set of possible targets will be large and a simple enumerative search through the possible targets will not be efficient.

Complexities of Parameter Setting

Theories based on the principles and parameters (*P&P*) paradigm hypothesize that languages share a central core of universal properties and that language variation can be accounted for by appeal to a finite number of points of variation, the so-called parameters. The parameters themselves may take on only a finite number of possible values, prespecified by Universal Grammar. A fully specied *P&P* theory would account for language acquisition by hypothesizing that the learner sets parameters to the appropriate values by monitoring the input stream for "triggering data"; triggers are sentences which cause the learner to set a particular parameter to a particular value. For example, the imperative in (1) is a trigger for the order "V(erb) O(bject)":

(1) Kiss grandma.

under the hypothesis that the learner analyzes *grandma* as the patient of kissing and is predisposed to treat patients as structural objects.

Notice that trigger-based parameter setting presupposes that, for each parameter p and each value v, the learner can identify the appropriate trigger in the input stream. This is the problem of *trigger detection*. That is, given a particular input item, the learner must be able to recognize whether or not it is a trigger and, if so, what parameter and value it is a trigger for. Similarly, the learner must be able to recognize that a particular input datum is *not* a trigger for a certain parameter even though it may share many properties with a trigger. In order to make the discussion more concrete, consider the following example:

(2) a. John$_i$ thinks that Mary likes him$_i$.

b. *John thinks that Mary$_j$ likes her$_j$.

English allows pronouns to be coreferent with a c-commanding nominal just in case that nominal is not contained within the same local syntactic domain as the pronoun; this is a universal property of pronouns and would seem to present little problem to the learner.

Notice, however, that some languages, including Chinese, Icelandic, Japanese and Korean, allow for long distance anaphors. These are elements which are obligatorily coreferent with another nominal in the sentence, but which may be separated from that nominal by several clause boundaries. Thus, the following example from Icelandic is grammatical even though the anaphor *sig* is separated from its antecedent *Jón* by a clause boundary (Anderson, 1986):

(3) Jón$_i$ segir ad María
John says that Mary
elski sig$_i$/hann$_i$
loves self/him
John says that Mary loves him.

Thus, UG includes a parameter which allows some languages to have long distance anaphors and which, perhaps, fixes certain other properties of this class of anaphora.

Notice that the example in (3) is of the same structure as the pronominal example in (2a). A learner whose target is English must not take examples like (2a) as a trigger for the long distance anaphor parameter; what prevents the learner from being deceived? Why doesn't the learner conclude that English *him* is comparable to Icelandic *sig*? We would argue that the learner is sensitive to distributional evidence. For example, the learner is aware of examples like (4):

(4) John$_i$ likes him$_j$.

where the pronoun is not coreferential with anything else in the sentence. The existence of (4) implies that *him* cannot be a pure anaphor, long distance or otherwise. Once the learner is aware of this distributional property of *him*, he or she can correctly rule out (2a) as a potential trigger for the long distance anaphor parameter.

Distributional evidence, then, is crucial for parameter setting; no theory of parameter setting can avoid statistical properties of the input text. How far can we push the statistical component of parameter setting? In this paper, we suggest that statistically-based algorithms can be exploited to set parameters involving phenomena as diverse as word order, particularly verb second constructions, and cliticization, the difference between free pronouns and proclitics. The work reported here can be viewed as providing the basis for a theory of trigger detection; it seeks to establish a theory of the connection between the raw input text and the process of parameter setting.

Parameter Setting Proposal

Let us suppose that there are n binary parameters each of which can take one of two values ('+' or '−') in a particular natural language. The core of a natural language is uniquely defined once all the n parameters have been assigned a value.[1]

Consider a random division of the parameters into some m groups. Let us call these groups $P_1, P_2, \ldots, P_m$. The Parameter Setting Machine first goes about setting all the parameters within the first group P_1 concurrently as sketched below. After these parameters have been fixed, the machine next tries to set the parameters in group P_2 in a similar fashion, and so on.

