Robust Processing of Real-World Natural-Language Texts

Jerry R. Hobbs, Douglas E. Appelt, John Bear,
Mabry Tyson, and David Magerman

Artificial Intelligence Center
SRI International
Menlo Park, California

Abstract

It is often assumed that when natural language processing meets the real world, the ideal of aiming for complete and correct interpretations has to be abandoned. However, our experience with TACITUS, especially in the MUC-3 evaluation, has shown that principled techniques for syntactic and pragmatic analysis can be bolstered with methods for achieving robustness. We describe and evaluate a method for dealing with unknown words and a method for filtering out sentences irrelevant to the task. We describe three techniques for making syntactic analysis more robust—an agenda-based scheduling parser, a recovery technique for failed parses, and a new technique called terminal substring parsing. For pragmatics processing, we describe how the method of abductive inference is inherently robust, in that an interpretation is always possible, so that in the absence of the required world knowledge, performance degrades gracefully. Each of these techniques have been evaluated and the results of the evaluations are presented.

1 Introduction

If automatic text processing is to be a useful enterprise, it must be demonstrated that the completeness and accuracy of the information extracted is adequate for the application one has in mind. While it is clear that certain applications require only a minimal level of competence from a system, it is also true that many applications require a very high degree of completeness and accuracy, and an increase in capability in either area is a clear

advantage. Therefore we adopt an extremely high standard against which the performance of a text processing system should be measured: it should recover all information that is implicitly or explicitly present in the text, and it should do so without making mistakes.

This standard is far beyond the state of the art. It is an impossibly high standard for human beings, let alone machines. However, progress toward adequate text processing is best served by setting ambitious goals. For this reason we believe that, while it may be necessary in the intermediate term to settle for results that are far short of this ultimate goal, any linguistic theory or system architecture that is adopted should not be demonstrably inconsistent with attaining this objective. However, if one is interested, as we are, in the potentially successful application of these intermediate-term systems to real problems, it is impossible to ignore the question of whether they can be made efficient enough and robust enough for actual applications.

1.1 The TACITUS System

The TACITUS text processing system has been under development at SRI International for the last six years. This system has been designed as a first step toward the realization of a system with very high completeness and accuracy in its ability to extract information from text. The general philosophy underlying the design of this system is that the system, to the maximum extent possible, should not discard any information that might be semantically or pragmatically relevant to a full, correct interpretation. The effect of this design philosophy on the system architecture is manifested in the following characteristics:

- TACITUS relies on a large, comprehensive lexicon containing detailed syntactic subcategorization information for each lexical item.
- TACITUS produces a parse and semantic interpretation of each sentence using a comprehensive grammar of English in which different possible predicate-argument relations are associated with different syntactic structures.
- TACITUS relies on a general abductive reasoning mechanism to uncover the implicit assumptions necessary to explain the coherence of the explicit text.

These basic design decisions do not by themselves distinguish TACITUS from a number of other natural-language processing systems. However, they

are somewhat controversial given the intermediate goal of producing systems that are useful for existing applications. Criticism of the overall design with respect to this goal centers on the following observations:

- The syntactic structure of English is very complex, and no grammar of English has been constructed that has complete coverage of the syntax one encounters in real-world texts. Much of the text that needs to be processed will lie outside the scope of the best grammars available, and therefore cannot be understood by a system that relies on a complete syntactic analysis of each sentence as a prerequisite to other processing.

- Typical sentences in newspaper articles are about 25-30 words in length. Many sentences are much longer. Processing strategies that rely on producing a complete syntactic analysis of such sentences will be faced with a combinatorially intractable task, assuming in the first place that the sentences lie within the language described by the grammar.

- Any grammar that successfully accounts for the range of syntactic structures encountered in real-world texts will necessarily produce many ambiguous analyses of most sentences. Assuming that the system can find the possible analyses of a sentence in a reasonable period of time, it is still faced with the problem of choosing the correct one from the many competing ones.

Designers of application-oriented text processing systems have adopted a number of strategies for dealing with these problems. Such strategies involve de-emphasizing the role of syntactic analysis (Jacobs et al., 1991), producing partial parses with stochastic or heuristic parsers (de Marcken, 1990; Weischedel et al 1991) or resorting to weaker syntactic processing methods such as conceptual or case-frame based parsing (e.g., Schank and Riesbeck, 1981) or template matching techniques (Jackson et al., 1991). A common feature shared by these weaker methods is that they ignore certain information that is present in the text, which could be extracted by a more comprehensive analysis. The information that is ignored may be irrelevant to a particular application, or relevant in only an insignificant handful of cases, and thus we cannot argue that approaches to text processing based on weak or even nonexistent syntactic and semantic analysis are doomed to failure in all cases and are not worthy of further investigation. However, it is not obvious how such methods can scale up to handle fine distinctions

in attachment, scoping, and inference, although some recent attempts have been made in this direction (Cardie and Lehnert, 1991).

In the development of TACITUS, we have chosen a design philosophy that assumes that a complete and accurate analysis of the text is being undertaken. In this paper we discuss how issues of robustness are approached from this general design perspective. In particular, we demonstrate that

- a statistical keyword filter can select the sentences to be processed, with a great savings in time and little loss of relevant information.
- useful partial analyses of the text can be obtained in cases in which the text is not grammatical English, or lies outside the scope of the grammar's coverage,
- substantially correct parses of sentences can be found without exploring the entire search space for each sentence,
- useful pragmatic interpretations can be obtained using general reasoning methods, even in cases in which the system lacks the necessary world knowledge to resolve all of the pragmatic problems posed in a sentence, and
- all of this processing can be done within acceptable bounds on computational resources.

