The TACITUS System: The MUC-3 Experience

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

Artificial Intelligence Center
SRI International

1 Background

SRI International has been engaged in research on text understanding for a number of years. The Naval Ocean Systems Center (NOSC) has sponsored three workshops in recent years for evaluating text understanding systems. SRI participated in the first Message Understanding Conference (MUC-1) in June 1987 as an observer, and subsequently as a participant. Our system was evaluated in the second and third workshops, MUC-2 and MUC-3.

The application domain for MUC-2 (June 1989) was naval operations reports. These were short messages containing much jargon, many misspellings and other mistakes, missing punctuation, and more sentence fragments than grammatical sentences. The task that the systems had to perform was to extract information for database entries saying who did what to whom, when, where, and with what result. The nine participating sites were given a training corpus of 105 messages in early March 1989. They were given 20 new messages in mid-May 1989 to test their system on. Then at the MUC-2 workshop the systems were tested on 5 new messages.

The application domain for MUC-3 was news articles on terrorist activities in Latin America, a sample of which is given in the appendix. The task was similar to that in MUC-2, though somewhat more information had to be extracted. The fifteen participating sites were given a development corpus of 1300 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 system SRI used for these evaluations is called TACITUS (The Abductive Commonsense Inference Text Understanding System). TACITUS is a system for interpreting natural language texts that has been under development since 1985. It has a preprocessor and postprocessor currently tailored to the MUC-3 application. It performs a syntactic analysis of the sentences in the text, using a fairly complete grammar of English, producing a logical form in first-order predicate calculus. Pragmatics problems are solved by abductive inference in a pragmatics, or interpretation, component.

The original purpose of TACITUS was to aid us in investigating the problems of inferencing in natural language. For that reason, the system employed a straight-line modularization, with syntactic analysis performed by the already developed DIALOGIC parser and grammar; only the correct parse was chosen and passed on to the inferencing component.

With the discovery of the abduction framework in 1987 (Hobbs et al., 1990), we realized that the proper way to deal with syntax-pragmatics interactions was in a unified abductive framework. However, the overhead in implementing such an approach at the level of syntactic coverage that the

DIALOGIC system already provided would have been enormous, so that effort was not pursued; we continued to focus on pragmatics problems.

When we began to participate in the MUC-2 and MUC-3 evaluations, we could no longer chose manually which syntactic analysis to process, so we began to invest more effort in the implementation of heuristics for choosing the right parse. We do not view this as the ideal way of handling syntax-pragmatics interactions, but, on the other hand, it has forced us into the development of these heuristics to a point of remarkable success, as the analysis of our results in the latest evaluation demonstrate.

We developed a preprocessor for MUC-2 and modified it for MUC-3. Our relevance filter was developed for MUC-3, as was our current template-generation component.

Those involved in the MUC-3 effort were Douglas Appelt, John Bear, Jerry Hobbs, David Magerman, Ann Podlozny, Mark Stickel, and Mabry Tyson. Others who have been involved the development of TACITUS over the years include Bonnie Lynn Boyd, William Croft, Todd Davies, Douglas Edwards, Kenneth Laws, Paul Martin, and Barney Pell.

2 Overall Results of the MUC-3 Evaluation

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. It is as yet unclear whether high recall, high precision systems will evolve more rapidly from low recall, high precision systems or high recall, low precision systems. It can therefore be argued that the accuracy of TACITUS puts it on the leading edge of the technology.

The significant drop in recall we experienced from Matched Templates to Matched/Missing is an indication that we were failing on messages with a large number of template entries. Much of this is probably due to failures in handling lists of names; NP conjunction rules are highly explosive and given to failure, resulting in the loss of all the template entries corresponding to the names. This problem could be ameliorated by specialized handling of this phenomenon.

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.

While there are a number of parameter settings possible in our system, we decided upon optimal values, and those values were used. An explanation of the parameters and how we decided what was optimal is too detailed and system-particular for this report. None of the decisions was made on the basis of total recall and precision on a test set; all the decisions were made on a much more local basis.