[1]Parameters can be looked at as fixed points of variation among languages. From a computational point of view, two different values of a parameter may simply correspond to two different bits of code in the parser. We are not committed to any particular scheme for the translation from a tuple of parameter values to the corresponding language. However, the sorts of parameters we consider have been listed in the previous section.

1. All parameters are unset initially, i.e., there are no preset values. The parser is organized to only obey all the universal principles. At this stage, utterances from any possible natural language are accommodated with equal ease, but no sophisticated structure can be built.
2. Both the values of each of the parameters $p_i \in P_1$ are 'competing' to establish themselves.
3. Corresponding to p_i, a pair of hypotheses are generated, say H^i_+ and H^i_-.
4. Next, these hypotheses are tested on the basis of input evidence.
5. If H^i_- fails or H^i_+ succeeds, set p_i's value to '+'. Otherwise, set p_i's value to '−'.

Formal Analysis of the Parameter Setting Machine

We next consider a particular instantiation of the hypotheses and their testing. The way we have in mind involves constructing suitable window-sizes during which the algorithm is sensitive to occurrence as well as non-occurrence of specific phenomena. Regular failure of a particular phenomenon to occur in a suitable window is one natural, robust kind of indirect negative evidence.

For example, the pair of hypotheses may be

1. Hypothesis H^i_+: Expect not to observe phenomena from a fixed set O^i_- of phenomena which support the parameter value '−'.
2. Hypothesis H^i_-: Expect not to observe phenomena from a fixed set O^i_+ of phenomena which support the parameter value '+'.

Let w_i and k_i be two small numbers. Testing the hypothesis H^i_+ involves the following procedure:

1. A window of size w_i sentences is constructed and a record is maintained whether or not a phenomenon from within the set O^i_- occurred among those w_i sentences.
2. This construction of the window is repeated k_i different times and a tally c_i is made of the fraction of times the phenomena occurred at least once in the duration of the window.
3. The hypothesis H_+ succeeds if and only if the ratio of c_i to k_i is less than 0.5.

Note that the phenomena under scrutiny are assumed to be such that the parser is always capable of analyzing (to whatever extent necessary) the input. This is because in our view the parser consists of a fixed, core program whose behavior can be modified by selecting from among a finite set of 'flags' (the parameters). Therefore, even if not all of the flags have been set to the correct values, the parser is such that it can at least partially represent the input. Thus, the parser is

always capable of analyzing the input. Also, there is no need to explicitly store any input evidence. Suitable window-sizes can be constructed during which the algorithm is sensitive to occurrence as well as non-occurrence of specific phenomena. By using windows, just the relevant bit of information from the input is extracted and maintained. (For detailed argumentation that this is a reasonable theoretical argument, see Kapur (1992, 1993).) Notice also that we have only sketched and analyzed a particular, simple version of our algorithm. In general, a whole range of window-sizes may be used and this may be governed by the degree to which the different hypotheses have earned corroboration. (For some ideas along this direction in a more general setting, see Kapur (1991, 1992).)

Order in which parameters get set

Notice that in our approach certain parameters get set quicker than others. These are the ones that are expressed very frequently. It is possible that these parameters also make the information extraction more efficient quicker, for example, by enabling structure building so that other parameters can be set. If our proposal is right, then, for example, the word order parameters which are presumably the very first ones to be set must be set based on a very primitive parser capable of handling any natural language. At this early stage, it may be that word and utterance boundaries cannot be reliably recognized and the lexicon is quite rudimentary. Furthermore, the only accessible property in the input stream may be the linear word order. Another particular difficulty with setting word-order parameters is that the surface order of constituents in the input does not necessarily reflect the underlying word-order. For example, even though Dutch and German are SOV languages, there is a preponderance of SVO forms in the input due to the V2 (verb-second) phenomenon. The finite verb in root clauses moves to the second position and then the first position can be occupied by the subject, objects (direct or indirect), adverbials or prepositional phrases. As we shall see, it is important to note that if the subject is not in the first position in a V2 language, it is most likely in the first position to the right of the verb. Finally, it has been shown by Gibson and Wexler (1992) that the parameter space created by the head-direction parameters along with the V2 parameter has *local maxima*, that is, incorrect parameter settings from which the learner can never escape.