Our experience with TACITUS suggests that extension of the system's capabilities to higher levels of completeness and accuracy can be achieved through incremental modifications of the system's knowledge, lexicon and grammar, while the robust processing techniques discussed in the following sections make the system usable for intermediate term applications. We have evaluated the success of the various techniques discussed here, and conclude from this evaluation that TACITUS offers substantiation of our claim that a text processing system based on principles of complete syntactic, semantic and pragmatic analysis need not be too brittle or computationally expensive for practical applications.

1.2 Evaluating the System

SRI International participated in the recent MUC-3 evaluation of text-understanding systems (Sundheim, 1991). The methodology chosen for this

evaluation was to score a system's ability to fill in slots in templates summarizing the content of newspaper articles approximately one page in length on Latin American terrorism. The template-filling task required identifying, among other things, the perpetrators and victims of each terrorist act described in the articles, the occupation of the victims, the type of physical entity attacked or destroyed, the date, the location, and the effect on the targets. Frequently, articles described multiple incidents, while other texts were completely irrelevant.

An example of a relatively short terrorist report is the following from a news report dated March 30, 1989:

> A cargo train running from Lima to Lorohia was derailed before dawn today after hitting a dynamite charge.
> Inspector Eulogio Flores died in the explosion.
> The police reported that the incident took place past midnight in the Carahuaichi-Jaurin area.

Some of the corresponding database entries are as follows:

Incident: Date	30 Mar 89
Incident: Location	Peru: Carahuaichi-Jaurin (area)
Incident: Type	Bombing
Physical Target: Description	"cargo train"
Physical Target: Effect	Some Damage: "cargo train"
Human Target: Name	"Eulogio Flores"
Human Target: Description	"inspector": "Eulogio Flores"
Human Target: Effect	Death: "Eulogio Flores"

The fifteen participating sites were given a development corpus of 1300 such texts in October 1990. In early February 1991, the systems were tested on 100 new messages (the TST1 corpus), and a workshop was held to debug the testing procedure. In May 1991 the systems were tested on a new corpus of 100 messages (TST2); this constituted the final evaluation. The results were reported at a workshop at NOSC in May 1991.

The principal measures in the MUC-3 evaluation were recall and precision. *Recall* is the number of answers the system got right divided by the number of possible right answers. It measures how comprehensive the system is in its extraction of relevant information. *Precision* is the number of answers the system got right divided by the number of answers the system

gave. It measures the system's accuracy. For example, if there are 100 possible answers and the system gives 80 answers and gets 60 of them right, its recall is 60% and its precision is 75%.

The database entries are organized into templates, one for each relevant event. In an attempt to factor out some of the conditionality among the database entries, recall and precision scores were given, for each system, for three different sets of templates:

- Templates for events the system correctly identified (Matched Templates).
- Matched templates, plus templates for events the system failed to identify (Matched/Missing).
- All templates, including spurious templates the system generated.

The results for TACITUS on the TST2 corpus were as follows.

	Recall	Precision
Matched Templates	44%	65%
Matched/Missing	25%	65%
All Templates	25%	48%

Our precision was the highest of any of the sites participating in the evaluation. Our recall was somewhere in the middle.

We also ran our system, configured identically to the TST2 run, on the first 100 messages of the development set. The results were as follows:

	Recall	Precision
Matched Templates	46%	64%
Matched/Missing	37%	64%
All Templates	37%	53%

Here recall was considerably better, as would be expected since the messages were used for development.

Although pleased with these overall results, a subsequent detailed analysis of our performance on the first 20 messages of the 100-message test set is much more illuminating for evaluating the success of the particular robust processing strategies we have chosen. In the remainder of this paper, we discuss the impact of the robust processing methods in the light of this detailed analysis.

We will divide our discussion into four parts: handling unknown words, our statistical relevance filter, syntactic analysis, and pragmatic interpretation. The performance of each of these processes will be described for Message 99 of TST1 (given in the Appendix) or on Message 100 of the development set (given in Section 5). Then their performance on the first 20 messages of TST2 will be summarized.

2 Handling Unknown Words

When an unknown word is encountered, three processes are applied sequentially.

1. Spelling Correction. A standard algorithm for spelling correction is applied, but only to words longer than four letters.

2. Hispanic Name Recognition. A statistical trigram model for distinguishing between Hispanic surnames and English words was developed and is used to assign the category `Last-Name` to some of the words that are not spell-corrected.

3. Morphological Category Assignment. Words that are not spell-corrected or classified as last names, are assigned a category on the basis of morphology. Words ending in "-ing" or "-ed" are classified as verbs. Words ending in "-ly" are classified as adverbs. All other unknown words are taken to be nouns. This misses adjectives entirely, but this is generally harmless, because the adjectives incorrectly classified as nouns will still parse as prenominal nouns in compound nominals. The grammar will recognize an unknown noun as a name in the proper environment.

There were no unknown words in Message 99, since all the words used in the TST1 set had been entered into the lexicon.