3 The Modules of the System

The system has six modules. As we describe them, their performance on Message 99 of TST1 or on Message 100 of the development set will be described in detail. (Message 99 is given in the Appendix; Message 100 is given in Section 3.4.2.) Then their performance on the first 20 messages of TST2 will be summarized.

3.1 Preprocessor

This component regularizes the expression of certain phenomena, such as dates, times, and punctuation. In addition, it decides what to do with unknown words. There are three choices, and these 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. 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.

3.2 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 them were 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%.

3.3 Syntactic Analysis

The sentences that are declared relevant are parsed and translated into logical form. This is done using 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. For example, for a word like "break", if the usage contains only a subject, it is taken to be the Patient, while if it contains a subject and object, the subject is taken to be ambiguously the Agent or Instrument and the object is taken to be the Patient. Thus, in all three of "The window broke," "John broke the window," and "The hammer broke the window," the window will be taken as the Patient of the breaking. Arguments inherited from control verbs are handled here as well; thus, in "Guerrillas launched an attack" the guerrillas will be taken as the Agent of the attacking as well as of the launching.

Our lexicon includes about 12,000 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.

3.3.1 Performance of the Scheduling Parser and the Grammar

The parser used by the system is a recently developed agenda-based scheduling chart parser with pruning. As nodes and edges are built, they are rated according to syntactic and selectional criteria for how likely they are to figure into 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 complete and incomplete constituents that are allowed to be stored in the chart. Again the algorithm for pruning is simple: Throw away all but the n highest scoring constituents at a given location in the chart, where a location in the chart is determined by a string position and an atomic grammatical category.

The nodes and edges are rated on the basis of their scores from the preference heuristics in DIALOGIC. One reason a correct or nearly correct parse is found so often by this method is that these preference heuristics are so effective; they are described in Hobbs and Bear (1990).

We have experimented with other criteria for rating constituents, such as their length, whether or not they are complete, and the occurrence of function words in them. None of these factors, however, have turned out to improve performance.

Prior to November 1990, we used a simple, exhaustive, bottom-up parser, with the result that sentences of more than 15 or 20 words could not be parsed. The use of the scheduling parser has made it feasible to parse sentences of up to 60 words.

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 TST2, 131 sentences were given to the scheduling parser. A parse was produced for 81 of the 131, 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. 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, 9 were due to problems in lexical entries, 18 were due to shortcomings in the grammar, and 6 were due to garbled text. The causes of 11 failures to parse have not been determined. These errors are spread out evenly across sentence lengths. In addition, 7 sentences of over 30 morphemes hit the time limit we had set, and terminal substring parsing, as described below, was invoked.

The shortcomings in the grammar were the following constructions, which are not currently covered:

which Adverbial VP: "which on 14 December will mark 6 years of subversive activity"

Subordinate-Conjunction Adverbial S: "because on various occasions Aguilar's chief told her about it"

NP and, Adverb, NP: "fellow countrymen whose jobs and, consequently, their right to a livelihood"

Adverb Conjunction Adverb: "sooner or later"

Infinitive Conjunction Infinitive: "to investigate the crime and to send all information from the Fourth Criminal Court of Santiago"

S (containing the word "following") : Conjoined-NPs: "The following people were aboard the Ecuadoran flagship: U.S. citizen Scott Heyndal, Ecuadorans Luis Antonio Meneses Benavides and Edwin Rodrigo Teneda Parreno, and Colombian pilot Julio Torres."

(NP, NP): "the administrative department of security (DAS, secret police)"

as VP: "as has been reported"

be as S/NP: "The situation is not as the Fascist Cristiani reported it to be."

of how S: "of how those of us who have something can show our solidarity with ..."

more Noun to X than to Y: "more harm to Bolivians than to the U.S. embassy"

"no longer"

"the next few days"

cut short NP: "cut short his stay in Japan" ("cut NP short" is handled)

PP is NP: "among the casualties is a lieutenant"

Most of these patterns are very close to patterns that *are* handled by the grammar. Thus the first two patterns in the list would parse without the adverbial, the third would parse without the commas, and the fifth without the second "to".

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 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.

