Behavior Research Methods, Instruments, & Computers
2004, 36 (1), 113-126

Automatic parsing of parental verbal input

KENJI SAGAE, BRIAN MACWHINNEY, and ALON LAVIE
Carnegie Mellon University, Pittsburgh, Pennsylvania

To evaluate theoretical proposals regarding the course of child language acquisition, researchers often need to rely on the processing of large numbers of syntacticallyparsed utterances, both from children and from their parents. Because it is so difficult to do this by hand, there are currently no parsed corpora of child language input data. To automate this process, we developed a system that combined the MOR tagger, a rule-based parser, and statistical disambiguation techniques. The resultant system obtained nearly 80% correct parses for the sentences spoken to children. To achieve this level, we had to construct a particular processing sequence that minimizes problems caused by the coverage/ambiguity tradeoff in parser design. These procedures are particularly appropriate for use with the CHILDES database, an international corpus of transcripts. The data and programs are now freely available over the Internet.

Explaining the enigma of child language acquisition is one of the core challenges facing cognitive science. Although all normal children succeed in learning their native tongue, neither psychology nor linguistics has yet succeeded in accounting for the many complexities of language learning. Within this general area, there has been particular attention to the acquisition of grammar, stimulated in large measure by Chomsky's (1982) theory of universal grammar and its attendant claims regarding innate principles and parameters.

To investigate these proposals, researchers have come to rely increasingly on large corpora of transcript data of verbal interactions between children and parents. The standard database in this area is the CHILDES database (MacWhinney, 2000; http://childes.psy.cmu.edu), which provides a large amount of transcript data for over 25 human languages. There are now several hundred studies in which the CHILDES database has been used to study the development of morphosyntax. However, most of these studies have been forced to use the database in its raw lexical form, without tags for parts of speech and without syntactic parses. Lacking this information, researchers have devoted long hours of hand analysis to locate and code the sentences relevant to their hypotheses. If tags and parses had been available, these analyses could have been automated, allowing the investigators to conduct a wider variety of tests in a more reliable fashion.

As an initial move in this direction, a morphological tagger called MOR (MacWhinney, 2000) has been developed for English, French, German, Italian, Spanish, Japanese, and Cantonese. The results of the MOR tagger can be disambiguated using the POST program (Parisse & Le Normand, 2000). The level of accuracy of this combination of the MOR and POST programs has now reached levels as high as 95%, which is close to the current state of the art for automatic tagging (Garside & Smith, 1997). In the present work, we seek to build on these advances in order to add a deeper layer of syntactic information to the utterances in the CHILDES corpora. This additional structure contains information on the constituent structure found in these utterances and the grammatical functions of these constituents.

The idea of annotating natural language corpora with syntactic structures is not a new one. Over the past decade, a number of annotation efforts have resulted in large amounts of text annotated with syntactic parse trees. These collections are known as *treebanks* (Marcus, Santorini, & Marcinkiewics, 1993). However, none of the treebanks currently in existence specifically addresses the needs of child language acquisition research. With a few exceptions, the language found in these treebanks has often been taken from written material, such as newspaper texts, and the annotation style has been designed to facilitate the training of statistical language analysis tools, rather than the study of language acquisition.

It would require months of work by a trained linguist to create even a relatively small annotated corpus (15,000 sentences). Recent advances in natural language processing have created the possibility of performing an automatic (or semiautomatic) syntactic analysis of natural language with a high degree of accuracy. This analysis is commonly referred to as *automatic syntactic parsing*, or simply *parsing*. In this article, we describe our efforts in using state-of-the-art natural language analysis technolo-

This work was supported by a grant from the National Science Foundation Linguistic Program to B. M. Thanks to Christophe Parisse for creating the POST program and to Yuriko Oshima-Takane and her students at McGill for producing these disambiguated tags for the training of POST. Maria Lamendola assisted with hand-checking of the results of LCFlex. Correspondence concerning this article should be addressed to K. Sagae, Language Technologies Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213 (e-mail: sagae@cs.cmu.edu).

gies to parse the parental input language used in one of the corpora from the CHILDES database. The output of the parsing process is an annotated relational structure suitable for use in the study of the acquisition of syntax.

Parsing

Parsers use a computational model of natural language to analyze a sentence. They produce a syntactic structure of the sentence as output. The syntactic structure may take several forms, such as a constituent tree (c-structure, or parse tree), a syntactic feature structure (or f-structure), or a dependency structure (Figure 1). These various representations of a sentence may differ in the level of information they contain (parts of speech, case, syntactic function labels, etc.), but each of them describes in some way how words combine to form a sentence. The choice of a particular representational format and, therefore, the choice of a syntactic parser depend mainly on the purpose the syntactic analyses will serve.

Natural language processing researchers have made use of a number of existing models and theories of language to develop systems that perform syntactic parsing with increasingly high levels of accuracy. Rule-based parsers rely on a relatively small set of grammatical rules that implement specific linguistic theories and principles of syntax (Hauser, 1999). Statistical parsers rely on regularities in the data and the training corpus to extract parts of speech and co-occurrence patterns (Charniak, 1997). Whichever design is chosen, no parser is able to achieve completely accurate results. There are at least four reasons for these problems. First, natural language is highly ambiguous. As native speakers with good intuitions, we often fail to sense the scope of this ambiguity. However, when we come to articulating a system for automatic parsing, the funda-

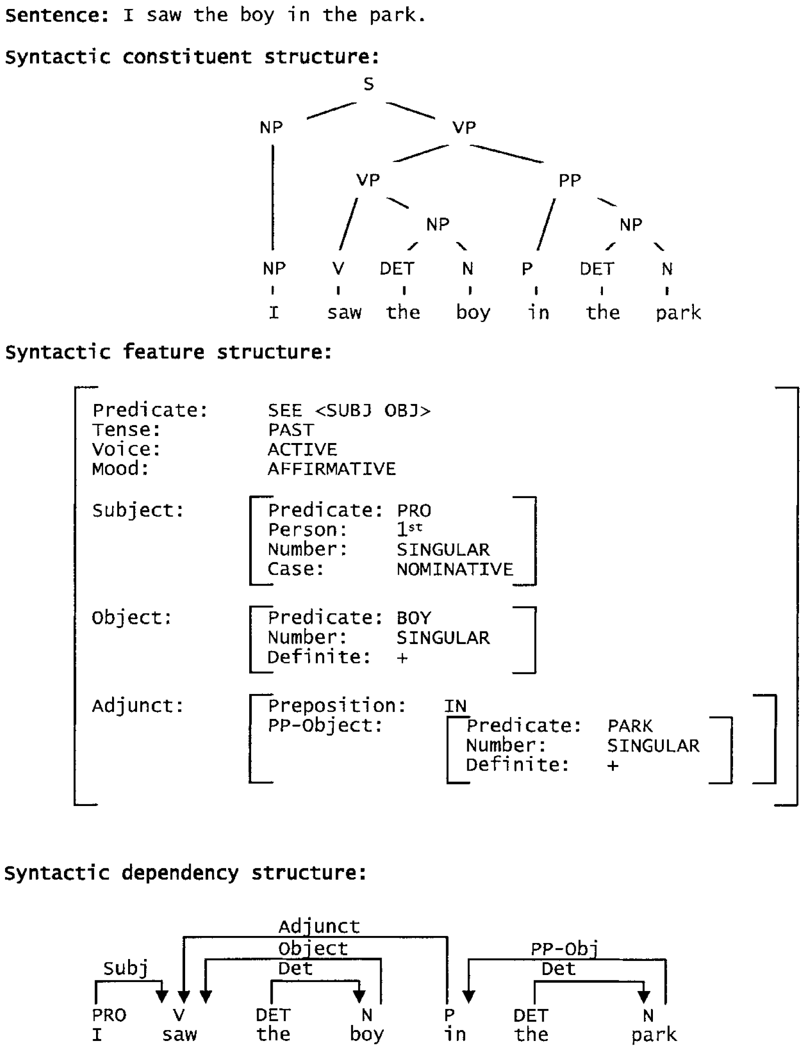


Figure 1. Syntactic constituent structure, feature structure, and dependency structure representations of a sentence.

mental ambiguity of attachment, roles, interpretations, and relations in human language stands as a formidable challenge. Second, much of the information needed to determine the correct parse lies outside the scope of the sentence, in the domain of the discourse or the situational context. Third, most parsers are trained with material from a specific genre or area of language use. When the parser is then extended to cover material from a new domain, often there is a serious decrement in performance. Finally, there are some problems that are unique to the task of parsing spoken language. Specifically, spoken language differs from written language by including large numbers of dysfluencies, filled pauses, retracings, and other conversational features.