In the first 20 messages of TST2, there were 92 unknown words. Each of the heuristics either did or did not apply to the word. If it did, the results could have been correct, harmless, or wrong. An example of a harmless spelling correction is the change of "twin-engined" to the adjective "twin-engine". A wrong spelling correction is the change of the verb "nears" to the preposition "near". An example of a harmless assignment of Hispanic surname to a word is the Japanese name "Akihito". A wrong assignment is the word "panorama". A harmless morphological assignment of a category

to a word is the assignment of `Verb` to "undispute" and "originat". A wrong assignment is the assignment of `Noun` to "upriver".

The results were as follows:

	Unknown	Applied	Correct	Harmless	Wrong
Spelling	92	25	8	12	5
Surname	67	20	8	10	2
Morphological	47	47	29	11	7

If we look just at the Correct column, only the morphological assignment heuristic is at all effective, giving us 62%, as opposed to 32% for spelling correction and 40% for Hispanic surname assignment. However, harmless assignments are often much better than merely harmless; they often allow a sentence to parse that otherwise would not, thereby making other information in the sentence available to pragmatic interpretation. If we count both the Correct and Harmless columns, then spelling correction is effective 80% of the time, Hispanic surname assignment 90% of the time, and morphological assignment 86%.

Using the three heuristics in sequence meant that 85% of the unknown words were handled either correctly or harmlessly.

3 Statistical Relevance Filter

The relevance filter works on a sentence-by-sentence basis and decides whether the sentence should be submitted to further processing. It consists of two subcomponents—a statistical relevance filter and a keyword antifilter.

The statistical relevance filter was developed from our analysis of the training data. We went through the 1300-text development set and identified the relevant sentences. For each unigram, bigram, and trigram, we determined an n-gram-score by dividing the number of occurrences in the relevant sentences by the total number of occurrences. A subset of these n-grams was selected as being particularly diagnostic of relevant sentences. A sentence score was then computed as follows. It was initialized to the n-gram score for the first diagnostic n-gram in the sentence. For subsequent nonoverlapping, diagnostic n-grams it was updated by the formula

$$\text{sentence score} \leftarrow \text{sentence score} + (1 - \text{sentence score}) * \text{next n-gram score}$$

This formula normalizes the sentence score to between 0 and 1. Because of the second term of this formula, each successive n-gram score "uses up"

some portion of the distance remaining between the current sentence score and 1.

Initially, a fixed threshold for relevance was used, but this gave poor results. The threshold for relevance is now therefore contextually determined for each text, based on the average sentence score for the sentences in the text, by the formula

$$.3 + .65 * (1- \text{ average sentence score})$$

Thus, the threshhold is lower for texts with many relevant sentences, as seems appropriate. This cutoff formula was chosen so that we would identify 85% of the relevant sentences and overgenerate by no more than 300%. The component is now apparently much better than this.

The keyword antifilter was developed in an effort to capture those sentences that slip through the statistical relevance filter. The antifilter is based on certain keywords. If a sentence in the text proves to contain relevant information, the next few sentences will be declared relevant as well if they contain those keywords.

In Message 99, the statistical filter determined nine sentences to be relevant. All of these were actually relevant except for one, Sentence 13. No relevant sentences were missed. The keyword antifilter decided incorrectly that two other sentences were relevant, Sentences 8 and 9. This behavior is typical.

In the first 20 messages of the TST2 set, the results were as follows: There were 370 sentences. The statistical relevance filter produced the following results:

	Actually Relevant	Actually Irrelevant
Judged Relevant	42	33
Judged Irrelevant	9	286

Thus, recall was 82% and precision was 56%. These results are excellent. They mean that by using this filter alone we would have processed only 20% of the sentences in the corpus, processing less than twice as many as were actually relevant, and missing only 18% of the relevant sentences.

The results of the keyword antifilter were as follows:

	Actually Relevant	Actually Irrelevant
Judged Relevant	5	57
Judged Irrelevant	4	229

Clearly, the results here are not nearly as good. Recall was 55% and precision was 8%. This means that to capture half the remaining relevant sentences, we had to nearly triple the number of irrelevant sentences we processed. Using the filter and antifilter in sequence, we had to process 37% of the sentences. Our conclusion is that if the keyword antifilter is to be retained, it must be refined considerably.

Incidentally, of the four relevant sentences that escaped both the filter and the antifilter, two contained only redundant information that could have been picked up elsewhere in the text. The other two contained information essential to 11 slots in templates, lowering overall recall by about 1%.

4 Syntactic Analysis

Robust syntactic analysis requires a very broad coverage grammar and means for dealing with sentences that do not parse, whether because they fall outside the coverage of the grammar or because they are too long for the parser. The grammar used in TACITUS is that of the DIALOGIC system, developed in 1980-81 essentially by constructing the union of the Linguistic String Project Grammar (Sager, 1981) and the DIAGRAM grammar (Robinson, 1982) which grew out of SRI's Speech Understanding System research in the 1970s. Since that time it has been considerably enhanced. It consists of about 160 phrase structure rules. Associated with each rule is a "constructor" expressing the constraints on the applicability of that rule, and a "translator" for producing the logical form.

