Parse Fitting and Prose Fixing: Getting a Hold on Ill-formedness[1]

K. Jensen, G. E. Heidorn, L. A. Miller, and Y. Ravin

Computer Sciences Department
IBM Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, New York 10598

Processing syntactically ill-formed language is an important mission of the EPISTLE system. Ill-formed input is treated by this system in various ways. Misspellings are highlighted by a standard spelling checker; syntactic errors are detected and corrections are suggested; and stylistic infelicities are called to the user's attention.

Central to the EPISTLE processing strategy is its technique of *fitted* parsing. When the rules of a conventional syntactic grammar are unable to produce a parse for an input string, this technique can be used to produce a reasonable *approximate* parse that can serve as input to the remaining stages of processing.

This paper first describes the fitting process and gives examples of ill-formed language situations where it is called into play. We then show how a fitted parse allows EPISTLE to carry on its text-critiquing mission where conventional grammars would fail either because of input problems or because of limitations in the grammars themselves. Some inherent difficulties of the fitting technique are also discussed. In addition, we explore how style critiquing relates to the handling of ill-formed input, and how a fitted parse can be used in style checking.

Introduction

In its current form, the EPISTLE system addresses the problems of grammar and style checking of texts written in ordinary English (letters, reports, and manuals, as opposed to novels, plays, and poems). It is this goal that involves us so intimately with the processing of ill-formed language. Grammar checking deals with such errors as disagreement in number between subject and verb; style checking calls attention to such infelicities as sentences that are too wordy or too complex. A standard spelling checker is also included.

Our grammar is written in NLP (Heidorn 1972), an augmented phrase structure language which is currently implemented in LISP/370. At this time the EPISTLE grammar uses syntactic, but not semantic, information. Access to an on-line standard dictionary with about 130,000 entries, including part-of-speech and some other syntactic information (such as transitivity of verbs), makes the system's vocabulary essentially unlimited. We test and improve the grammar by regularly running it on a data base of 2254 sentences from 411 actual business letters. Most of these sentences are rather complicated; the longest contains 63 words, and the average length is 19.2 words.

Since the subset of English represented in business documents is very large, we need a very comprehensive grammar and a robust parser. We take a heuristic approach and consider that a natural language parser can be divided into three parts:

(a) a set of rules, called the *core grammar,* that precisely defines the central, agreed-upon grammatical structures of a language;

(b) peripheral procedures that handle parsing ambiguity: when the core grammar produces more than one parse, these procedures decide which of the multiple parses is to be preferred;

(c) peripheral procedures that handle parsing failure: when the core grammar cannot define an acceptable parse, these procedures assign some reasonable structure to the input.

[1] The work described here is a continuation of work first presented at the Conference on Applied Natural Language Processing in Santa Monica, California (Jensen and Heidorn 1983).

0362-613X/83/030147-14$03.00

In EPISTLE,

(a) the core grammar consists at present of a set of about 300 syntax rules;
(b) ambiguity is resolved by using a metric that ranks alternative parses (Heidorn 1982); and
(c) parse failure is handled by the fitting procedure described here.

In using the terms *core grammar* and *periphery,* we are consciously echoing recent work in generative grammar, but we are applying the terms in a somewhat different way. Core grammar, in current linguistic theory, suggests the notion of a set of very general rules that define universal properties of human language and effectively set limits on the types of grammars any particular language may have; periphery phenomena are those constructions that are peculiar to particular languages and that require rules beyond what the core grammar will provide (Lasnik and Freidin 1981). Our current work is not concerned with the meta-rules of a Universal Grammar. But we have found that a distinction between core and periphery is useful even within a grammar of a particular language – in this case, English.

Parsing in EPISTLE

EPISTLE's parser is written in the NLP programming language, which works with augmented phrase structure rules and with attribute-value records, which are manipulated by the rules. When NLP is used to parse natural language text, the records describe constituents, and the rules put these constituents together to form ever larger constituent (or record) structures. Records contain all the computational and linguistic information associated with words, with larger constituents, and with the parse formation. At this time our grammar is sentence-based; we do not, for instance, create record structures to describe paragraphs. Details of the EPISTLE system and of its core grammar may be found in Heidorn et al. (1982). An earlier overview of the system is presented in Miller et al. (1981).

A close examination of parse trees produced by the core grammar will often reveal branch attachments that are not quite right: for example, semantically incongruous prepositional phrase attachments. In line with our pragmatic parsing philosophy, our core grammar is designed to produce unique approximate parses. (Recall that we currently have access only to syntactic and morphological information about constituents.) In the cases where semantic or pragmatic information is needed before a proper attachment can be made, rather than produce a confusion of multiple parses we force the grammar to try to assign a single parse. This is usually done by forcing some attachments to be made to the closest, or rightmost, available constituent. This strategy only rarely impedes the type of grammar-checking and style-checking that we are working on. And we feel that a single parse with a consistent attachment scheme will yield much more easily to later semantic processing than would a large number of different structures.

The rules of the core grammar (CG) produce a single approximate parse for almost 70% percent of input text, and a small number of multiple parses for another 16%. The CG can always be improved and its coverage extended; work on improving the EPISTLE CG is continual. But the coverage of a core grammar will never reach 100%. For those strings that cannot be fully parsed by rules of the core grammar we use a heuristic *best fit* procedure that produces a reasonable parse structure.