In addition to the inherent ambiguity commonly expected in natural languages, parsers must consider a remarkably large number of alternative parses for all but the simplest sentences. Allowing for the analysis of a large number of syntactic constructions leads to the development of a model that suffers from more ambiguity than does a more restrictive model. The balance of coverage and ambiguity is a crucial issue in parsing. As an example, consider the simple context-free grammar and sentences:

S → NP VP
NP → DET N
NP → PRO
VP → V NP
VP → VP PP
PP → P NP
V → saw

(7) DET → the
(8) N → boy
(9) N → dog
(10) N → telescope
(11) P → with
(12) PRO → I

(S1) I saw the boy with the dog.

(S2) I saw the dog with the telescope.

Despite the similar part-of-speech sequences in Sentences S1 and S2, their syntactic structures differ in the place where the prepositional phrases (PP) "with the dog" and "with the telescope" should be attached. The correct analysis of S1 has the prepositional phrase attached to the noun phrase (Figure 2A), yielding the paraphrase "I saw the boy who was with a dog." In other words, the phrase "with the dog" modifies the phrase "the boy." However, the grammar above lacks the rule used to make the prepositional phrase attachment in Figure 2A, and the only analysis it allows for S1 is the incorrect analysis seen in Figure 2B (which might be paraphrased as "*I used the dog to see the boy," where the prepositional phrase attaches to the verb, having the phrase "with the dog" modify the verb "saw"). Conversely, the correct analysis of S2 has the prepositional phrase "with the telescope" attached to the verb phrase (Figure 2C), thus modifying the verb "saw" and yielding the paraphrase "I used the telescope to see the dog." This is the only analysis of S2 allowed by our grammar. The incorrect attachment of "with the telescope" to the noun phrase "the dog" (which could be paraphrased as "*I saw the dog that had a telescope"; Figure 2D) is not allowed. In summary, the grammar above allows for exactly one syntactic analysis of each sentence, but in the case of S1, the analysis is incorrect.

For the grammar to cover the correct analysis of S1, we need to add an additional rule that allows prepositional phrases to modify noun phrases:

(2′) NP → NP PP.

However, the addition of a new rule to handle a previously uncovered syntactic structure may have adverse side effects in the overall performance of the grammar. Even though the addition of Rule 29 to the grammar allows for the correct analysis of S1 (Figure 2A), the incorrect analysis (Figure 2B) is still possible. What is even worse is that this modification allows for the incorrect analysis of S2 (Figure 2D), which could be analyzed correctly and unambiguously only before the addition of Rule 29. Although there are ways to resolve the ambiguities in S1 and S2 and produce the correct analysis in each case (e.g., by using additional knowledge sources, more complex grammar constructions, feature unification, or statistical disambiguation models), this example illustrates how increasing the coverage of a grammar may result in unwanted ambiguity.

The CHILDES Database and the Eve Corpus

Among the corpora in the CHILDES database, we chose to focus on the Eve corpus (Brown, 1973). Our choice was motivated by the fact that we had already created a clean transcription with manually verified part-of-speech tags for the child utterances, as well as its central role in child language acquisition research (Moerk, 1983). The corpus includes utterances from the child (Eve), as well as from her parents. An example of child utterances in the corpus can be seen in Figure 3. In this example, the first line is a transcription of one of Eve's utterances (indicated by *CHI :), and the following line contains part-of-speech and morphological annotations for that utterance (indicated by %mor:). Adult sentences in the corpus include a line with part-of-speech information, but that information is produced fully automatically and is often ambiguous. An example can be seen in Figure 4.

Although the adult language in the CHILDES corpora generally conforms to standard spoken language, the child language in the corpora varies from the language of a child in the very early stages of language learning to fairly complex syntactic constructions. We believe that the child and the adult utterances differ significantly enough that we may be able to analyze them more accurately by doing so separately, possibly with different strategies. In this article, we explore the "easier" (in the sense that it is better defined) problem of analyzing the adult utterances in the Eve corpus, whose role in child language acquisition has been the subject of extensive research (Moerk, 1983). Although parsing of the adult input is easier than parsing of the child's forms, it is theoretically of equal importance, since theories of learning depend heavily on consideration of the range of constructions provided to children in the input (MacWhinney, 1999).

In our work, we used rule-based parsing techniques to analyze each adult utterance in the corpus, to produce syn-

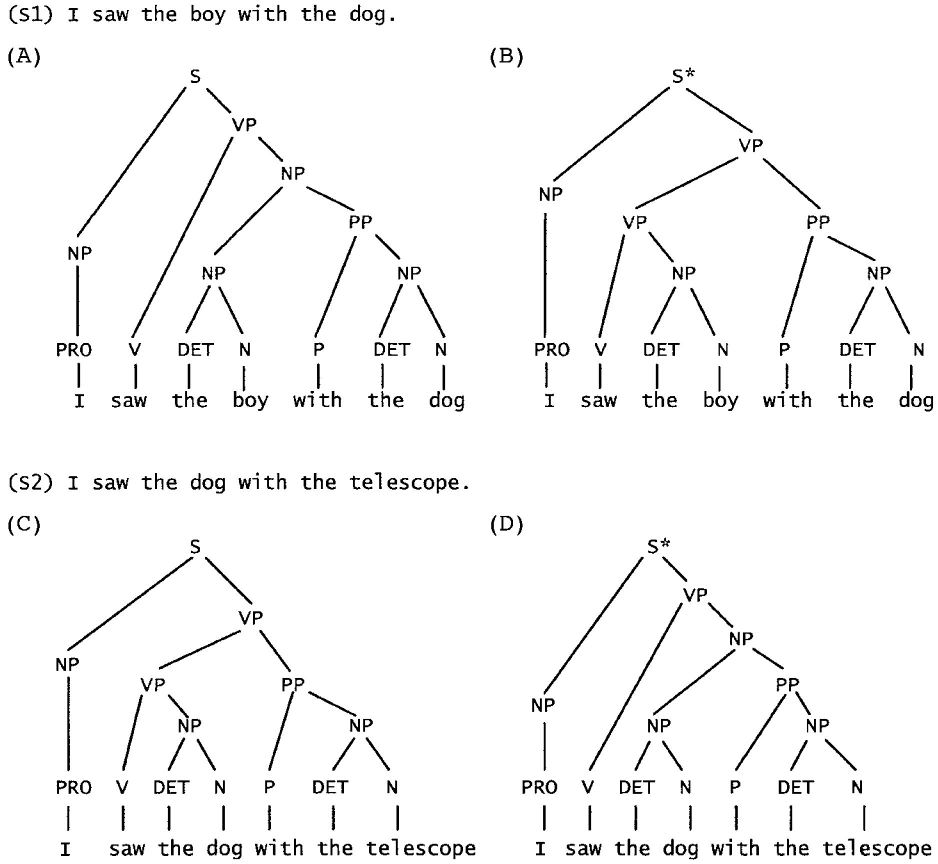

(S1) I saw the boy with the dog.
(A)
S
VP
NP
PP
NP
NP
NP
PRO V DET N P DET N
I saw the boy with the dog
(B)
S*
VP
NP
VP
PP
NP
NP
PRO V DET N P DET N
I saw the boy with the dog
(S2) I saw the dog with the telescope.
(C)
S
VP
NP
VP
PP
NP
NP
PRO V DET N P DET N
I saw the dog with the telescope
(D)
S*
VP
NP
PP
NP
NP
NP
PRO V DET N P DET N
I saw the dog with the telescope

Figure 2. Parse trees for "I saw the boy with the dog" and "I saw the dog with the telescope" (trees rooted with S* describe an incorrect analysis). The trees in panels B and C can be generated with the given grammar, but the desired trees are those in panels A and C. Adding a rule to allow for the analysis in panel A has the undesired side effect of also allowing the (incorrect) tree in panel D.

tactic annotations in the form of syntactic feature structures (marked by %syn:), as is illustrated in Figure 5. Figure 6 shows a graphical representation of the annotations in Figure 5. The syntactic feature structures we produced resemble the feature–value pairs in the functional structures of lexical functional grammar (LFG; Bresnan, 2001), although we made no attempt to follow LFG theory closely. In our f-structures, the features are typically syntactic functions, and the values are syntactic constituents or syntactic characteristics of the sentence. The index feature provides a cross-reference between a feature structure and its corresponding constituent structure.