The grammar is comprehensive and includes subcategorization, sentential complements, adverbials, relative clauses, complex determiners, the most common varieties of conjunction and comparison, selectional constraints, some coreference resolution, and the most common sentence fragments. The parses are ordered according to heuristics encoded in the grammar.

The parse tree is translated into a logical representation of the meaning of the sentence, encoding predicate-argument relations and grammatical subordination relations. In addition, it regularizes to some extent the role assignments in the predicate-argument structure, and handles arguments inherited from control verbs.

Our lexicon contains about 20,000 entries, including about 2000 personal names and about 2000 location, organization, or other names. This number does not include morphological variants, which are handled in a separate morphological analyzer.

The syntactic analysis component was remarkably successful in the MUC-3 evaluation. This was due primarily to three innovations.

- An agenda-based scheduling chart parser.
- A recovery heuristic for unparsable sentences that found the best sequence of grammatical fragments.
- The use of "terminal substring parsing" for very long sentences.

Each of these techniques will be described in turn, with statistics on their performance in the MUC-3 evaluation.

4.1 Performance of the Scheduling Parser and the Grammar

The fastest parsing algorithms for context-free grammars make use of prediction based on left context to limit the number of nodes and edges the parser must insert into the chart. However, if robustness in the face of possibly ungrammatical input or inadequate grammatical coverage is desired, such algorithms are inappropriate. Although the heuristic of choosing the longest possible substring beginning at the left that can be parsed as a sentence could be tried (e.g. Grishman and Sterling, 1989), sometimes, the best fragmentary analysis of a sentence can only be found by parsing an intermediate or terminal substring that excludes the leftmost words. For this reason, we feel that bottom-up parsing without strong constraints based on left context is required for robust syntactic analysis.

Bottom-up parsing is favored for its robustness, and this robustness derives from the fact that a bottom-up parser will construct nodes and edges in the chart that a parser with top-down prediction would not. The obvious problem is that these additional nodes do not come without an associated cost. Moore and Dowding (1991) observed a ninefold increase in time required to parse sentences with a straightforward CKY parser as opposed to a shift-reduce parser. Prior to November 1990, TACITUS employed a simple, exhaustive, bottom-up parser with the result that sentences of more than 15 to 20 words were impossible to parse in reasonable time. Since the average length of a sentence in the MUC-3 texts is approximately 27 words, such techniques were clearly inappropriate for the application.

We addressed this problem by adding an agenda mechanism to the bottom-up parser, based on Kaplan (1973), as described in Winograd (1983).

The purpose of the agenda is to allow us to order nodes (complete constituents) and edges (incomplete constituents) in the chart for further processing. As nodes and edges are built, they are rated according to various criteria for how likely they are to figure in a correct parse. This allows us to schedule which constituents to work with first so that we can pursue only the most likely paths in the search space and find a parse without exhaustively trying all possibilities. The scheduling algorithm is simple: explore the ramifications of the highest scoring constituents first.

In addition, there is a facility for pruning the search space. The user can set limits on the number of nodes and edges that are allowed to be stored in the chart. Nodes are indexed on their atomic grammatical category (i.e., excluding features) and the string position at which they begin. Edges are indexed on their atomic grammatical category and the string position where they end. The algorithm for pruning is simple: Throw away all but the n highest scoring constituents for each category/string-position pair.

It has often been pointed out that various standard parsing strategies correspond to various scheduling strategies in an agenda-based parser. However, in practical parsing, what is needed is a scheduling strategy that enables us to pursue only the most likely paths in the search space and to find the correct parse without exhaustively trying all possibilities. The literature has not been as illuminating on this issue.

We designed our parser to score each node and edge on the basis of three criteria:

- The length of the substring spanned by the constituent.
- Whether the constituent is a node or an edge, that is, whether the constituent is complete or not.
- The scores derived from the preference heuristics that have been encoded in DIALOGIC over the years, described and systematized in Hobbs and Bear (1990).

However, after considerable experimentation with various weightings, we concluded that the length and completeness factors failed to improve the performance at all over a broad range of sentences. Evidence suggested that a score based on preference factor alone produces the best results. The reason a correct or nearly correct parse is found so often by this method is that these preference heuristics are so effective.

In Message 99, of the 11 sentences determined to be relevant, only Sentence 14 did not parse. This was due to a mistake in the sentence itself,

the use of "least" instead of "at least". Of the 10 sentences that parsed, 5 were completely correct, including the longest, Sentence 7 (27 words in 77 seconds). There were three mistakes (Sentences 3, 4, and 9) in which the preferred multiword senses of the phrases "in front of" and "Shining Path" lost out to their decompositions. There were two attachment mistakes. In Sentence 3 the relative clause was incorrectly attached to "front" instead of "embassy", and in Sentence 8, "in Peru" was attached to "attacked" instead of "interests". All of these errors were harmless. In addition, in Sentence 5, "and destroyed the two vehicles" was grouped with "Police said ..." instead of "the bomb broke windows"; this error is not harmless. In every case the grammar prefers the correct reading. We believe the mistakes were due to a problem in the scheduling parser that we discovered the week of the evaluation but felt was too deep and far-reaching to attempt to fix at that point.

In the first 20 messages of the test set, 131 sentences were given to the scheduling parser, after statistically based relevance filtering. A parse was produced for 81 of the 131 sentences, or 62%. Of these, 43 (or 33%) were completely correct, and 30 more had three or fewer errors. Thus, 56% of the sentences were parsed correctly or nearly correctly.