The Fitting Procedure

The fitting procedure begins after the CG rules have been applied in a bottom-up, parallel fashion, but have failed to produce an S node that covers the string. At this point, as a by-product of bottom-up parsing, records are available for inspection that describe the various segments of the input string from many perspectives, according to the rules that have been applied. The term *fitting* has to do with selecting and fitting these pieces of the analysis together in a reasonable fashion.

The fitting algorithm, which is itself implemented as a set of NLP rules, proceeds in two main stages: first, a *head constituent* is chosen; next, *remaining constituents* are fitted in. In our current implementation, candidates for the head are tested preferentially as follows, from most to least desirable:

(a) VPs with tense and subject;
(b) VPs with tense but no subject;
(c) phrases without verbs (e.g., NPs, PPs);
(d) non-finite VPs;
(e) others.

If more than one candidate is found in any category, the one preferred is the widest (covering most text). If there is a tie for widest, the leftmost of those is preferred. If there is a tie for leftmost, the one with the best value for the parse metric is chosen. If there is still a tie (a very unlikely case), an arbitrary choice is made. (Note that we consider a VP to be any segment of text that has a verb as its head element.)

The fitting process is complete if the head constituent covers the entire input string (as would be the case if the string contained just a noun phrase, for example, "Salutations and congratulations"). If the head constituent does not cover the entire string, remaining constituents are added on either side, with the following order of preference:

(a) segments other than VP;
(b) untensed VPs;
(c) tensed VPs.

As with the choice of head, the widest candidate is preferred at each step. The fit moves outward from

```
FITTED|---NP------NOUN*---"Example"
      |---PUNC----":"
      |---VP*|----NP|-----DET-----ADJ*------"Your"
      |      |      |-----NOUN*---"percentage"
      |      |      |-----PP|-----PREP------"of"
      |      |              |-----MONEY*----"$250.00"
      |      |----VERB*---"is"
      |      |----NP------MONEY*--"$187.50"
      |---PUNC----"."
```

Figure 1. An example of a fitted parse tree.

the head, both leftward to the beginning of the string, and rightward to the end, until the entire input string has been fitted into a best approximate parse tree. The overall effect of the fitting process is to select the largest chunk of sentence-like material within a text string and consider it to be central, with left-over chunks of text attached in some reasonable manner.

As a simple example, consider this text string:
"Example: Your percentage of $250.00 is $187.50."
Because this string has a capitalized first word and a period at its end, it is submitted to the core grammar for consideration as a sentence. But it is not a sentence, and so the CG will fail to arrive at a completed parse. However, during processing, the CG will have assigned many structures to its substrings. Looking for a head constituent among these structures, the fitting procedure will first seek VPs with tense and subject. Several are present: "$250.00 is", "percentage of $250.00 is", "$250.00 is $187.50", and so on. The widest and leftmost of these VP constituents is the one which covers the string "Your percentage of $250.00 is $187.50", so it will be chosen as head.

The fitting process then looks for additional constituents to the left, favoring ones other than VP. It finds first the colon, and then the word "Example". In this string the only constituent following the head is the final period, which is duly added. The complete fitted parse is shown in Figure 1.

The form of parse tree used here shows the top-down structure of the string from left to right, with the terminal nodes being the last item on each line. At each level of the tree (in a vertical column), the head element of a constituent is marked with an asterisk. The other elements above and below are pre- and post-modifiers. The highest element of the trees shown here is FITTED, rather than the more usual SENT. (It is important to remember that these parse diagrams are only shorthand representations for the NLP record structures, which contain an abundance of information about the string processed.)

The tree of Figure 1, which would be lost if we restricted ourselves to the rules of the core grammar, is now available for examination, for grammar and style checking, and ultimately for semantic interpretation. It can take its place in the stream of continuous text and be analyzed for what it is – a sentence fragment, interpretable only by reference to other sentences in context.

Further Examples

The fitted parse approach can help to deal with many difficult natural language problems, including fragments, difficult cases of ellipsis, proliferation of rules to handle single phenomena, phenomena for which no rule seems adequate, and punctuation horrors. Each of these is discussed here with examples.

Fragments. There are many of these in running text; they are frequently NPs, as in Figure 2, and include common greetings, farewells, and sentiments. (N.B., most of the examples in this paper are taken from the EPISTLE data base.)

Difficult cases of ellipsis. In the sentence of Figure 3, what we really have semantically is a conjunction of two propositions which, if generated directly, would read: "Secondly, the Annual Commission

```
FITTED|---NP*|----NP|-----AJP-----ADJ*----"Good"
      |      |      |-----NOUN*---"luck"
      |      |----CONJ*---"and"
      |      |----NP|-----AJP-----ADJ*----"good"
      |             |-----NOUN*---"selling"
      |---PUNC----"."
```

Figure 2. Fitted noun phrase (fragment).

```
FITTED|---VP*|----AVP|----ADV*----"Secondly"
       |       |       |----PUNC----","
       |       |----NP|-----AJP-----ADJ*----"the"
       |       |      |-----NP------NOUN*---"Annual"
       |       |      |-----NP------NOUN*---"Commission"
       |       |      |-----NP------NOUN*---"Statement"
       |       |      |-----NOUN*---"total"
       |       |----VERB----"should"
       |       |----VERB*---"be"
       |       |----NP------MONEY*--"$14,682.61"
       |---PUNC----","
       |---AVP-----ADV*----"not"
       |---NP------MONEY*--"$14,682.67"
       |---PUNC----"."
```