Once a corpus has been annotated with automatically generated syntactic feature structures, a corpus browser allows a user to view a graphical representation of the syntactic analysis for a sentence and to label it as correct or incorrect (Figure 7). Although the annotation process still relies on human expertise at this step, the time required to judge the correctness of a feature structure is only a small fraction of the time it would take a person to generate the analysis. When the analysis of a sentence is found to be incorrect, the user may enter a comment to accompany the analysis.

The Syntactic Analysis System

To produce the syntactic analyses necessary for annotation of the corpus, we developed a syntactic analysis system based on grammar-driven robust parsing and statistical disambiguation. Grammar-driven (or rule-based) parsers use a set of production rules that specify how each syntactic constituent may be expanded into other constituents or words as a model of natural language. The type of grammar used by our system follows a formalism based on a context-free backbone augmented with feature unification constraints. As an example, Figure 8 shows a simple grammar that can be used with our system.

The grammar in Figure 8 can be used to parse the sentence "He sees the ball" as follows. A lexical analysis of the words in the input determines the part-of-speech and agreement features (in the cases of "he" and "sees") of each word. Rule 3 allows the formation of a noun phrase

```
*CHI:   more cookie.

%mor:   qn|more n|cookie .
```

Figure 3. Sample child utterance from the Eve corpus.

```
*MOT:    how about another graham
         cracker ?

%mor:    adv:wh|how prep|about^adv|about
         det|another n|graham n|cracker ?
```

Figure 4. Sample adult utterance from the Eve corpus.

(NP) from the pronoun (PRO) "he." The single unification equation associated with that rule states that every feature in the first element of the right-hand side of the context-free portion of the rule (represented in the equation by x1 and corresponding to the pronoun) will be passed to the newly formed constituent, or the left-hand side of the context-free rule (the noun phrase, represented in the equation by x0). Rule 2 forms another noun phrase from the words "the" and "ball." The first equation in the rule specifies that the first element in the right-hand side of the context-free rule (DET, represented in the equation by x1) is the value of a feature named determiner of the second element of the right-hand side. The second equation specifies that every feature of the second element of the right-hand side (including the newly specified determiner feature) be passed to the newly created constituent (NP). Rule 4 can then be applied to form a verb phrase (VP) from the verb "sees" and the noun phrase "the ball." According to the first equation of Rule 4, the noun phrase becomes the object of the verb. Finally, the noun phrase "he" and the verb phrase "sees the ball" can be combined by Rule 1 to produce a sentence (S) constituent that spans the entire input string, completing the parse. The first equation of Rule 1 requires that the value of the agreement features of the verb phrase and the noun phrase match. These values are provided by the lexical analysis performed before the parsing process. The second equation makes the noun phrase the subject of the sentence. The final analysis can be seen in Figure 9.

Robust parsing technologies seek to augment the coverage of a parser by allowing it to analyze language phenomena that fall outside of the coverage of the parser's model (in our case, a syntactic grammar). Our use of robustness is targeted toward the analysis of unforeseen spoken language phenomena. This is achieved by allowing the parser to (1) insert lexical or nonterminal items (constituents) that are not present in the input string and (2) skip certain words in the input string. The specific uses of these techniques are discussed in the Parser Flexibility section below. Although the expansion of coverage provided by robust parsing increases the ambiguity problem faced by the analysis system, we employ statistical techniques to allow the parser to cope with such ambiguity. By providing a training corpus of manually disambiguated sentences (of much smaller size than the total amount of text ultimately analyzed), we can build a statistical model of grammar usage to make certain decisions in the parsing process that result in fairly accurate disambiguation.

The input to our system is a sequence of transcribed utterances, and the output is a syntactic analysis for each of those utterances, as can be seen in the example in Figure 5. At a lower level, the system can be divided into three main components (Figure 10): (1) MOR, a part-of-speech tagger, developed especially for CHILDES corpora, and a statistical disambiguator of these tags called POST (Parisse & Le Normand, 2000); (2) LCFlex (Rosé & Lavie, 2001), a robust parser that provides special features for parsing spoken language; and (3) a statistical disambiguation module to pick the correct analysis from the many produced by the parser. One of the goals of our work is to reduce the overall need for manual work to the point at which reliable annotations can be generated for 80% of a 15,000-sentence corpus in just a few days, as opposed to the months of work that a trained linguist would require to produce the annotations from scratch. However, it will still be necessary for a linguist to check over the results of the automatic parsing. This check involves only a binary decision. In 80% of the cases, the linguist simply has to accept the results of the parser. In the remaining 20%, further processing will be needed. In the following sections, we will take a closer look at the three main components of the analysis system.

MOR and POST. POST (Parisse & Le Normand, 2000) operates on the tags inserted by the MOR program to pro-

```
*MOT:   you kicked it .

%mor:   pro|you v|kick-PAST pro|it .

%syn:   ((mood *declarative) (tense *past)
         (index 2)
         (subject ((cat pro) (num *sg) (pers 2)
                   (case *nom)(index 1) (root *you)))
         (object ((cat pro) (sum sg) (pers 3)
                  (case acc) (index 3) (root *it)))
         (root *kick) (cat v))

%cst:   (sentence (decl (np (pro you))
                        (vp (vbar (v kicked)
                                  (np (pro it)))))
                  (period .))
```

Figure 5. Sample syntactic annotations in the Eve corpus.

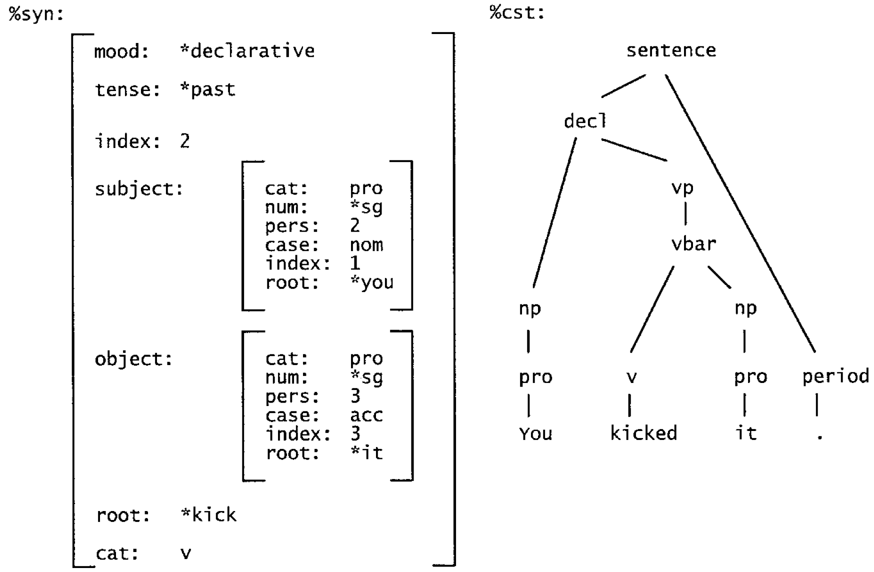


Figure 6. Graphical representation of the %syn and %cst lines in Figure 5.