These results naturally vary depending on the length of the sentences. There were 64 sentences of under 30 morphemes (where by "morpheme" we mean a word stem or an inflectional affix). Of these, 37 (58%) had completely correct parses and 48 (75%) had three or fewer errors. By contrast, the scheduling parser attempted only 8 sentences of more than 50 morphemes, and only two of these parsed, neither of them even nearly correctly.

Of the 44 sentences that would not parse, nine were due to problems in lexical entries. Eighteen were due to shortcomings in the grammar, primarily involving adverbial placement and less than fully general treatment of conjunction and comparatives. Six were due to garbled text. The causes of eleven failures to parse have not been determined. These errors are spread out evenly across sentence lengths. In addition, seven sentences of over 30 morphemes hit the time limit we had set, and terminal substring parsing, as described below, was invoked.

A majority of the errors in parsing can be attributed to five or six causes. Two prominent causes are the tendency of the scheduling parser to lose favored close attachments of conjuncts and adjuncts near the end of long sentences, and the tendency to misanalyze the string

$$[[\text{Noun Noun}]_{NP}\ \text{Verb}_{trans}\ \text{NP}]_S$$

as

$$[\text{Noun}]_{NP}\ [\text{Noun Verb}_{ditrans}\ ()\ \text{NP}]_{S/NP},$$

again contrary to the grammar's preference heuristics. We believe that most of these problems are due to the fact that the work of the scheduling parser is not distributed evenly enough across the different parts of the sentence, and we expect that this difficulty could be solved with relatively little effort.

Our results in syntactic analysis are quite encouraging since they show that a high proportion of a corpus of long and very complex sentences can be parsed nearly correctly. However, the situation is even better when one considers the results for the best-fragment-sequence heuristic and for terminal substring parsing.

4.2 Recovery from Failed Parses

When a sentence does not parse, we attempt to span it with the longest, best sequence of interpretable fragments. The fragments we look for are main clauses, verb phrases, adverbial phrases, and noun phrases. They are chosen on the basis of length and their preference scores, favoring length over preference score. We do not attempt to find fragments for strings of less than five morphemes. The effect of this heuristic is that even for sentences that do not parse, we are able to extract nearly all of the propositional content.

For example, sentence (14) of Message 99 in the TST1 corpus,

> The attacks today come after Shining Path attacks during which least 10 buses were burned throughout Lima on 24 Oct.

did not parse because of the use of "least" instead of "at least". Hence, the best fragment sequence was sought. This consisted of the two fragments "The attacks today come after Shining Path attacks" and "10 buses were burned throughout Lima on 24 Oct." The parses for both these fragments were completely correct. Thus, the only information lost was from the three words "during which least". Frequently such information can be recaptured by the pragmatics component. In this case, the burning would be recognized as a consequence of an attack, and inconsistent dates would rule out "the attacks today".

In the first 20 messages of the TST2 corpus, a best sequence of fragments was sought for the 44 sentences that did not parse for reasons other than timing. A sequence was found for 41 of these; the other three were too short,

with problems in the middle. The average number of fragments in a sequence was two. This means that an average of only one structural relationship was lost. Moreover, the fragments covered 88% of the morphemes. That is, even in the case of failed parses, 88% of the propositional content of the sentences was made available to pragmatics. Frequently the lost propositional content is from a preposed or postposed, temporal or causal adverbial, and the actual temporal or causal relationship is replaced by simple logical conjunction of the fragments. In such cases, much useful information is still obtained from the partial results.

For 37% of the 41 sentences, correct syntactic analyses of the fragments were produced. For 74%, the analyses contained three or fewer errors. Correctness did not correlate with length of sentence.

These numbers could probably be improved. We favored the longest fragment regardless of preference scores. Thus, frequently a high-scoring main clause was rejected because by tacking a noun onto the front of that fragment and reinterpreting the main clause bizarrely as a relative clause, we could form a low-scoring noun phrase that was one word longer. We therefore plan to experiment with combining length and preference score in a more intelligent manner.

4.3 Terminal Substring Parsing

For sentences of longer than 60 words and for faster, though less accurate, parsing of shorter sentences, we developed a technique we are calling *terminal substring parsing*. The sentence is segmented into substrings, by breaking it at commas, conjunctions, relative pronouns, and certain instances of the word "that". The substrings are then parsed, starting with the last one and working back. For each substring, we try either to parse the substring itself as one of several categories or to parse the entire set of substrings parsed so far as one of those categories. The best such structure is selected, and for subsequent processing, that is the only analysis of that portion of the sentence allowed. The categories that we look for include main, subordinate, and relative clauses, infinitives, verb phrases, prepositional phrases, and noun phrases.

A simple example is the following, although we do not apply the technique to sentences or to fragments this short.

George Bush, the president, held a press conference yesterday.

This sentence would be segmented at the commas. First "held a press con-

ference yesterday" would be recognized as a VP. We next try to parse both "the president" and "the president, VP". The string "the president, VP" would not be recognized as anything, but "the president" would be recognized as an NP. Finally, we try to parse both "George Bush" and "George Bush, NP, VP". "George Bush, NP, VP" is recognized as a sentence with an appositive on the subject.