Figure 3. Fitted sentence with ellipsis.

```
FITTED|---NP------NOUN*---"Bill"
        |---PUNC----","
        |---VP*|----NP------PRON*---"I"
        |      |----VERB----"'ve"
        |      |----VERB----"been"
        |      |----VERB*---"asked"
        |      |----INFCL|--INFTO---"to"
        |                |--VERB*---"clarify"
        |                |--NP|-----AJP-----ADJ*----"the"
        |                     |-----AJP-----VERB*---"enclosed"
        |                     |-----NOUN*---"letter"
        |---PUNC----"."
```

Figure 4. Fitted sentence with initial vocative.

```
FITTED|---NP|-----AJP-----ADJ*----"Good"
        |     |-----NOUN*---"luck"
        |---PP|-----PREP----"to"
        |     |-----NP------PRON*---"you"
        |     |-----CONJ*---"and"
        |     |-----NP------PRON*---"yours"
        |---CONJ----",and"
        |---VP*|----NP------PRON*---"I"
        |      |----VERB*---"wish"
        |      |----NP------PRON*---"you"
        |      |----NP|-----AJP-----ADJ*----"the"
        |      |      |-----ADV-----"VERY"
        |      |      |-----ADJ*----"best"
        |      |----PP|-----PREP----"in"
        |             |-----AJP-----ADJ*----"your"
        |             |-----AJP-----ADJ*----"future"
        |             |-----NOUN*---"efforts"
        |---PUNC----"."
```

Figure 5. Fitted conjunction of noun phrase with clause.

Statement total should be $14,682.61; the Annual Commission Statement total should *not* be *$14,682.67."* Deletion processes operating on the second proposition are lawful (deletion of identical elements) but massive. It would be unwise to write a core grammar rule that routinely allowed negativized NPs to follow main clauses, because:

(a) the proper analysis of this sentence would be obscured: some pieces – namely, the inferred concepts – are missing from the second part of the surface sentence;
(b) the linguistic generalization would be lost: any two conjoined propositions can undergo deletion of identical (recoverable) elements.

A fitted parse such as Figure 3 allows us to inspect the main clause for syntactic and stylistic deviances, and at the same time makes clear the breaking point between the two propositions and opens the door for a later semantic processing of the elided elements.

Proliferation of rules to handle single phenomena. There are some English constructions that, although they have a fairly simple and unitary form, do not hold anything like a unitary ordering relation within clause boundaries. The vocative is one of these:

(a) *Bill,* I've been asked to clarify the enclosed letter.
(b) I've been asked, *Bill,* to clarify the enclosed letter.
(c) I've been asked to clarify the enclosed letter, *Bill.*

In longer sentences there would be even more possible places to insert the vocative.

Rules could be written that would explicitly allow the placement of a proper name, surrounded by commas, at different positions in the sentence – a different rule for each position. But this solution lacks elegance, makes a simple phenomenon seem complicated, and always runs the risk of overlooking yet one more position where some other writer might insert a vocative. The parse fitting procedure provides an alternative that preserves the integrity of the main clause and allows the vocative to be added onto the structure, as shown, for example, in Figure 4. Other similar phenomena, such as parenthetical expressions, can be handled in this same fashion.

Phenomena for which no rule seems adequate. The sentence "Good luck to you and yours, and I wish you the very best in your future efforts." is, on the face of it, a conjunction of a noun phrase (or NP plus PP) with a finite verb phrase. Such constructions are not usually considered to be fully grammatical, and a core grammar that contains a rule describing this construction ought probably to be called a faulty grammar. Nevertheless, ordinary English correspondence abounds with strings of this sort, and readers have no difficulty construing them. The fitted parse for this sentence in Figure 5 presents the finite clause as its head and adds the remaining constituents in a reasonable fashion. From this structure later semantic processing could infer that "Good luck to you and yours" really means "I express/send/wish good luck to you and yours" – a special case of formalized, ritualized ellipsis.

Punctuation horrors. In any large sample of natural language text, there will be many irregularities of punctuation that, although perfectly understandable to readers, can completely disable an explicit computational grammar. In business text these difficulties are frequent. Some can be caught and corrected by punctuation checkers and balancers. But others cannot, sometimes because, for all their trickiness, they are not really wrong. Yet few grammarians would care to dignify, by describing it with rules of the core grammar, a text string like:

> "Options: A1-(Transmitter Clocked by Dataset) B3-(without the 605 Recall Unit) C5-(with ABC Ring Indicator) D8-(without Auto Answer) E10-(Auto Ring Selective)."

Our parse fitting procedure handles this example by building a string of NPs separated with punctuation marks, as shown in Figure 6. This solution at least enables us to get a handle on the contents of the string.

Benefits

There are two main benefits to be gained from using the fitted parse approach. First, it allows for syntactic processing – for our purposes, grammar and style checking – to proceed in the absence of a perfect parse. Second, it provides a promising structure to submit to later semantic processing routines. And parenthetically, a fitted parse diagram is a great aid to grammar rule debugging. The place where the first break occurs between the head constituent and its pre- or post-modifiers usually indicates fairly precisely where the core grammar failed.