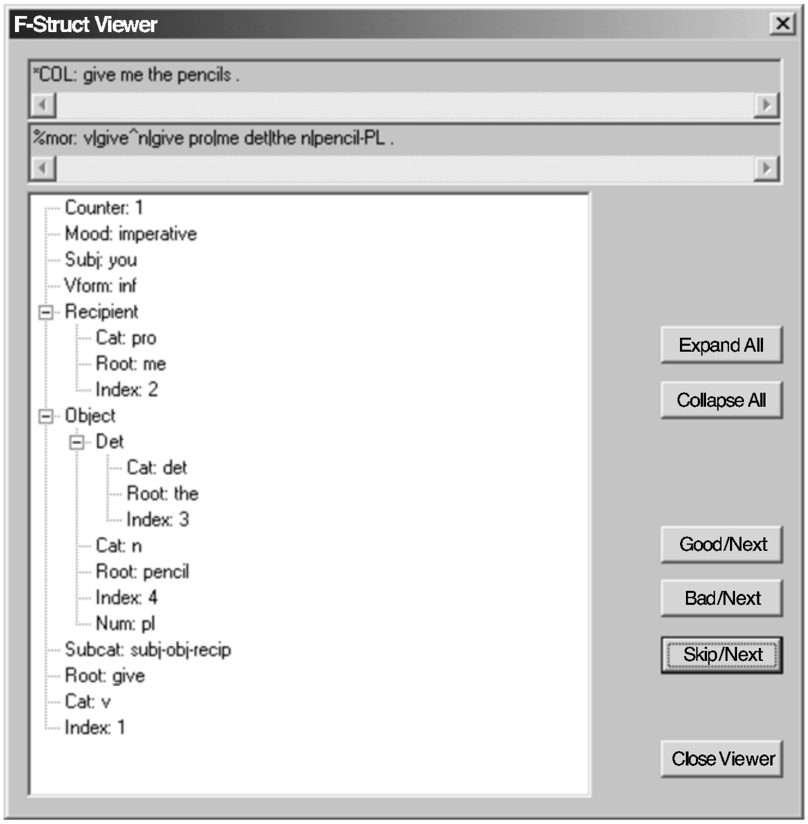
vide an unambiguous morphosyntactic analysis at the lexical (word) level with high accuracy. MOR uses a lexicon and morphological rules to provide all the possible parts of speech and morphological analyses for each word in transcribed utterances in the CHILDES corpora. POST chooses one single interpretation for each word from the options provided by MOR. Automatic tagging and disambiguation of thousands of words can be done in less than 1 min.

To make disambiguation decisions, POST uses a set of rules that must be obtained from a training corpus, which consists of several sentences with unambiguous and correct part-of-speech tags for each word. These rules basically describe which pairs of part-of-speech tags have been seen in the training corpus following one another (and how frequently). During disambiguation, POST uses this information to determine the most likely sequence of part-of-speech tags for a sentence. For example, in the sentences "I saw a can" and "I can fly," the word "can" is used as a noun and as an auxiliary. POST decides which is the correct interpretation in each case by considering what parts of speech typically precede or follow nouns and auxiliaries (according to the training corpus). In the case of "I saw a can," the determiner–noun sequence is much more likely for "a can" than the determiner–auxiliary sequence. In the case of "I can fly," the sequences auxiliary–verb and noun–verb (for "can fly") are both likely. However, the pronoun–auxiliary sequence is more likely than the pronoun–noun sequence (for "I can"). By considering the frequencies of all consecutive pairs of tags in a sentence, POST determines the most likely tag for each word. Although this example provides an idea of how lexical disambiguation is accomplished, it offers only a much simplified view of how POST works. For a complete description of POST, consult Parisse and Le Normand (2000).

LCFlex. Because the corpora in the CHILDES database consist only of transcribed spontaneous speech (with its dysfluencies and other spontaneous conversational features), having a parser designed to handle such language is of great importance. Through a set of parameters, LCFlex can be tuned to allow the insertion of specific missing syntactic constituents into a sentence and to skip extra-grammatical material that would prevent an analysis from being found with the grammar in use. These features allow great flexibility in parsing spoken language, but their parameters must be tuned carefully to balance benefits and the increased ambiguity caused by allowing insertions and skipping.

LCFlex is an agenda-driven bottom-up chart parser. A detailed description of how such parsers work is provided in Allen (1995). In a nutshell, bottom-up parsers start from words (or part-of-speech tags) to form the smallest constituents (such as noun phrases) first, placing them in a chart (in the case of a chart parser). In an agenda-driven parser, newly formed constituents are inserted into an agenda. At each iteration of the parser, a constituent is taken from the agenda, and the parser consults its grammar rules to find out how that constituent may be combined with other constituents in the chart to form new larger constituents, which will be then added to the agenda. Two constituents can be combined into a new constituent only if they are adjacent. Once the possibilities for a given constituent are exhausted, the parser inserts it into the chart. Parsing finishes when the agenda is empty. At that point, we check the chart for constituents of certain types (usually "sentence," or S) that span the entire input sentence.

Grammars used by LCFlex are of the type discussed earlier in this section (context-free backbone with unification equations) and illustrated in Figure 8. These grammars (including the one used in our system) can be edited man-



```
*COL: give me the pencils .
%mor: v|give^n|give pro|me det|the n|pencil-PL .
```

```
Counter: 1
Mood: imperative
Subj: you
Vform: inf
Recipient
    Cat: pro
    Root: me
    Index: 2
Object
    Det
        Cat: det
        Root: the
        Index: 3
    Cat: n
    Root: pencil
    Index: 4
    Num: pl
Subcat: subj-obj-recip
Root: give
Cat: v
Index: 1
```

Figure 7. Feature structure viewer. The user may rate the analysis as correct or incorrect simply by using the appropriate buttons.

ually in any text-editing program. The output of LCFlex is also in text form (Figure 5) and can also be edited manually for error correction.

LCFlex's capability for inserting missing syntactic constituents in the analysis of a given sentence is due to a modification in the general bottom-up chart-parsing algorithm. When the parser considers how a constituent (picked from the agenda) may combine with other existing constituents, it also considers the combination of the constituent with the specific constituents we allow the parser to insert. These inserted constituents do not correspond to any actual words in the sentence. For example, if the sentence "went home" is given as input to the parser and if the insertion of a noun phrase is allowed, the parser may find an analysis that includes a noun phrase with no lexical content as the subject of the input sentence. A different modification allows word skipping. When the parser considers how a constituent may combine with other constituents, it may also consider combinations of constituents that are one word apart (if we allow skipping of a single word), in which case the word between the two constituents is ignored.

```
(1) S -> NP VP
      (x1 AGREEMENT) =c (x2 AGREEMENT)
      (x2 SUBJECT) = x1
      x0 = x2

(2) NP -> DET N
      (x2 DETERMINER) = x1
      x0 = x2

(3) NP -> PRO
      x0 = x1

(4) VP -> V NP
     (x1 OBJECT) = x2
      x0 = x1
```

Figure 8. A simple grammar composed of a context-free backbone and unification equations.

A detailed description of LCFlex and the modifications to the bottom-up chart-parsing algorithm that allows for limited insertions and word skipping can be found in Rosé and Lavie (2001). LCFlex is implemented in Common Lisp, and it processes about 300 sentences from the CHILDES corpora per minute on a 600-MHz Pentium III with 128 Mb of RAM.