Computational Analysis of the Parameter Setting Machine

V2 parameter In this section, we summarize results we have obtained which show that word order parameters can plausibly be set in our model.[2] The key concept we use is that of *entropy*, an information-theoretic statistical measure of randomness of a random variable. The entropy $H(X)$ of a random variable X, measured in bits, is $-\sum_X p(x)\log p(x)$. To give a concrete example, the outcome of a fair coin has an entropy of $-(.5 * \log(.5) + .5 * \log(.5)) = 1$ bit. If the coin is not fair and has .9 chance of heads and .1 chance of tails, then the entropy is around .5 bits. There is less uncertainty with the unfair coin—it is most likely going to turn up heads. Entropy can also be thought of as the number of bits on the average required to describe a random variable. Entropy of one variable, say X, conditioned on another, say Y, denoted as $H(X|Y)$ is a measure of how much better the first variable can be predicted when the value of the other variable is known.

Descriptively, verb second (V2) languages place the tensed verb in a position that immediately follows the first constituent of the sentence. For example, German is V2 in root clauses, as shown in (refex:v2-root), but not in embedded clauses, as shown in (refex:embedding):[3]

(5) a. Hans *hat* Maria
H. has M.
getroffen.
met
"Hans has met Maria."

b. Hans *wird* Maria
H. will M.
getroffen haben.
met has
"Hans will have met Maria."

(6) a. weil Hans Maria
because H. M.
getroffen. hat.
met has
"Hans has met Maria."

b. weil Hans Maria
because H. M.
getroffen haben wird.
met has will
"because Hans will have met Maria."

In the examples in (5), a constituent, XP, has

[2] Preliminary results obtained with Eric Brill were presented at the 1993 Georgetown Roundtable on Language and Linguistics: Pre-session on Corpus-based Linguistics.

[3] See the papers collected in Haider & Prinzhorn (1985) for a general discussion of V2 constructions.

been moved into the Specifier position of CP, triggering movement of the finite verb to C^0. This results in the structure shown in (7). Notice that the constituent XP can be of any category, may be extracted from an embedded clause ormay be an adverbial; thus, the XP need not be related to the finite verb via selectional restrictions or subcategorization:

(7) $[_{CP}\ XP_i\ [_{C^0}\ V_j] \ldots t_i \ldots t_j]$
where V_j is a finite verb.

The V2 parameter (or set of parameters) would regulate the movement of a constituent to the Specifier of CP, forcing movement of the finite verb to C^0 as well as determining whether the V2 structures are restricted to the root clause or may occur in embedded clauses.

We considered the possibility that by investigating the behavior of the entropy of positions in the neighborhood of verbs in a language, word order characteristics of that language may be discovered.[4] For a V2 language, we expect that there will be more entropy to the left of the verb than to its right, i.e., the position to the left will be less predictable than the one to the right. This is because the first position need not be related to the verb in any systematic way while the position following the verb will be drawn from a more restricted class of elements (it will either be the subject or an element internal to the VP); hence, there is more uncertainty (higher entropy) about the first position than about the position following the verb. We first show that using a simple distributional analysis technique based on the five verbs the algorithm is assumed to know, another fifteen words most of which turn out to be verbs can readily be obtained.

Consider text as generating tuples of the form (v, d, w), where v is one of the top twenty words (most of which are verbs), d is either the position to the left of the verb or to the right, and w is the word at that position.[5] V, D and W are the corresponding random variables.