It should be emphasized that a fitting procedure cannot be used as a substitute for explicit rules, and that it in no way lessens the importance of the core grammar. There is a tight interaction between the two components. The success of the fitted parse depends on the accuracy and completeness of the core rules; a fit is only as good as its grammar.

Correcting Syntactic Errors in a Fitted Parse

Suppose the text string in Figure 1 had contained an ungrammaticality, such as disagreement in number between its subject and its verb. Then our troubles would be compounded. There would be two reasons for the CG to reject that string: (a) it is a fragment; and (b) it contains a syntax error.

```
FITTED|---NP------NOUN*---"Options"
      |---PUNC----":"
      |---NP------NOUN*---"A1"
      |---PUNC----"-"
      |---PUNC----"("
      |---NP|-----NP------NOUN*---"Transmitter"
      |     |-----NOUN*---"Clocked"
      |---PP|-----PREP----"by"
      |     |-----NOUN*---"Dataset"
      |---PUNC----")"
      |---NP------NOUN*---"B3"
      |---PUNC----"-"
      |---PP*|----PUNC----"("
      |      |----PREP----"without"
      |      |----AJP-----ADJ*----"the"
      |      |----QUANT---NUM*----"605"
      |      |----NP------NOUN*---"Recall"
      |      |----NOUN*---"Unit"
      |      |----PUNC----")"
      |---NP------NOUN*---"C5"
      |---PUNC----"-"
      |---PP|-----PUNC----"("
      |     |-----PREP----"with"
      |     |-----NP------NOUN*---"ABC"
      |     |-----NP------NOUN*---"Ring"
      |     |-----NOUN*---"Indicator"
      |     |-----PUNC----")"
      |---NP------NOUN*---"D8"
      |---PUNC----"-"
      |---PP|-----PUNC----"("
      |     |-----PREP----"without"
      |     |-----NP------NOUN*---"Auto"
      |     |-----NOUN*---"Answer"
      |     |-----PUNC----")"
      |---NP------NOUN*---"E10"
      |---PUNC----"-"
      |---NP|-----PUNC----"("
      |     |-----NP------NOUN*---"Auto"
      |     |-----NP------NOUN*---"Ring"
      |     |-----NOUN*---"Selective"
      |     |-----PUNC----")"
      |---PUNC----"."
```

Figure 6. Fitted list.

But the CG can recover from many syntax errors: it can diagnose and correct them, producing the parse tree that would be appropriate if the correction were made. Figure 7 illustrates this ability. This number-disagreement phenomenon is fairly common in current American English. The tensed verb seems to want to agree with its closest noun neighbor (in this sentence, "forms...are") rather than with its subject NP ("a carbon copy...is"). A prescriptive rule still insists that subject and verb should agree in number, however, and the EPISTLE grammar provides a correction for such cases. Note that in the last line of Figure 7 the word "are" has been changed to "is". (See Heidorn et al. (1982) for a more thorough discussion of the error correction technique.)

And now the fitting procedure allows us to continue this work even under wildly ungrammatical conditions. Figure 8 is a fitted parse for the string in Figure 1, with a number disagreement error introduced into the fragment.

```
DECL|---NP|-----DET-----ADJ*----"A"
    |      |-----NP------NOUN*---"carbon"
    |      |-----NOUN*---"copy"
    |      |---- PP|-----PREP----"of"
    |              |-----DET-----ADJ*----"the"
    |              |-----NP|-----NOUN*---"Workman"
    |              |       |-----POSS----"'s"
    |              |-----NP------NOUN*---"Compensation"
    |              |-----NOUN*---"forms"
    |---VERB----"are"
    |---VERB*---"enclosed"
    |---PP|-----PREP----"for"
    |     |-----DET-----ADJ*----"your"
    |     |-----NOUN*---"information"
    |---PUNC----"."

GRAMMATICAL ERROR:  SUBJECT-VERB NUMBER DISAGREEMENT.
    A carbon copy...ARE enclosed for your information.
  CONSIDER:
    A carbon copy...IS enclosed for your information.
```

Figure 7. Diagnosis and correction of a syntax error (not a fitted parse).

```
FITTED|---NP------NOUN*---"Example"
      |---PUNC----":"
      |---VP*|----NP|-----DET-----ADJ*----"your"
      |      |      |-----NOUN*---"percentage"
      |      |      |-----PP|-----PREP----"of"
      |      |              |-----MONEY*--"$250.00"
      |      |----VERB*---"are"
      |      |----NP------MONEY*--"$187.50"
      |---PUNC----"."

POSSIBLE GRAMMATICAL ERROR:  SUBJECT-VERB NUMBER DISAGREEMENT.
     Example: your percentage...ARE $187.50.
  CONSIDER:
     Example: your percentage...IS $187.50.
```

Figure 8. Fitted parse containing clause with syntax error.

```
FITTED|---PP*|----PREP----"Between"
      |      |----NP------PRON*---"you"
      |      |----CONJ*---"and"
      |      |----NP------PRON*---"I"
      |---PUNC----"."

POSSIBLE GRAMMATICAL ERROR:  WRONG PRONOUN IN OBJECT POSITION.
      BETWEEN you and I.
  CONSIDER:
      BETWEEN you and ME.
```

Figure 9. Case error in prepositional phrase.