Statistical disambiguation. The idea behind statistical syntactic disambiguation in LCFlex is that each analysis of a particular utterance is obtained through an ordered succession of grammar rule applications and the correct analysis should be the one resulting from the most probable succession of rules. The probability of each competing analysis is determined on the basis of a statistical

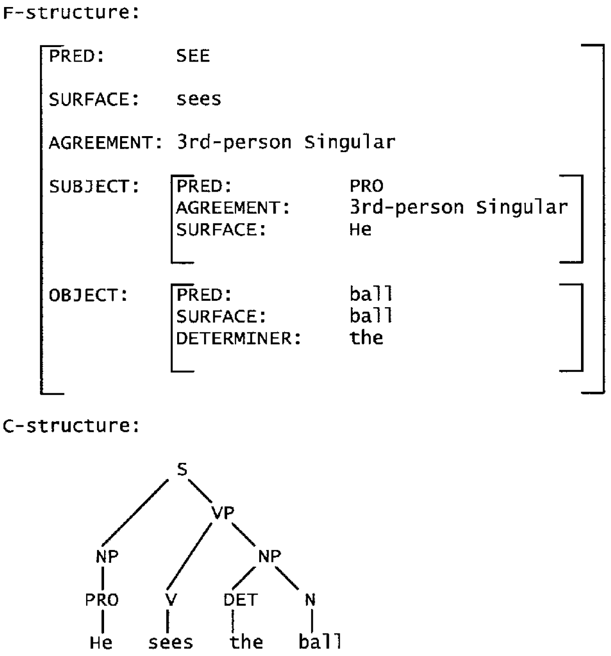


Figure 9. Analysis of the sentence "He sees the ball" according to the grammar in Figure 8. The *pred*, *agreement*, and *surface* features are created during lexical analysis.

model of bigrams of rule applications (Rosé & Lavie, 2001) obtained from training examples.

The training corpus required by the statistical disambiguation model consists of sentences paired with their correct analyses. The parser uses the training corpus to count the usage of bigrams of grammar rules in the process of parsing a sentence into its correct analysis. This corresponds to a *two-level* probabilistic context-free grammar model. In a standard probabilistic context-free grammar, each rule in the grammar is associated with a probability. A training corpus containing correct parses can be used to count the usage of each rule and determine their frequencies. In the two-level case, or with bigrams of rule applications, instead of estimating the probability of each rule in isolation, we estimate the probability of each rule, given the previous rule. For example, instead of having a rule VP → V NP with probability of .8, we might determine from training data that the probability of rule VP → V NP is .6 if its parent rule is S → NP VP, and .2 if its parent rule is VP → VP PP. This allows for rule probabilities to be sensitive to context. Probabilistic grammars of this type are sometimes referred to as *pseudocontext sensitive*. Details on the statistical disambiguation model used by LCFlex can be found in Rosé and Lavie (2001).

Tailoring a High-Performance Analysis System

To obtain high-quality syntactic analyses from a system composed of the pieces described in the previous section, each component must be tuned carefully, keeping in mind the behavior of the overall system. This section will focus on the specific issues that relate to the components of the parser, as well as their integration into a high-performance analysis system.

Grammar. The grammar needed by LCFlex consists of context-free rules augmented with feature unification constraints. Although general purpose English grammars are available, we found that they were not suitable for our analysis task. The main problems associated with "off-the-shelf" grammars are related to the large amount of ambiguity allowed by such grammars (a practically unavoidable consequence of a grammar designed to analyze unconstrained English sentences) and the lack of support for certain phenomena we find in the corpora in the

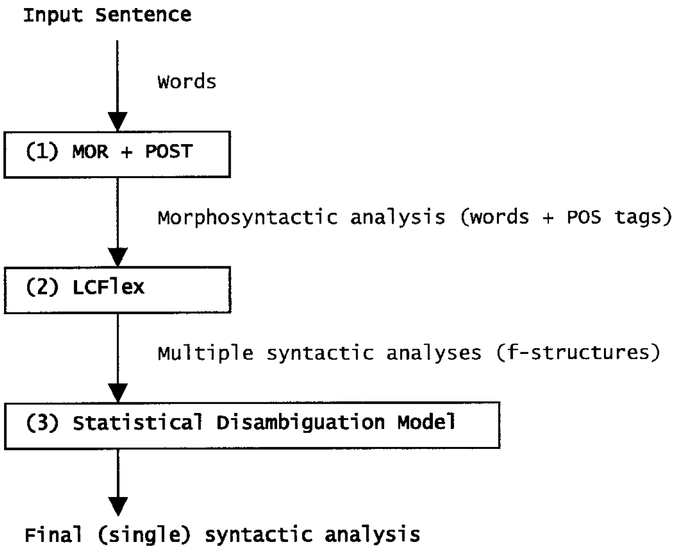

Input Sentence
Words
(1) MOR + POST
Morphosyntactic analysis (words + POS tags)
(2) LCFlex
Multiple syntactic analyses (f-structures)
(3) Statistical Disambiguation Model
Final (single) syntactic analysis

Figure 10. Components of the analysis system.

CHILDES database (such as the extensive use of communicators and vocatives commonly used in casual spoken language, onomatopoeia, etc.). It has been reported that practical grammars used to analyze newspaper articles written in English produce an astronomical number of parses per sentence (Moore, 2000), with the vast majority of these parses being completely uninterpretable from a human point of view. As a simple example of this phenomenon, Charniak (1997) used the sentence "Salespeople sold the dog biscuits" and a grammar not unlike the one discussed earlier in relation to Figure 2, but including the rule NP → NP NP.

Although this rule may seem unusual, it is used to analyze such phrases as "10 dollars a share," where both "10 dollars" and "a share" are noun phrases that combine to form a larger noun phrase. Charniak (1997) gave three analyses for his example sentence (Figure 11). Whereas the first two analyses can be easily interpreted as "dog biscuits were sold by the salespeople" and "biscuits were sold to the dog by the salespeople," respectively, the third analysis does not seem to have a meaningful interpretation. In fact, it was the result of the application of a rule designed to cover a syntactic construction not present in any plausible interpretation of this sentence. The interested reader is encouraged to see Charniak for a more detailed account of the ambiguity problem in practical natural language grammars.

Because of the nature and the domain of our target corpus, it does not contain many of the complex syntactic constructions found in newspaper-style text or even in adult conversations. We can take advantage of that fact and attempt to reduce ambiguity by using a grammar that fits our target language more tightly. It is a fairly accepted notion in natural language processing that parsing within a specific domain can be made more accurately with the use of domain-specific resources.

Starting with a general purpose English grammar with about 600 rules, we pruned or simplified a large number of rules that would never (or rarely) be used in correct analyses for the target corpus. For example, the noun phrase rule mentioned above can be safely discarded, since we are not interested in covering constructions such as "10 dollars a share." The result was a completely rewritten compact grammar with 152 rules. This final grammar included rules to handle the specific language phenomena likely to appear in the CHILDES database and represents a much cleaner and tighter model of the language in the domain we are attempting to analyze. As a result, the potential for ambiguity in parsing was significantly reduced.

Lexical ambiguity. Even though a more suitable grammar is a first step toward managing ambiguity, it is not a complete solution to the problem, and further techniques to resolve syntactic ambiguity are needed. One such way is to eliminate lexical ambiguity by selecting a single part-of-speech tag for each word, using the part-of-speech tagger. The first step is to have a corpus of correctly tagged text to train the tagger. Unfortunately, the CHILDES database contains no unambiguous part-of-speech tagged data for adult utterances. Although tagged data for child utterances are available, the child and the adult languages are

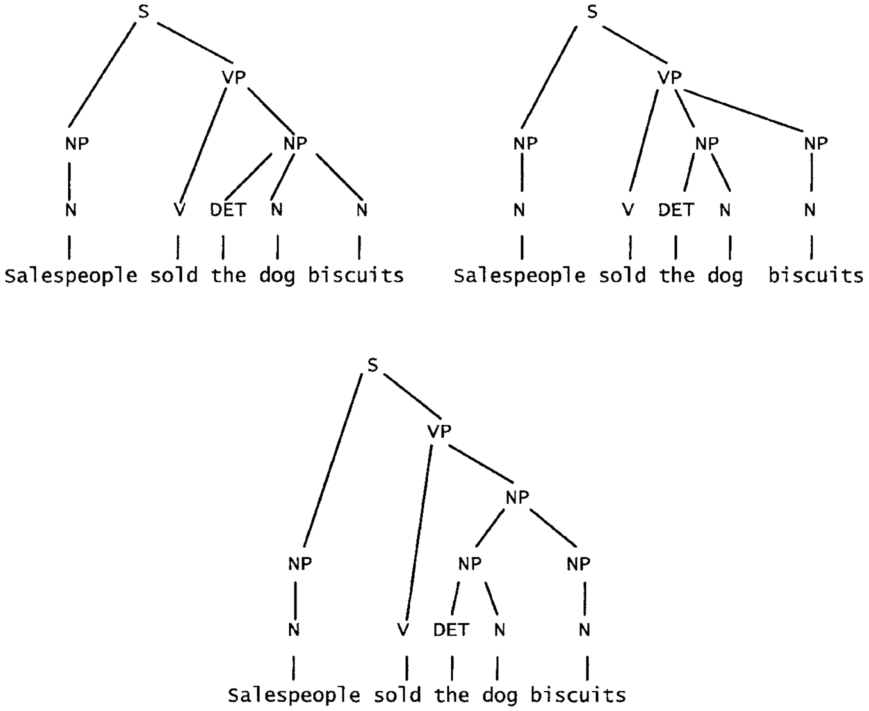

S
VP
NP
NP
N
V
DET
N
N
Salespeople sold the dog biscuits
S
VP
NP
NP
NP
N
V
DET
N
N
Salespeople sold the dog biscuits
S
VP
NP
NP
NP
NP
N
V
DET
N
N
Salespeople sold the dog biscuits