This algorithm is superior to a more obvious algorithm we had been considering earlier, namely, to parse each fragment individually in a left-to-right fashion and then to attempt to piece the fragments together. The latter algorithm would have required looking inside all but the last of the fragments for possible attachment points. This problem of recombining parts is in general a difficulty that is faced by parsers that produce phrasal rather than sentential parses (e.g., Weischedel et al., 1991). However, in terminal substring parsing, this recombining is not necessary, since the favored analyses of subsequent segments are already available when a given segment is being parsed.

The effect of this terminal substring parsing technique is to give only short inputs to the parser, without losing the possibility of getting a single parse for the entire long sentence. Suppose, for example, we are parsing a 60-word sentence that can be broken into six 10-word segments. At each stage, we will only be parsing a string of ten to fifteen "words", the ten words in the segment, plus the nonterminal symbols dominating the favored analyses of the subsequent segments. When parsing the sentence-initial 10-word substring, we are in effect parsing at most a "15-word" string covering the entire sentence, consisting of the 10 words plus the nonterminal symbols covering the best analyses of the other five substrings. In a sense, rather than parsing one very long sentence, we are parsing six fairly short sentences, thus avoiding the combinatorial explosion.

Although this algorithm has given us satisfactory results in our development work, its numbers from the MUC-3 evaluation do not look good. This is not surprising, given that the technique is called on only when all else has already failed. In the first 20 messages of the test set, terminal substring parsing was applied to 14 sentences, ranging from 34 to 81 morphemes in length. Only one of these parsed, and that parse was not good. However, sequences of fragments were found for the other 13 sentences. The average number of fragments was 2.6, and the sequences covered 80% of the morphemes. None of the fragment sequences was without errors. However, eight of the 13 had three or fewer mistakes. The technique therefore allowed us to make use of much of the information in sentences that have hitherto been

beyond the capability of virtually all parsers.

5 Robust Pragmatic Interpretation

When a sentence is parsed and given a semantic interpretation, the relationship between this interpretation and the information previously expressed in the text as well as the interpreter's general knowledge must be established. Establishing this relationship comes under the general heading of pragmatic interpretation. The particular problems that are solved during this step include

- Making explicit information that is only implicit in the text. This includes, for example, explicating the relationship underlying a compound nominal, or explicating causal consequences of events or states mentioned explicitly in the text.
- Determining the implicit entities and relationships referred to metonymically in the text.
- Resolving anaphoric references and implicit arguments.
- Viewing the text as an instance of a schema that makes its various parts coherent.

TACITUS interprets a sentence pragmatically by proving that its logical form follows from general knowledge and the preceding text, allowing a minimal set of assumptions to be made. In addition, it is assumed that the set of events, abstract entities, and physical objects mentioned in the text is to be consistently minimized. The best set of assumptions necessary to find such a proof can be regarded as an explanation of its truth, and constitutes the implicit information required to produce the interpretation (Hobbs et al., 1990). The minimization of objects and events leads to anaphora resolution by assuming that objects that share properties are identical, when it is consistent to do so.

In the MUC-3 domain, explaining a text involves viewing it as an instance of one of a number of explanatory schemas representing terrorist incidents of various types (e.g. bombing, arson, assassination) or one of several event types that are similar to terrorist incidents, but explicitly excluded by the task requirements (e.g. an exchange of fire between military groups of opposing factions). This means that assumptions that fit into

incident schemas are preferred to assumptions that do not, and the schema that ties together the most assumptions is the best explanation.

In this text interpretation task, the domain knowledge performs two primary functions:

1. It relates the propositions expressed in the text to the elements of the underlying explanatory schemas.

2. It enables and restricts possible coreferences for anaphora resolution.

It is clear that much domain knowledge may be required to perform these functions successfully, but it is not necessarily the case that more knowledge is always better. If axioms are incrementally added to the system to cover cases not accounted for in the existing domain theory, it is possible that they can interact with the existing knowledge in such a way that the reasoning process becomes computationally intractable, and the unhappy result would be failure to find an interpretation in cases in which the correct interpretation is entailed by the system's knowledge. In a domain as broad and diffuse as the terrorist domain, it is often impossible to guarantee by inspection that a domain theory is not subject to such combinatorial problems.

The goal of robustness in interpretation therefore requires one to address two problems: a system must permit a graceful degradation of performance in those cases in which knowledge is incomplete, and it must extract as much information as it can in the face of a possible combinatorial explosion.

The general approach of abductive text interpretation addresses the first problem through the notion of a "best interpretation." The best explanation, given incomplete domain knowledge, can succeed at relating some propositions contained in the text to the explanatory schemas, but may not succeed for all propositions. The combinatorial problems are addressed through a particular search strategy for abductive reasoning described as *incremental refinement of minimal information proofs.*

The abductive proof procedure as employed by TACITUS (Stickel, 1988) will always be able to find *some* interpretation of the text. In the worst case—the absence of any commonsense knowledge that would be relevant to the interpretation of a sentence—the explanation offered would be found by assuming each of the literals to be proved. Such a proof is called a "minimal information proof" because no schema recognition or explication of implicit relationships takes place. However, the more knowledge the system has, the more implicit information can be recovered.