Thanks to the flexibility of this approach, it is possible to check grammar within the smallest imaginable constituents (Figure 9) – and in the largest imaginable (Figure 10).

In summary, there are many different causes for syntactic ill-formedness in the processing of text: misspellings, ungrammaticalities, fragments, crazy punctuation, deficits in the processing grammar, etc. The techniques described here give us a chance to recover from all such cases of ill-formedness. First we develop a core grammar, which itself is capable of detecting spelling mistakes, and of correcting certain

```
FITTED|---VP*|----SUBCL|--CONJ----"Before"
      |      |          |--NP|-----DET-----ADJ*----"an"
      |      |          |    |-----NOUN*---"approval"
      |      |          |--VERB----"can"
      |      |          |--VERB----"be"
      |      |          |--VERB*---"issued"
      |      |----NP------PRON*---"it"
      |      |----VERB----"will"
      |      |----VERB*---"be"
      |      |----AJP|----ADJ*----"necessary"
      |               |----INFCL|--INFTO---"to"
      |                         |--VERB*---"submit"
      |                         |--NP|-----NP------NOUN*   "blueprint"
      |                         |    |-----NOUN*---"drawings"
      |                         |--PP|-----PREP----"in"
      |                         |    |-----AJP-----ADJ*----"triplicate"
      |                         |    |-----NOUN*---"sets"
      |                         |--PP|-----PREP----"on"
      |                              |-----NOUN*---"sheets"
      |                              |-----AJP|----AVP-----ADV*----"no"
      |                                       |----ADJ*----"smaller"
      |                                       |----PP|-----PREP----"than"
      |                                              |-----QUANT---ADJ*---"15"
      |                                              |-----NOUN*---"inches"
      |---CONJ----"and"
      |---PTPRTCL|VERB*---"drawn"
      |          |PP|-----PREP----"to"
      |             |-----DET-----ADJ*----"a"
      |             |-----NOUN*---"scale"
      |             |-----AJP|----AVP-----ADV*    "no"
      |                      |----ADJ*----"smaller"
      |                      |----PP|-----PREP----"than"
      |                      |      |-----NOUN*---"1/8th"
      |                      |      |-----PP|-----PREP----"of"
      |                      |              |-----DET-----ADJ*----"an"
      |                      |              |-----NOUN*---"inch"
      |                      |----PP|-----PREP----"to"
      |                             |-----DET-----ADJ*----"the"
      |                             |-----NOUN*---"foot"
      |---PUNC----"."

POSSIBLE GRAMMATICAL ERROR:  MISSING COMMA.
    Before an approval can be issued it will be necessary...
  CONSIDER:
    Before an approval can be issued, it will be necessary...
  A COMMA IS NEEDED TO DEFINE CLAUSE BOUNDARIES
```

Figure 10. Comma error in long complex sentence.

syntactic mistakes when they occur in otherwise legitimate sentences. To this core grammar we couple a fitting procedure that produces a reasonable best-guess parse for all other text strings, regardless of whether they meet the grammar's criteria for sentencehood. The fitted parse then allows us to check even non-sentences for those categories of syntactic errors that we can correct.

Critiquing Stylistic Ill-Formedness in EPISTLE

The style component of EPISTLE uses the sentence structures provided by the parser as input for stylistic critiquing. It consists of a set of NLP rules that apply to parse trees of sentences, identify stylistic errors and suggest corrections. There is a fundamental difference between style analysis and grammar analysis, however: the grammar rule-system is based on a set of objective syntactic criteria that determine whether the input is well-formed or not. The style rules, by contrast, are based on relative criteria. They place the input on a continuum of stylistic acceptability so that what is a stylistic error becomes a matter of degree.

Types of stylistic ill-formedness.

(1) Punctuation. Stylistic ill-formedness is relative since it depends on both the linguistic and the extra-linguistic contexts. Linguistically, a combination of grammatical factors in the sentence can make a sentence more or less ill-formed. For example, the need for a comma in a compound sentence increases with the length of the conjoined clauses. Thus, in "a decision was reached and the meeting ended," a comma before the "and" is optional; but in "a decision which was moderate enough to satisfy even my objections was reached, and the meeting was finally adjourned," a comma is necessary. An even longer sentence containing several other commas might require a semi-colon before the "and".

To be able to detect the missing comma, the stylistic rule must have access to syntactic information provided by the parser. It has to know that the sentence is compound. Moreover, it has to know that each clause contains its own subject (in this case, "a decision" and "the meeting"), since a compound sentence with only one subject does not take a comma. After all the syntactic conditions are checked and met, the rule measures the length of the clauses to determine how badly the comma is needed. The output is shown in Figure 11.

(2) Other Types of Stylistic Ill-Formedness. The style component of EPISTLE detects other instances of missing or faulty punctuation (a comma at the end of a subordinate clause, no colon before a single noun-phrase, etc). It also addresses some types of complicated grammatical constructions that may impede the reader's comprehension, such as excessive length, excessive noun-modification (e.g. "early childhood thought disorder misdiagnosis"), and excessive negation (e.g., "*neither* the professor, nor his two assistants, who have been working with him on this project, *haven't* noticed the theft"). Some usage violations are signaled, such as "split infinitives" and the usage of "most" instead of "almost"; and finally, some cosmetic changes are proposed when the syntactic structure is too uniform or when there is excessive repetition.