Figure 11. Three syntactic analyses for the sentence "Salespeople sold the dog biscuits."

significantly different so that a tagger trained on child utterances would perform poorly in tagging adult ones. To create a part-of-speech tagging training corpus for the adult language in the corpus, we used the following bootstrapping process: (1) use tagged child utterances to train a part-of-speech tagger for adult utterances; (2) tag adult utterances (4,000 words) and hand correct them; and (3) retrain the tagger with the newly corrected data and iterate from Step 2. By performing four iterations of the procedure above, we improved the accuracy of the part-of-speech tagger from an initial 87.2% to 94.3%. The improvement in accuracy for each iteration decreased at a rapid pace, and it is unlikely that further iterations would yield significant benefits (at least not cost effectively).

Syntactic ambiguity. Once syntactic ambiguity has been reduced through the elimination of lexical ambiguity, we can attempt to find the single correct analysis produced by the parser (when one exists), using statistical disambiguation. For that, we need a training corpus of correct sentence analysis pairs. We create this data in a way similar to the bootstrapping process used to generate part-of-speech training data, but this time we start with the results of parsing lexically unambiguous input: (1) parse a set of utterances tagged with parts of speech; (2) examine the analyses that are unambiguous (or nearly unambiguous); (3) add correct analyses to the training corpus; (4) train the statistical disambiguation module with the training corpus; (5) use the parser with statistical disambiguation on the utterances that were not previously added to the training corpus, obtaining at most a single analysis per utterance; (6) examine the resulting analyses manually and add the correct ones to the training corpus; and (7) iterate from Step 4. We started with an initial training corpus of fewer than 500 sentences and increased its size to 3,000 sentences in four iterations of the process described above, although the benefits of successive iterations decreased at a fast rate. Lexical disambiguation led to unambiguous parses for only a few sentences. However, these sentences can be very useful in obtaining an initial training corpus for the statistical disambiguation module, since resolving large amounts of syntactic ambiguity manually may be a practically intractable task.

Once we obtain an initial training corpus for statistical disambiguation, we can increase its size while also increasing the size of our part-of-speech tagging training corpus by using a feedback loop between part-of-speech tagging and parsing. We assume that the input sentences for which the parser produces correct analyses have correct part-of-speech tag assignments, and we add those sentences to our part-of-speech training corpus. Improvements in part-of-speech tagging, in turn, result in more correct analyses being produced by the parser.

Parser flexibility. Even after grammar development, a large number of sentences in the Eve corpus still could not be parsed with our compact grammar, due to specific characteristics of the casual conversational language in the corpus (and not to general syntactic structures). The majority of such sentences were not covered successfully because of omitted words or filled pauses in otherwise fully grammatical utterances—for example, (1) missing auxiliary verbs in questions ("[*Do*] You want to go outside?"); (2) missing noun phrases, as elided subjects ("[*I*] Don't think she wants to play now.") and even as elided objects ("Give [*it*] to me."); (3) missing auxiliary verbs and noun phrases ("[*Do*] [*you*] Want to go outside?"); and (4) filled pause ("I'd like to tell you, *uh*, something."). Adding explicit ad hoc grammar rules to handle such sentences would cause the grammar to deviate from the clean model of language we were hoping to achieve and would add much harmful ambiguity to the analysis process. Instead, we turned to the robustness features of the parser to handle these sentences. LCFlex allows the addition of specific syntactic nodes to an analysis, or skipping of words in a sentence, making it conform to the grammar and leading to a successful analysis.

Balancing coverage and ambiguity. We will now examine the effects of each of the strategies above on the coverage/ambiguity tradeoff. First, we will consider the methods that we used to decrease ambiguity in the parses. Ambiguity is a serious problem, since parsing with the initial general English grammar and without proper training of the disambiguation module yielded few unambiguous parses. We considered an analysis "correct" only if it contained no errors or ambiguity. Using the initial general grammar coupled with the statistical disambiguation provided by LCFlex, we obtained less than 50% accuracy in analyzing the Eve corpus (measured with a 200-utterance test corpus). We define accuracy as the ratio between correct analyses and the total number of sentences analyzed. Although incorrect analyses often contained correctly analyzed portions that could be considered useful information, our evaluation methodology considers "correct" only an analysis that contains no errors. Using our final rewritten grammar and statistical disambiguation (trained on 3,000 correctly parsed utterances), we reached close to 65% accuracy. This reflects improvements in grammar coverage and better ambiguity resolution resulting from the use of a smaller task-specific grammar.

The use of part-of-speech tagging and morphological analysis to eliminate lexical ambiguity improved syntactic ambiguity resolution even further. However, the accuracy of the part-of-speech tagger is just below 95%, so we can expect an incorrect tag in about every 20 words. This means that in every 4 or 5 sentences, 1 is likely to contain an error in automatic part-of-speech assignment. Such errors in a sentence typically make it impossible for the parser to find a correct syntactic analysis and often prevent the parser from finding an analysis at all. When the final grammar with part-of-speech tagged input sentences was used to eliminate lexical ambiguity, the number of correct analyses in the 200-sentence test corpus decreased to 57.5%. In terms of the ambiguity/coverage tradeoff, we decreased ambiguity significantly, but at the cost of a severe reduction in coverage. We achieved a 1.1% improvement in part-of-speech tagging by using transformation-based learning of Brill-style rules (Brill, 1995), which resulted in a slight improvement in coverage. However, overall parser accuracy obtained with part-of-speech tagged input was still under 60%.

Next, let us examine the methods that we used to increase the coverage of the parser. Setting the parser to allow limited insertions (a single noun phrase and/or a single auxiliary may be inserted during parsing) led to an improvement in recognition for about 5% of the sentences in the Eve corpus. However, the percentage of improvement in accuracy was less than 3%, due to the increased ambiguity and over-generation that resulted from increasing the search space of possible analyses with insertions. Allowing limited skipping (a single word in the input utterance may be skipped during parsing) actually decreased the overall accuracy. In other words, the number of sentences that were parsed incorrectly due to the increased search space was greater than the number of correct analyses that resulted from limited skipping.

Each of the strategies that we used to reduce ambiguity or increase coverage has a different impact on the coverage/ambiguity tradeoff, and the effect of applying them together by naively combining them all at once is far from optimal. Our efforts to reduce ambiguity come at the cost of reducing coverage, and our efforts to increase coverage result in increased ambiguity. By applying lexical disambiguation, limited insertions, and skipping and relying only on the statistical model of bigrams of rule applications for parse selection, we achieve less than 70% parsing accuracy with the Eve corpus.