Because a minimal information proof is always available for any sentence of the text that is internally consistent, it provides a starting point for incremental refinement of explanations that can be obtained at next to no cost. TACITUS explores the space of abductive proofs by finding incrementally better explanations for each of the constituent literals. A search strategy is adopted that finds successive explanations, each of which is better than the minimal information proof. This process can be halted at any time in a state that will provide at least *some* intermediate results that are useful for subsequent interpretation and template filling.

Consider again Message 100 from the MUC-3 development corpus:

> A cargo train running from Lima to Lorohia was derailed before dawn today after hitting a dynamite charge.
> Inspector Eulogio Flores died in the explosion.
> The police reported that the incident took place past midnight in the Carahuaichi-Jaurin area.

The correct interpretation of this text requires recovering certain implicit information that relies on commonsense knowledge. The compound nominal phrase "dynamite charge" must be interpreted as "charge composed of dynamite." The interpretation requires knowing that dynamite is a substance, that substances can be related via compound nominal relations to objects composed of those substances, that things composed of dynamite are bombs, that hitting bombs causes them to explode, that exploding causes damage, that derailing is a type of damage, and that planting a bomb is a terrorist act. The system's commonsense knowledge base must be rich enough to derive each of these conclusions if it is to recognize the event described as a terrorist act, since all derailings are not the result of bombings. This example underscores the need for fairly extensive world knowledge in the comprehension of text. If the knowledge is missing, the correct interpretation cannot be found. (A few simple heuristics can capture some of the information, but at the expense of accuracy.)

However, if there is missing knowledge, all is not necessarily lost. If, for example, the knowledge was missing that hitting a bomb causes it to explode, the system could still hypothesize the relationship between the charge and the dynamite to reason that a bomb was placed. When processing the next sentence, the system may have trouble figuring out the time and place of Flores's death if it can't associate the explosion with hitting the bomb. However, if the second sentence were "The Shining Path claimed that their guerrillas had planted the bomb," the partial information would be sufficient

to allow "bomb" to be resolved to dynamite charge, thereby connecting the event described in the first sentence with the event described in the second.

It is difficult to evaluate the pragmatic interpretation component individually, since to a great extent its success depends on the adequacy of the syntactic analysis it operates on. However, in examining the first 20 messages of the MUC-3 test set in detail, we attempted to pinpoint the reason for each missing or incorrect entry in the required templates.

There were 269 such mistakes, due to problems in 41 sentences. Of these, 124 are attributable to pragmatic interpretation. We have classified their causes into a number of categories, and the results are as follows.

Reason	Mistakes
Simple Axiom Missing	49
Combinatorics	28
Unconstrained Identity Assumptions	25
Complex Axioms or Theory Missing	14
Underconstrained Axiom	8

An example of a missing simple axiom is that "bishop" is a profession. An example of a missing complex theory is one that assigns a default causality relationship to events that are simultaneous at the granularity reported in the text. An underconstrained axiom is one that allows, for example, "damage to the economy" to be taken as a terrorist incident. Unconstrained identity assumptions result from the knowledge base's inability to rule out identity of two different objects with similar properties, thus leading to incorrect anaphora resolution. "Combinatorics" simply means that the theorem-prover timed out, and the minimal-information proof strategy was invoked to obtain a partial interpretation.

It is difficult to evaluate the precise impact of the robustness strategies outlined here. The robustness is an inherent feature of the overall approach, and we did not have a non-robust control to test it against. However, the implementation of the minimal information proof search strategy virtually eliminated all of our complete failures due to lack of computational resources, and cut the error rate attributable to this cause roughly in half.

6 Conclusion

We felt that the treatment of unknown words was for the most part adequate. The statistical relevance filter was extremely successful. The keyword

antifilter, on the other hand, is apparently far too coarse and needs to be refined or eliminated.

We felt syntactic analysis was a stunning success. At the beginning of this effort, we despaired of being able to handle sentences of the length and complexity of those in the MUC-3 corpus, and indeed many sites abandoned syntactic analysis altogether. Now, however, we feel that the syntactic analysis of material such as this is very nearly a solved problem. The coverage of our grammar, our scheduling parser, and our heuristic of using the best sequence of fragments for failed parses combined to enable us to get a very high proportion of the propositional content out of every sentence. The mistakes that we found in the first 20 messages of TST2 can, for the most part, be attributed to about five or six causes, which could be remedied with a moderate amount of work.

On the other hand, the results for terminal substring parsing, our method for dealing with sentences of more than 60 morphemes, are inconclusive, and we believe this technique could be improved.

In pragmatics, much work remains to be done. A large number of fairly simple axioms need to be written, as well as some more complex axioms. In the course of our preparation for MUC-3, we made sacrifices in robustness for the sake of efficiency, and we would like to re-examine the trade-offs. We would like to push more of the problems of syntactic and lexical ambiguity into the pragmatics component, rather than relying on syntactic heuristics. We would also like to further constrain factoring, which now sometimes results in the incorrect identification of distinct events.

It is often assumed that when natural language processing meets the real world, the ideal of aiming for complete and correct interpretations has to be abandoned. However, our experience with TACITUS, especially in the MUC-3 evaluation, has shown that principled techniques for syntactic and pragmatic analysis can be bolstered with methods for achieving robustness, yielding a system with some utility in the short term and showing promise of more in the long term.

Acknowledgments

This research has been funded by the Defense Advanced Research Projects Agency under Office of Naval Research contracts N00014-85-C-0013 and N00014-90-C-0220.