(3) Repetition. Repetition is another instance of the relative nature of stylistic ill-formedness. Generally, repetition of strings is to be avoided; however, some cases of repetition are more acceptable than others. The degree of acceptability depends on the syntactic function of the repeated strings. In "the meeting is very very important," the two instances of "very" have the same syntactic function: they both intensify the adjective "important." This double repetition, lexical and syntactic, is considered poor style. The error correction for this sentence can be seen in Figure 12. By contrast, in "it does not surprise me that that institution no longer exists," the two instances of "that" have different syntactic roles – one is a conjunction; the other, a determiner. This sentence is stylistically more acceptable. Finally, in "what he does does not concern us," the two instances of "does" belong to two different clauses. The style rules accept lexical repetition of this kind.

Correcting stylistic errors in a fitted parse. Because syntactic information is always available, the style rules can apply to fitted parses, as they do to regular sentences. They not only signal stylistic errors within the fitted parse but also assign different degrees of acceptability to different types of fitted parses. As noun phrases are the most commonly encountered type of fragment, the rules accept noun phrases ("My warmest regards to your son") but mark subordinate clauses as incomplete ("Because he refused to sign the papers"), as shown in Figures 13 and 14.

Extra-linguistic factors. The degree of stylistic ill-formedness of a sentence depends on extra-linguistic factors. The use of contracted verb-forms (e.g. "don't", "I'll") is quite acceptable in informal writing; it is to be avoided, though, in formal documents. Style rules should accommodate different degrees of formality. They should also be sensitive to the stylistic norms observed in different domains. In a technical manual, for example, a uniform sentence-pattern is preferred, as it facilitates the reader's comprehension; in freshman compositions, on the other hand, a variety of sentence-patterns is more appropriate as it breaks the monotony. The style component of EPISTLE will address such extra-linguistic factors in addition to the purely linguistic factors. In order to do so, it will present the user with a menu of style options. The selection of the formal option will activate the "no verb-contraction" rule; the selection of the informal option will suppress it. Similarly, the

```
CMPD|---VP|-----NP|-----DET-----ADJ*----"A"
    |     |       |-----NOUN*---"decision"
    |     |       |-----RELCL|--NP------PRON*---"which"
    |     |                  |--VERB*---"was"
    |     |                  |--AJP|----ADJ*----"moderate"
    |     |                        |----ADV-----"enough"
    |     |                        |----INFCL|--INFTO---"to"
    |     |                                  |--VERB*---"satisfy"
    |     |                                  |--NP|-----AJP-----ADV*----"even"
    |     |                                       |-----DET-----ADJ*----"my
    |     |                                       |-----NOUN*---"objections"
    |     |-----VERB----"was"
    |     |-----VERB*---"reached"
    |---CONJ*---"and"
    |---VP|-----NP|-----DET-----ADJ*----"the"
    |     |       |-----NOUN*---"meeting"
    |     |-----VERB----"was"
    |     |-----AVP-----ADV*----"finally"
    |     |-----VERB*---"adjourned"
    |---PUNC----"."

STYLISTIC WEAKNESS:  MISSING COMMA IN COMPOUND SENTENCE.
  WHY NOT HAVE A COMMA BEFORE THE CONJUNCTION?
    ...was reached, and the meeting was finally adjourned.
```

Figure 11. Diagnosis of a punctuation problem.

```
DECL|---NP|-----DET-----ADJ*----"The"
     |     |-----NOUN*---"meeting"
     |---VERB*---"is"
     |---AJP|----AVP-----ADV*----"very"
     |      |----AVP-----ADV*----"very"
     |      |----ADJ*----"important"
     |---PUNC----"."

STYLISTIC WEAKNESS:  REPETITION.
   WHY NOT AVOID REPETITION?
     The meeting is very important.
```

Figure 12. Diagnosis of a repetition problem.

```
FITTED|---NP*|----DET-----ADJ*----"My"
       |      |----AJP-----ADJ*----"warmest"
       |      |----NOUN*---"regards"
       |---PP|-----PREP----"to"
       |     |-----DET-----ADJ*----"your"
       |     |-----NOUN*---"son"
       |---PUNC----"."
```

Figure 13. Fitted noun phrase (no style problems).

```
FITTED|---CONJ----"Because"
      |---VP*|----NP------PRON*---"he"
      |      |----VERB*---"refused"
      |      |----INFCL|--INFTO---"to"
      |                |--VERB*---"sign"
      |                |--NP|-----DET-----ADJ*----"the"
      |                     |-----NOUN*---"papers"
      |---PUNC----"."

POSSIBLE STYLISTIC WEAKNESS:  INCOMPLETE SENTENCE.
  WHY NOT COMPLETE THIS SENTENCE BY ADDING A MAIN CLAUSE?
```

Figure 14. Fragment (subordinate clause) with diagnosis.

technical-writing option will diagnose excessive syntactic variety, whereas the creative-writing option will diagnose monotonous regularity.