We must then attempt to balance the coverage/ambiguity tradeoff to benefit from both decreased ambiguity and increased coverage. We do so by controlling the amount of ambiguity and coverage in several passes of parsing. We start with the most restrictive settings and the least ambiguity, and upon failures in parsing, gradually increase coverage (and ambiguity). The idea is that we pay the cost of an increased search space only as it becomes necessary, taking advantage of both more limited ambiguity, when possible, and increased coverage, when needed. Through empirical observation, we arrived at the settings shown in Table 1 for each of the passes. In this way, we reached 78.5% correct parses—the highest level we were able to obtain. In the first pass, we parse lexically unambiguous input and use no coverage-enhancing techniques. From passes two through six, we allow limited lexical ambiguity and gradually increase coverage through the robust parsing features of LCFlex. Limited lexical ambiguity means that not every possible part-of-speech tag (according to a lexicon available with the CHILDES database) is allowed for each lexical item, which would cause a greater increase in syntactic ambiguity. Instead, we allow lexical ambiguity only for certain lexical categories where the automatic part-of-speech tagger was observed to make frequent mistakes, causing parser failures. We determined those highly confusable parts of speech simply by analyzing the cause of failed analyses and keeping track of the parts of speech most frequently associated with those failures. The following sets of tags accounted for more than 95% of the failures caused by a part-of-speech tagging error: {verb, auxiliary}, {verb particle, preposition}, {adverb, adjective}, and {noun, verb}.

The reason for not combining multiple coverage-increasing techniques in further passes of parsing is that we prefer having no analysis for an utterance to having an analysis that is very likely to be incorrect. This multipass approach not only increases ambiguity gradually only as needed, but also allows us to have some sense of how confident we are that an analysis is correct. Figures 12 and 13 illustrate how our final multipass analysis system works. Note that in a situation where "parser failure" (denoted by parse failed in Passes 1 and 2 in Figure 13) occurs, no analyses are returned for a given utterance. It is the absence of an analysis at a parsing pass that automatically triggers the next pass.

Results

To assess the effectiveness of our methods, we evaluated our current system on 200 randomly chosen previously unused utterances from the Eve corpus and checked their generated syntactic analyses for correctness. The contribution of each of the six passes to the total number of correct analyses can be seen in the Table 2. The overall level of correct parses obtained here is 78.5%. The causes of the remaining errors in incorrect analyses are shown in Table 3. The row labeled "insertion" refers to the utterances that were not covered by the grammar but were assigned an incorrect analysis due to limited insertions. The row labeled "over-generation" refers to utterances for which the parser did not produce an appropriate analysis, due to lack of grammar coverage, but where the utterance was still covered in an incorrect way, due to grammar over-generation. Finally, the causes of parsing failures where no analysis was produced for an utterance is shown in Table 4.

We are unaware of any other efforts to produce syntactic analyses for the CHILDES corpora or similar data. Because of the level of specialization of our system, direct quantitative comparisons with other systems are not possible. With that in mind, we will provide a general idea of what levels of performance are usually expected in the analysis of spoken language by briefly mentioning a few other spoken language parsers. The statistical parsers of Charniak and Johnson (2001) and Roark (2001) produce constituent structures (a shallower form of analysis than

Table 1
Coverage and Ambiguity Settings for Different Passes of Parsing

Pass	POS Ambiguity	Insertion	Skipping
1	None	None	None
2	Limited	None	None
3	Limited	Auxiliary	None
4	Limited	NP	None
5	Limited	Auxiliary and NP	None
6	Limited	None	One word

Note—POS, part of speech.

```
Input:
      we'll play with Sandy later .

Ambiguous morphosyntactic analysis (ambiguous POS):
      pro|we~v:aux|will v|play^n|play prep|with n:prop|Sandy
      adv|later^adj|late-CP .

Disambiguated morphosyntactic analysis (disambiguated POS):
      pro|we~v:aux|will v|play prep|with n:prop|Sandy adv|later .

Pass 1 (using disambiguated POS):
      ((COUNTER 1)
       (MOOD *DECLARATIVE)
       (AUX ((MODAL +) (ROOT *WILL) (CAT V-AUX) (INDEX 2)))
       (SUBJ ((CAT PRO) (ROOT *WE) (INDEX 1)))
       (ADJUNCT (*MULTIPLE* ((CAT ADV) (ROOT *LATER) (INDEX 6))
                            ((PP-OBJ ((CAT N-PROP) (ROOT *SANDY) (INDEX 5)))
                                     (CAT PREP) (ROOT *WITH) (INDEX 4))))
       (SUBCAT *SUBJ) (CAT V) (VFORM INF) (ROOT *PLAY) (INDEX 3))
```

Figure 12. The input sentence is correctly analyzed in the first pass, and no further passes are performed.

the feature structures obtained with our system) for transcriptions of the Switchboard corpus (adult–adult telephone conversations). Each of the two parsers obtains a complete match of parser output to gold-standard analyses on fewer than 60% of the sentences. However, longer sentences and the larger vocabulary size in their target corpus make a complete match more difficult. On the other hand, such parsers are trained on tens of thousands of hand-made syntactic analyses and are extremely robust. Even though the rate of completely correct analyses may seem low, these statistical parsers have virtually no coverage problems, and most of the constituents with a sentence (about 84% of all constituents) are typically recognized correctly. The search-augmented neural network parser of Buø and Waibel (1996) produces a level of syntactic analysis that is closer to what our system produces. On an evaluation of their parser on task-oriented, limited-domain, transcribed spoken language, they achieve 71.8% accuracy. GLR* (Lavie, 1996), a rule-based robust parser (on which the development of LCFlex was based) achieves 60%–70% accuracy on different evaluations involving transcribed spoken language. Once again, these numbers cannot be compared directly with our accuracy figures, since each of the evaluations for the different parsers differed significantly—from the test data and grammars used to the level of syntactic analysis and strictness of the accuracy measures. A comparison with parsers designed to analyze written text (such as newspaper articles) is even

```
Input:
      you want a cookie ?

Ambiguous morphosyntactic analysis (ambiguous POS):
      pro|you v|want det|a n|cookie ?

Disambiguated morphosyntactic analysis (disambiguated POS):
      pro|you v|want det|a n|cookie ?

Pass 1 (using disambiguated POS):
      (Parse failed)

Pass 2 (using ambiguous POS):
      (Parse failed)

Pass 3 (allowing insertion of auxiliary):
      ((COUNTER 1)
       (MOOD *INTERROGATIVE)
       (AUX ((DUMMY +)))
       (SUBJ ((CAT PRO) (ROOT *YOU) (INDEX 1)))
       (OBJECT ((DET ((CAT DET) (ROOT *A) (INDEX 3)))
                (CAT N) (ROOT *COOKIE) (INDEX 4)))
       (SUBCAT *SUBJ-OBJ) (ROOT *WANT) (CAT V) (INDEX 2))
```

Figure 13. The input sentence is a question with a missing auxiliary, not covered by the grammar. Parsing fails in the first and second passes. A successful analysis is obtained in the third pass, with the insertion of an auxiliary.

Table 2
Contribution of Each Pass to Correct Analyses

Pass	No. of Correct Analyses
1 (Unambiguous POS, no robustness)	115
2 (Ambiguous POS)	29
3 (Insertion of AUX)	3
4 (Insertion of NP)	2
5 (Insertion of AUX and NP)	4
6 (One word skipping)	4
All passes (total)	157 (78.5%)

Note—POS, part of speech; AUX, auxilliary; NP, noun phrase.

less meaningful, since our system (especially our grammar) was designed specifically for CHILDES data and would perform poorly on written data, such as newspaper text.

Availability

One of the main goals of this project is to provide data to the language acquisition research community. The results of the research described in this article (a version of the Eve corpus with syntactic annotations for adult utterances), as well as related tools and other resources, are available for research purposes at the CHILDES Web site (http://childes.psy.cmu.edu) or by request from the authors.[1] It is our hope that the data we have produced will be useful in current research efforts in language acquisition,[2] as well as inspire and fuel new research on natural language learning and various aspects of grammar acquisition.

Conclusions and Future Work

Our system is quite effective in producing accurate syntactic annotations for the adult utterances in the Eve corpus. The number of incorrect analyses is acceptably small, making the task of manually checking and possibly correcting the resulting annotations fairly manageable, or even unnecessary if an error rate of about 10% can be tolerated. Most of the utterances that failed to be analyzed by the system were not handled, due to the occurrence of rare syntactic constructions. It is generally believed that the usage of grammar rules that describe specific syntactic constructions follows a Zipfian distribution (Souter, 1990), where the probability of a rule is roughly inversely proportional to its rank. In practice, this means that to achieve complete grammatical coverage of a corpus of significant size, a large number of highly specific rules must be present in the grammar to cover syntactic constructions that appear very few times in the corpus. In our grammar, all (or nearly all) of the more common syntactic structures in the Eve corpus are covered. However, certain sentences contain structures we would be able to cover only with the addition of very specific grammar rules. For example, the following sentences are not covered: (1) We'll buy you another one; (2) Change your record, would you please? and (3) Look what I have.