The procedure for setting the V2 parameter is the following:

If $H(W|V, D = \mathit{left}(\mathrm{L})) > H(W|V, D = \mathit{right}(\mathrm{R}))$ **then** +V2 **else** -V2.

Language	$H(W\|V, D = \mathrm{L})$	$H(W\|V, D = \mathrm{R})$
English	4.22	4.26
French	3.91	5.09
Italian	4.91	5.33
Polish	4.09	5.78
Tamil	4.01	5.04
Turkish	3.69	4.91
Dutch	4.84	3.61
Danish	4.42	4.24
German	5.55	4.97

Table 1. Entropy in the Neighborhood of Verbs

On each of the 9 languages on which it has been possible to test our algorithm, the correct result was obtained. (Only the last three languages in the table are V2 languages.) Furthermore, in almost all cases, it was also shown to be statistically significant. The amount (only 3000 utterances) and the quality of the input (unstructured unannotated input caretaker speech subcorpus from the CHILDES database (MacWhinney, 1991)), and the computational resources needed for parameter setting to succeed are psychologically plausible. Further tests were successfully conducted in order to establish both the robustness and the simplicity of this learning algorithm. It is also clear that once the value of the V2 parameter has been correctly set, the input is far more revealing with regard to other word order parameters and they too can be set using similar techniques.

In order to make clear how this procedure fits into our general parameter setting proposal, we spell out what the hypotheses are. In the case of the V2 parameter, the two hypotheses are not separately necessary since one hypothesis is the exact complement of the other. So the hypothesis H_+ may be as shown.

Hypothesis H_+: Expect not to observe that the entropy to the left of the verbs is lower than that to the right.

The window size that may be used could be around 300 utterances and the number of repetitions need to be around 10. Our previous results provide empirical support that this should suffice.

By assuming that besides knowing a few verbs, as before, the algorithm also recognizes some of the first and second person pronouns of the language, we can not only determine aspects of the

[4] In the competition model for language acquisition (MacWhinney, 1987), the child considers cues to determine properties of the language but while these cues are reinforced in a statistical sense, the cues themselves are not information-theoretic in the way that ours are. In some recent discussion of triggering, Niyogi and Berwick (1993) formalize parameter setting as a Markov process. Crucially, there again the statistical assumption on the input is merely used to ensure that convergence is likely, and triggers are simple sentences.

[5] We thank Steve Abney for suggesting this formulation to us.

pronoun system (see below) but also get information about the V2 parameter. The first step of learning is same as above; that is, the learner acquires additional verbs based on distributional analysis. We expect that in the V2 languages (Dutch and German), the pronouns will appear more often immediately to the right of the verb than to the left. For French, English and Italian exactly the reverse is predicted. Our results (2 to 1 or better ratio in the predicted direction) confirm these predictions.[6]

Clitic pronouns We now show that our techniques can lead to straightforward identification and classification of clitic pronouns.[7] Briefly, clitic pronouns are phonologically reduced elements which obligatorily attach to another element. Syntactic clitics have a number of syntactic consequences including special word order properties and an inability to participate in conjunctions and disjunctions. For example, in French,, full direct objects occur after the lexical verb but accusative clitics appear before the verb:

(8) a. Jean a vu les filles.
J. has seen the girls
"Jean saw the girls."

b. Jean les a vues.
J. clitic has seen
"Jean saw them."

Restricting our attention, for the moment to French, we should note that clitic pronouns may occur in sequences, in which case there are a number of restrictions on their relative order. Thus, nominative clitics (eg., "je", "tu", "il", etc.) occur first, followed by the negative element "ne", followed by accusative clitics (eg., "la", "me", "te") and dative clitics ("lui"), followed, at last, by the first element of the verbal sequence (an auxiliary or the main verb). There are further ordering constraints within the accusative and dative clitcs based on the person of the clitic; see Perlmutter (1971) for an exhaustive description of clitic pronouns in French.