Potential Difficulties

Because the error-detection and fitting procedures permit all sorts of non-sentences to survive, they will inevitably increase the number of ambiguities that the system produces, and will require that additional effort be spent to restrict the number of possible parses. This is a difficult but by no means impossible task, since all it entails is the addition of more thorough constraints on the core grammar.

As an example, consider the input string

"What exactly does that 15 months do."

The intended meaning of this string could probably be paraphrased as "What exactly does that 15-month period mean?" But there are two problems with the input. First and most confusingly, the phrase "that 15 months," as given, has a plural head noun ("months") and a singular determiner ("that"). Since the CG cannot understand meaning, it has no way of telling that the given phrase might be an elided form of "that 15-month period." It therefore detects and corrects a syntactic error: number disagreement between premodifier and noun. Secondly, the input string should end with a question mark. But in order to diagnose this error, the grammar needs to realize that a question was intended.

When the problem sentence was submitted to an earlier version of the CG, three parses resulted (Figures 15-17).

The parse in Figure 15 would be appropriate for a sentence such as "Who(ever) exactly does that job prevails." However, it is thoroughly unhelpful for the sentence at hand. It diagnoses two errors, neither of which really exists.

Figure 16 is close to acceptable for the input string. If the time adverbial NP (AVPNP in the parse tree) were replaced by a subject NP, the syntactic structure would be correct for the intended meaning. As things stand, this parse is only 50% helpful: it correctly diagnoses the missing question mark, but it incorrectly insists that "month" should be singular in number.

The third parse (Figure 17) gives the desired single error correction, but it does so on the basis of a totally inappropriate parse. The structure in this figure would fit a question like "Who exactly suffers (in order) that many people might live?"

The current version of the CG blocks the three parses in Figures 15 through 17 on a principled basis. Figure 15 can be blocked by tightening some constraints on the diagnosis of subject-verb number disagreement in fitted parses. Figure 16 is blocked because of the presence of an adverbial NP where the subject NP ought to be. Figure 17 is blocked by stipulating that all subordinate clauses beginning with a "that" conjunction should have modal or subjunctive predicates.

An acceptable parse (Figure 18) is provided when the number agreement restriction is removed from phrases like "that 15 months." This is accomplished by telling the proper rule to ignore number agreement in particular cases that involve a small subset of quantified English time words. Admittedly, the fix would be more pleasing if it were part of a larger scheme for understanding meaning and context. But the correction moves in the right direction, and certainly does not prevent future processing with a more intelligent semantic component. This situation clearly illustrates how error detection results in the addition of finer constraints on the core grammar.

Related Work

The parsing approach closest in spirit to our fitting procedure is that described in Slocum (1983, p. 170): the LRC Machine Translation System uses a "shortest path" technique to construct a "phrasal analysis" of ungrammatical input. With this analysis, phrases can be translated separately, even in the absence of a total sentence parse. Aside from Slocum's work, most of the reports in this field suggest that unparsable or

```
DECL|---NP|-----PRON*---"What"
    |     |-----VP|-----AVP-----ADV*----"exactly"
    |             |-----VERB----"does"
    |             |-----NP|-----DET-----ADJ*----"that"
    |                     |-----QUANT---ADJ*----"15"
    |                     |-----NOUN*---"months"
    |---VERB*---"do"
    |---PUNC----"."

GRAMMATICAL ERROR:  SUBJECT-VERB NUMBER DISAGREEMENT.
    WHAT exactly does that 15 months DO.
  CONSIDER:
    WHAT exactly does that 15 months DOES.

GRAMMATICAL ERROR:  PREMODIFIER-NOUN NUMBER DISAGREEMENT.
    What exactly does THAT...MONTHS do.
  CONSIDER:
    What exactly does THAT...MONTH do.

THE COMBINED GRAMMATICAL CORRECTIONS ARE:
    What exactly does that 15 month does.
```

Figure 15. Two faulty error diagnoses; inappropriate parse.

ill-formed input should be handled by *relaxation techniques,* that is, by relaxing restrictions in the grammar rules in some principled way. This is undoubtedly a useful strategy – one which EPISTLE makes use of, in fact, in its rules for detecting grammatical errors (Heidorn et al. 1982). However, it is questionable whether such a strategy can ultimately succeed in the face of the overwhelming (for all practical purposes, infinite) variety of ill-formedness with which we are faced when we set out to parse truly unrestricted natural language input. If all ill-formedness is *rule-based* (Weischedel and Sondheimer 1981, p. 3), it can only be by some very loose definition of the term *rule,* such as that which might apply to the fitting algorithm described here.

Thus Weischedel and Black (1980) suggest three techniques for responding intelligently to unparsable inputs:

(a) using presuppositions to determine user assumptions; this course is not available to a syntactic grammar like EPISTLE's;
(b) using relaxation techniques;
(c) supplying the user with information about the point where the parse is blocked; this would require an interactive environment, which would not be possible for every type of natural language processing application.

Kwasny and Sondheimer (1981) are strong proponents of relaxation techniques, which they use to handle both cases of clearly ungrammatical structures, such as *co-occurrence violations* like subject/verb disagreement, and cases of perfectly acceptable but difficult constructions (ellipsis and conjunction).

Weischedel and Sondheimer (1982) describe an improved ellipsis processor. No longer is ellipsis handled with relaxation techniques, but by predicting *transformations* of previous parsing paths that would allow for the matching of fragments with plausible contexts. This plan would be appropriate as a next step after the fitted parse, but it does not guarantee a parse for all elided inputs.