In the case of Sentence 1, out of more than 15,000 sentences spoken by adults in the Eve corpus, only three contained a ditransitive use of "buy." Although it would be simple to add a rule to allow for ditransitive usage of any verb, the explosion in ambiguity caused by such a rule would certainly result in many more sentences receiving an incorrect analysis than the three we would be able to cover. Instead, our grammar allows ditransitive constructions only for a limited number of verbs, which are listed explicitly. Adding "buy" to the list of possibly ditransitive verbs may solve the problem for those three sentences (while possibly causing errors in other sentences in which "buy" appears), but such an approach would require a large effort to weigh the costs and benefits of allowing ditransitive constructions for every one of the hundreds of verbs in the corpus that are not commonly ditransitive. It should be noted that our system does analyze ditransitive constructions involving verbs that are often present in such constructions (e.g., "give"). We recognize the need to analyze sentences such as Sentence 1 correctly, and we have already begun investigating parsing approaches involving corpus-based techniques that may be more appropriate for sentences involving rare subcategorization frames.

Sentence 2 features topicalization of the verb phrase within a question. This syntactic structure appears only once in over 15,000 sentences. The structure in Sentence 3, where "look" has a clausal complement (instead of a prepositional phrase, as in "look *at* what I have"), appears in only five sentences in the corpus.

Although increasing grammar coverage through the creation of new rules to properly handle these rare constructions is possible (resulting in a grammar with several hundreds of rules), the increased ambiguity resulting from a larger grammar may hurt the overall accuracy of the system. Even though the net effect of such an increase in both coverage and ambiguity remains to be investigated, the amount of work involved in the creation of such rules is hardly justifiable, since the resulting grammar is likely to suffer from overfitting to the corpus used during grammar development. The performance of such a grammar would decrease considerably when analyzing other corpora, where the distribution of rare constructions would most likely be different. A more promising direction toward increasing the coverage of our system is the combination of rule-based parsing and corpus-based methods in natural language processing, such as statistical parsing. Although state-of-the-art statistical parsers lack the linguistic depth to produce output comparable to the f-structures produced by our system, we are currently investigating combina-

Table 3
Causes of Errors in Incorrect Analyses

Cause	No. of Incorrect Analyses
Lack of grammar coverage	
Insertion	7
Overgeneration	5
POS tag error	4
Transcription error	1
Total	17 (8.5%)

Note—POS, part of speech.

Table 4
Causes of Parsing Failures

Cause	No. of Parsing Failures
Lack of grammar coverage	19
Lack of lexical coverage	5
Transcription error	1
Ungrammatical sentence	1
Total	26 (13%)

tions of multiple syntactic analysis strategies to mitigate the problem of grammatical coverage.

Although our efforts to produce syntactic annotations for the child utterances in the corpus (as opposed to the utterances of parents) is still in very early stages, a preliminary evaluation of the current system on a set of such utterances revealed that more than 60% of them could probably be analyzed correctly with the system as is. However, significant changes to the overall system would be necessary for analyzing a high percentage of the child utterances accurately and reliably. We are currently working on a different analysis strategy for child utterances, which acknowledges both the global (utterance level) differences and the local (fragment or constituent level) similarities between the child and the adult languages in the corpus. The analyses produced with this strategy report constituents found in child utterances, without trying to combine them into single global structures when the utterances do not conform to our adult grammar. Our initial heuristic in searching for these constituents is to try to cover as much of the utterance as possible, with as few constituents as possible. Although we recognize the simplistic nature of this approach, our preliminary experiments have yielded very promising levels of accuracy in the analysis of child utterances in the Eve corpus. Further research on analyzing child language is planned as the immediate next step in our work. We also plan to investigate the effectiveness of the current system on other corpora in the CHILDES database and, possibly, the automatic adaptation of the system to other corpora.

REFERENCES

Allen, J. (1995). *Natural language understanding* (2nd ed.). Redwood City, CA: Benjamin/Cummings.

Bresnan, J. (2001). *Lexical-functional syntax*. Oxford: Blackwell.

Brill, E. (1995). Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. *Computational Linguistics*, **21**, 543-565.

Brown, R. (1973). *A first language: The early stages*. Cambridge, MA: Harvard University Press.

Bu¿, F. D., & Waibel, A. (1996). Search in a learnable spoken language parser. In W. Wahlster (Ed.), *Proceedings of the 12th European Conference on Artificial Intelligence* (pp. 562-566). Chickester, U.K.: Wiley.

Charniak, E. (1997). Statistical techniques for natural language parsing. *AI Magazine*, **18**, 33-44.

Charniak, E., & Johnson, M. (2001). Edit detection and parsing for transcribed speech. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics* (pp. 118-126). New Brunswick, NJ: ACL.

Chomsky, N. (1982). *Some concepts and consequences of the theory of government and binding*. Cambridge, MA: MIT Press.

Garside, R., & Smith, N. (1997). A hybrid grammatical tagger: CLAWS4. In R. Garside, G. Leech, & A. McEnery (Eds.), *Corpus annotation: Linguistic information from computer text corpora* (pp. 102-121). London: Longman.

Hauser, R. (1999). *Foundations of computational linguistics*. Berlin: Springer-Verlag.

Lavie, A. (1996). *GLR*: A robust grammar-focused parser for spontaneously spoken language* (Tech. Rep. CMU-CS-96-126). Pittsburgh: Carnegie Mellon University, Computer Science Department.

MacWhinney, B. (Ed.) (1999). *The emergence of language*. Mahwah, NJ: Erlbaum.

MacWhinney, B. (2000). *The CHILDES project: Tools for analyzing talk*. Hillsdale, NJ: Erlbaum.

Marcus, M. P., Santorini, B., & Marcinkiewics, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, **19**, 313-330.

Moerk, E. (1983). *The mother of Eve as a first language teacher*. Norwood, NJ: Ablex.

Moore, R. C. (2000). Improved left-corner chart parsing for large context-free grammars. In *Proceedings of the Sixth International Workshop on Parsing Technologies* (pp. 171-182). New Brunswick, NJ: ACL.

Parisse, C., & Le Normand, M.-T. (2000). Automatic disambiguation of morphosyntax in spoken language corpora. *Behavior Research, Methods, Instruments, & Computers*, **32**, 468-481.

Roark, B. (2001). *Robust probabilistic predictive syntactic processing: Motivations, models, and applications*. Unpublished doctoral dissertation, Brown University, Department of Cognitive and Linguistic Sciences, Providence, RI.

Rosé, C. P., & Lavie, A. (2001). Balancing robustness and efficiency in unification-augmented context-free parsers for large practical applications. In G.-C. Junqua & G. van Noord (Eds.), *Robustness in language and speech technology* (pp. 239-269). Dordrecht: Kluwer.

Souter, C. (1990). Systemic-functional grammars and corpora. In J. Aarts & W. Meijs (Eds.), *Theory and practice in corpus linguistics* (pp. 179-211). Amsterdam: Editions Rodopi.

Villavicencio, A. (2000). The acquisition of word order by a computational learning system. In C. Cardie, W. Daelemans, C. Nedellec, & E. T. K. Sam (Eds.), *Proceedings of the Second Learning Language in Logic Workshop* (pp. 209-218). New Brunswick, NJ: ACL.

NOTES

[1] Researchers interested in obtaining the LCFlex parser (free for research purposes) should contact Kenji Sagae (sagae@cs.cmu.edu) or Alon Lavie (alavie@cs.cmu.edu).

[2] See the word order acquisition investigation in Villavicencio (2000), in which similar data are used in lesser amounts, for an example.

(Manuscript received July 1, 2002;
revision accepted for publication May 29, 2003.)