In order to correctly set the parameters governing the syntax of pronominals, the learner must distinguish clitic pronouns from free and weak pronouns as well as sort all pronoun systems according to their proper case system (e.g., nominative pronouns, accusative pronouns). Furthermore, the learner must have some reliable method for identifying the presence of clitic pronouns in the input stream. The above considerations suggest that free pronouns occur in a wider range of syntactic environments than clitic pronouns and, so, should carry less information about the syntactic nature of the positions that surround them. Clitic pronouns, on the other hand, occur in a limited number of environments and, hence, carry more information about the surrounding positions. Furthermore, since there are systematic constraints on the relative ordering of clitics, we would expect them to fall into distribution classes depending on the information they carry about the positions that surround them. The algorithm we report, which is also based on the observation of entropies of positions in the neighborhood of pronouns, not only distinguishes accurately between clitic and free-standing pronouns, but also successfully sorts clitic pronouns into linguistically natural classes.

It is assumed that the learner knows a set of first and second person pronouns. The learning algorithm computes the entropy profile for three positions to the left and right of the pronouns ($H(W|P = p)$ for the six different positions), where ps are the individual pronouns. These profiles are then compared and those pronouns which have similar profiles are clustered together. Interestingly, it turns out that the clusters are syntactically appropriate categories.

In French, for example, based on the Pearson correlation coefficients we could deduce that the object clitics "me" and "te", the subject clitics "je" and "tu", the non-clitics "moi" and "toi", and the ambiguous pronouns "nous" and "vous" are most closely related only to the other element in their own class.

Table 2. Correlation Matrix for the French Pronouns

VOUS	1							
TOI	0.62	1						
MOI	0.57	0.98	1					
ME	0.86	0.24	0.17	1				
JE	0.28	0.89	0.88	-0.02	1			
TU	0.41	0.94	0.94	0.09	0.97	1		
TE	0.88	0.39	0.30	0.95	0.16	0.24	1	
NOUS	0.91	0.73	0.68	0.82	0.53	0.64	0.87	1
	VOUS	TOI	MOI	ME	JE	TU	TE	NOUS

In fact, the entropy signature for the ambiguous pronouns can be analyzed as a mathematical combination of the signatures for the conflated forms. To distinguish clitics from non-clitics, we use the measure of *stickiness* (proportion of times they are sticking to the verbs compared to the times

[6] We also verified that the object clitics in French were not primarily responsible for the correct result.

[7] Preliminary results were presented at the Berne workshop on L1- and L2-acquisition of clause-internal rules: scrambling and cliticization in January, 1994.

they are two or three positions away). These results are quite good. The stickiness is as high as 54-55% for the subject clitics; non-clitics have stickiness no more than 17%.

The Dutch clitic system is far more complicated than the French pronoun system. (See for example, Zwart (1993).) Even so, our entropy calculations made some headway towards classifying the pronouns. We are able to distinguish the weak and strong subject pronouns. Since even the strong subject pronouns in Dutch tend to stick to their verbs very closely and two clitics can come next to each other, the raw stickiness measure seems to be inappropriate. Although the Dutch case is problematic due to the effects of V2 and scrambling, we are in the process of treating these phenomena and anticipate that the pronoun calculations in Dutch will sort out properly once the influence of these other word order processes are factored in appropriately.

Conclusions

It needs to be emphasized that in our statistical procedure there is a mechanism available to the learning mechanism by which it can determine when it has seen enough input to reliably determine the value of a certain parameter. (Such means are non-existent in any trigger-based error-driven learning theory.) In principle at least, the learning mechanism can determine the variance in the quantity of interest as a function of the text size and then know when enough text has been seen to be sure that a certain parameter has to be set in a particular way.

We are currently extending the results we have obtained to other parameters and other languages. We are convinced that the word order parameters (for example, those in (1-2) in the section Preliminaries) should be fairly easy to set and amenable to an information-theoretic analysis along the lines sketched earlier. Scrambling also provides a case where calculations of entropy should provide an immediate solution to the parameter-setting problem. Notice however that both scrambling and V2 interact in an interesting way with the basic word order parameters; a learner may be potentially misled by both scrambling and V2 into mis-setting the basic word order parameters since both parameters can alter the relationship between heads, their complements and their specifiers.