References

[1] Cardie, Claire and Wendy Lehnert, 1991. "A Cognitively Plausible Approach to Understanding Complex Syntax," *Proceedings*, Ninth National Conference on Artificial Intelligence, pp. 117–124.

[2] Grishman, R., and J. Sterling, 1989. "Preference Semantics for Message Understanding, *Proceedings*, DARPA Speech and Natural-Language Workshop, pp. 71–74.

[3] Hobbs, Jerry R., 1978. "Resolving Pronoun References", *Lingua*, Vol. 44, pp. 311-338. Also in *Readings in Natural Language Processing*, B. Grosz, K. Sparck-Jones, and B. Webber, editors, pp. 339-352, Morgan Kaufmann Publishers, Los Altos, California.

[4] Hobbs, Jerry R., and John Bear, 1990. "Two Principles of Parse Preference", in H. Karlgren, ed., *Proceedings*, Thirteenth International Conference on Computational Linguistics, Helsinki, Finland, Vol. 3, pp. 162-167, August, 1990.

[5] Hobbs, Jerry R., Mark Stickel, Douglas Appelt, and Paul Martin, 1990. "Interpretation as Abduction", SRI International Artificial Intelligence Center Technical Note 499, December 1990.

[6] Jackson, Eric, Douglas Appelt, John Bear, Robert Moore, and Ann Podlozny, 1991. "A Template Matcher for Robust NL Interpretation", *Proceedings*, DARPA Speech and Natural Language Workshop, February 1991, Asilomar, California, pp. 190-194.

[7] Jacobs, Paul S., George R. Krupka, and Lisa F. Rau, 1991. "Lexico-Semantic Pattern Matching as a Companion to Parsing in Text Understanding", *Proceedings*, DARPA Speech and Natural Language Workshop, February 1991, Asilomar, California, pp. 337-341.

[8] Kaplan, Ronald, 1973. "A General Syntactic Processor," in Randall Rustin, (Ed.) *Natural Language Processing*, Algorithmics Press, New York, pp. 193-241.

[9] de Marcken, C.G., 1990. "Parsing the LOB Corpus," *Proceedings,* 28th Annual Meeting of the Association for Computational Linguistics, pp. 243–251.

[10] Moore, R.C., and J. Dowding, 1991. "Efficient Bottom-Up Parsing," *Proceedings*, DARPA Speech and Natural Language Workshop, February 1991, Asilomar, California, pp. 200–203.

[11] Robinson, Jane, 1982. "DIAGRAM: A Grammar for Dialogues", *Communications of the ACM*, Vol. 25, No. 1, pp. 27-47, January 1982.

[12] Sager, Naomi, 1981. *Natural Language Information Processing: A Computer Grammar of English and Its Applications*, Addison-Wesley, Reading, Massachusetts.

[13] Schank, Roger and C. Riesbeck, 1981. *Inside Computer Understanding: Five Programs Plus Miniatures*, Lawrence Erlbaum, Hillsdale, New Jersey.

[14] Stickel, Mark E., 1988. "A Prolog-like Inference System for Computing Minimum-Cost Abductive Explanations in Natural-Language Interpretation", *Proceedings of the International Computer Science Conference-88*, pp. 343–350, Hong Kong, December 1988. Also published as Technical Note 451, Artificial Intelligence Center, SRI International, Menlo Park, California, September 1988.

[15] Sundheim, Beth (editor), 1991. *Proceedings*, Third Message Understanding Conference (MUC-3), San Diego, California, May 1991.

[16] Weischedel, R., D. Ayuso, S. Boisen, R. Ingria, and J. Palmucci, 1991. "Partial Parsing: A Report on Work in Progress, *Proceedings*, DARPA Speech and Natural Language Workshop, February 1991, Asilomar, California, pp. 204–209.

[17] Winograd, Terry, 1983. *Language as a Cognitive Process,* Addison-Wesley, Menlo Park, California.

Appendix

Message 99 of the TST1 corpus:

(1) Police have reported that terrorists tonight bombed the embassies of the PRC and the Soviet Union.
(2) The bombs caused damage but no injuries.

(3) A car-bomb exploded in front of the PRC embassy, which is in the Lima residential district of San Isidro.
(4) Meanwhile, two bombs were thrown at a USSR embassy vehicle that was parked in front of the embassy located in Orrantia district, near San Isidro.

(5) Police said the attacks were carried out almost simultaneously and that the bombs broke windows and destroyed the two vehicles.

(6) No one has claimed responsibility for the attacks so far.
(7) Police sources, however, have said the attacks could have been carried out by the Maoist "Shining Path" group or the Guevarist "Tupac Amaru Revolutionary Movement" (MRTA) group.
(8) The sources also said that the Shining Path has attacked Soviet interests in Peru in the past.

(9) In July 1989 the Shining Path bombed a bus carrying nearly 50 Soviet marines into the port of El Callao.
(10) Fifteen Soviet marines were wounded.

(11) Some 3 years ago two marines died following a Shining Path bombing of a market used by Soviet marines.

(12) In another incident 3 years ago, a Shining Path militant was killed by Soviet embassy guards inside the embassy compound.
(13) The terrorist was carrying dynamite.

(14) The attacks today come after Shining Path attacks during which least 10 buses were burned throughout Lima on 24 Oct.