Hayes and Mouradian (1981) also use the relaxation method. They achieve flexibility in their parser by relaxing *consistency constraints* (grammatical restrictions, like Kwasny and Sondheimer's co-occurrence violations) and also by relaxing ordering constraints. However, they are working with a restricted-domain semantic system and their approach, as they admit, "does not embody a solution for flexible parsing of natural language in general" (p. 236).

The work of Wilks is heavily semantic and therefore quite different from EPISTLE, but his general philosophy meshes nicely with the philosophy of the fitted parse: "It is proper to prefer the normal ... but it would be absurd ... not to accept the abnormal if it is described" (Wilks 1975, p. 267).

```
DECL|---NP-------PRON*---"What"
    |---AVP------ADV*----"exactly"
    |---VERB-----"does"
    |---AVPNP|---DET-----ADJ*----"that"
    |        |---QUANT---ADJ*----"15"
    |        |---NOUN*---"months"
    |---VERB*----"do"
    |---PUNC-----"."

GRAMMATICAL ERROR:  MISSING QUESTION MARK.
    What exactly does that 15 months do.
  CONSIDER:
    What exactly does that 15 months do?

GRAMMATICAL ERROR:  PREMODIFIER-NOUN NUMBER DISAGREEMENT.
    What exactly does THAT...MONTHS do.
  CONSIDER:
    What exactly does THAT...MONTH do.

THE COMBINED GRAMMATICAL CORRECTIONS ARE:
    What exactly does that 15 month do?
```

Figure 16. One faulty diagnosis, one correct; near-satisfactory parse.

```
DECL|---NP------PRON*---"What"
     |---AVP-----ADV*----"exactly"
     |---VERB*---"does"
     |---SUBCL|--CONJ----"that"
     |        |--NP|------QUANT---ADJ*----"15"
     |        |    |----- NOUN*---"months"
     |        |--VERB*---"do"
     |---PUNC----"."

GRAMMATICAL ERROR:  MISSING QUESTION MARK.
     What exactly does that 15 months do.
   CONSIDER:
     What exactly does that 15 months do?
```

Figure 17. Correct error diagnosis but misleading parse.

```
DECL|---NP------PRON*---"What"
     |---AVP-----ADV*----"exactly"
     |---VERB----"does"
     |---NP|-----DET-----ADJ*----"that"
     |     |-----QUANT---ADJ*----"15"
     |     |-----NOUN*---"months"
     |---VERB*---"do"
     |---PUNC----"."

GRAMMATICAL ERROR:  MISSING QUESTION MARK.
     What exactly does that 15 months do.
   CONSIDER:
     What exactly does that 15 months do?
```

Figure 18. Correct error diagnosis; satisfactory parse.

References

Hayes, P.J. and Mouradian, G.V. 1981 Flexible Parsing. *Am. J. Comp. Ling.* 7(4): 232-242.

Heidorn, G.E. 1972 Natural Language Inputs to a Simulation Programming System. Technical Report NPS-55HD72101A. Monterey, California: Naval Postgraduate School.

Heidorn, G.E. 1982 Experience with an Easily Computed Metric for Ranking Alternative Parses. *Proc. 20th Annual Meeting of the ACL.* Toronto, Canada: 82-84.

Heidorn, G.E.; Jensen, K.; Miller, L.A.; Byrd, R.J.; and Chodorow, M.S. 1982 The EPISTLE Text-Critiquing System. *IBM Systems Journal* 21(3): 305-326.

Jensen, K. and Heidorn, G.E. 1983 The Fitted Parse: 100% Parsing Capability in a Syntactic Grammar of English. *Proc. Conf. on Applied Natural Language Processing.* Santa Monica, California: 93-98.

Kwasny, S.C. and Sondheimer, N.K. 1981 Relaxation Techniques for Parsing Ill-Formed Input. *Am. J. Comp. Ling.* 7(2): 99-108.

Lasnik, H. and Freidin, R. 1981 Core Grammar, Case Theory, and Markedness. *Proc. 1979 GLOW Conf.* Pisa, Italy.

Miller, L.A.; Heidorn, G.E; and Jensen, K. 1981 Text-Critiquing with the EPISTLE System: An Authors's Aid to Better Syntax. *AFIPS Conf. Proc.*, Vol. 50. Arlington, Virginia: 649-655.

Slocum, Jonathan. 1983 A Status Report on the LRC Machine Translation System. *Proc. Conf. on Applied Natural Language Processing.* Santa Monica, California: 166-173.

Weischedel, R.M. and Black, J.E. 1980 Responding Intelligently to Unparsable Inputs. *Am. J. Comp. Ling.* 6(2): 97-109.

Weischedel, R.M. and Sondheimer, N.K. 1981 A Framework for Processing Ill-Formed Input. Research Report. University of Delaware.

Weischedel, R.M. and Sondheimer, N.K. 1982 An Improved Heuristic for Ellipsis Processing. *Proc. 20th Annual Meeting of the ACL.* Toronto, Canada: 85-88.

Wilks, Yorick 1975 An Intelligent Analyzer and Understander of English. *Comm. ACM* 18(5): 264-274.