Parameters involving adverb placement, extraposition and wh-movement should be relatively more challenging to the learning algorithm given the relatively low frequency with which adverbs are found in adult speech to children. These cases provide good examples which motivate the use of multiple trials by the learner. The interaction between adverb placement and head movement, then, will pose an interesting problem for the learner since the two parameters are interdependent; what the learner assumes about adverb placement is contingent on what it assumes about head placement and vice versa.

References

Anderson, S. 1986. The typology of anaphoric dependencies: Icelandic (and other) reflexives in L. Hellan & K. Christensen (eds) *Topics in Scandinavian Syntax.* D. Reidel Publishing Company, Dordrecht, the Netherlands, pp. 65-88.

Robin Clark. 1990. Papers on learnability and natural selection. *Technical Report 1,* Université de Genève, Département de Linguistique générale et de linguistique française, Faculté des Lettres, CH-1211, Genève 4, 1990. Technical Reports in Formal and Computational Linguistics.

Robin Clark. 1992. The selection of syntactic knowledge. *Language Acquisition,* 2(2):83–149.

Robin Clark. 1994. Hypothesis formation as adaptation to an environment: Learnability and natural selection. In Barbara Lust, Magui Suñer, and Gabriella Hermon, editors, *Syntactic Theory and First Language Acquisition: Crosslinguistic Perspectives.* Lawrence Erlbaum Assoc.. Presented at the 1992 symposium on 'Syntactic Theory and First Language Acquisition: Cross Linguistic Perspectives' at Cornell University.

Huber Haider and Martin Prinzhorn (eds). 1985. *Verb Second Phenomena in Germanic Languages.* Foris Publications, Dordrecht, the Netherlands.

Edward Gibson and Kenneth Wexler. 1992. Triggers. Presented at GLOW.

Shyam Kapur. 1991. *Computational Learning of Languages.* PhD thesis, Cornell University. Computer Science Department Technical Report 91-1234.

Shyam Kapur. 1993. How much of what? Is this what underlies parameter setting? In *Proceedings of the 25th Stanford University Child Language Research Forum.* Also in Cognition. (To appear.)

Shyam Kapur. 1994. Some applications of formal learning theory results to natural language acquisition. In Barbara Lust, Magui Suñer, and Gabriella Hermon, editors, *Syntactic Theory and First Language Acquisition: Crosslinguistic Perspectives.* Lawrence Erlbaum Assoc.. Presented at the 1992 symposium on 'Syntactic Theory and First Language Acquisition: Cross Linguistic Perspectives' at Cornell University.

Shyam Kapur and Gianfranco Bilardi. 1992. Language learning from stochastic input. In *Proceedings of the fifth conference on Computational Learning Theory.* Morgan-Kaufman.

Brian MacWhinney. 1987. The competition model. In Brian MacWhinney, editor, *Mechanisms of Language Acquisition.* Lawrence Erlbaum Assoc..

Brian MacWhinney. 1991. *The CHILDES Project: Tools for analyzing Talk.* L. Erlbaum Assoc., Hillsdale, New Jersey.

Partha Niyogi and Robert C. Berwick. 1993. Formalizing triggers: A learning model for finite spaces. Technical Report A.I. Memo No. 1449, Massachusetts Institute of Technology. Also Center for Biological Computational Learning, Whitaker College Paper No. 86.

David Perlmutter. (1971). *Deep and Surface Constraints in Syntax.* Holt, Reinhart and Winston, New York.

C. Jan-Wouter Zwart. 1993. Notes on clitics in dutch. In Lars Hellan, editor, *Clitics in Germanic and Slavic,* pages 119–155. Eurotyp working papers, Theme Group 8, Vol. 4, University of Tilburg.