3.3.2 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. We do not attempt to find fragments for strings of less than five words. 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.

Sentence 14 of Message 99 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".

In the first 20 messages of TST2, 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.

For 37% of these 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.

We have noticed one relatively easily correctable problem with the recovery heuristic as it is currently implemented. Right now we first find the longest fragments, then find the highest ranking fragment among those. This sometimes results in a mangled NP interpretation in which the first word is taken to be the head noun or determiner and the remainder of the string is taken to be a relative clause with no relative pronoun and a gap depending on an unusual ditransitive sense of the verb, whereas an excellent interpretation as a sentence could be obtained from all but the first word of that fragment. For example, the fragment

> ... that Casolo used to meet with terrorist leader Fernando ...

was taken to be a headless NP with the determiner "that" and the remainder a relative clause with the object of "use" as the gap. This problem can be corrected by combining length and preference score in a single rating for the fragments.

3.3.3 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.

First "held a press conference yesterday" would be recognized as a VP. The string "the president, VP" would not be recognized as anything, but "the president" would be recognized as an NP. Finally, "George Bush, NP, VP" would be 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, whereas in the former algorithm this is not necessary.

The effect of this technique is to give only short "sentences" to the parser, without losing the possibility of getting a single parse for the entire long sentence. Suppose a 60-word sentence is broken into six 10-word substrings. Then the parsing, instead of taking on the order of 60^3 in time, will only take on the order of $6 * 15^3$. (When parsing the 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 TST2, 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 prior to this no parser in existence could have possibly handled.

3.4 Pragmatics, or Interpretation

3.4.1 Abductive Interpretation

The pragmatics, or interpretation, component employs the general method of abductive explanation to understand texts (Hobbs et al., 1990). The fundamental idea is that the interpretation of a text is the best explanation for what would make it true. This method of explanation is quite well suited to the narrative texts of the MUC-3 domain, because the texts consist almost entirely of declarative sentences that are intended to convey information to the reader. TACITUS does not have an explicit discourse processing module, and does not currently employ any theory of discourse structure, but rather relies on the assumption that the correct resolution of anaphora and individuation of events will be a consequence of generating the best explanation for the truth of its constituent sentences, subject to minimizing the extension of certain predicates. The justification for this assumption is described below.

TACITUS processes each sentence incrementally, seeking the best explanation for why that sentence would be true, given its domain knowledge, and all of the text that it has processed up to the given point in the message. An explanation consists of finding some minimal set of assumptions from which the logical form of the sentence can be derived. The minimality of alternative sets of assumptions is evaluated by adding the assumption cost of each literal assumed. The assumption costs of each literal comprising the logical form is assigned initially in accordance with heuristics reflecting the relative importance of that literal's contribution to the interpretation. Assumption costs can be passed from consequent literals to antecedent literals in Horn clause axioms by means of weighting factors associated with each literal. When the best interpretation of a sentence is found, the set of assumptions is added to the text theory, which then forms part of the base theory for the interpretation of the next sentence in the text. At any time, the contents of the text theory can be examined by the template generation component of the system to generate a set of templates reflecting the text as it has been analyzed up to that point.

Generally, in this domain, the best explanation of the text is one that involves seeing the text as an instance of an "Interesting Act" schema, a schema which includes the principal roles in bombings, kidnappings, and so forth. The explanation of a sentence is identified with an abductive proof of its logical form. This proof may include assumptions of unprovable literals, and each assumption incurs a cost. Different proofs are compared according to the cost of the assumptions they introduce, and the lowest cost proof is taken to be the best explanation, provided that all the assumptions are consistent.

The agents and objects of "Interesting Acts" are required to be "bad guys" and "good guys" respectively. "Bad guys" are terrorists, guerrillas, and their organizations, and good guys are civilians, judges, government officials, etc. Members of the armed forces can be "bad guys" on certain occasions, but they are never "good guys", because as the task was defined, guerrilla attacks on the armed forces are military actions, and hence irrelevant, rather than terrorist acts.

The knowledge base includes a taxonomy of people and objects in the domain. The primary information that is derived from this taxonomy is information about the disjointness of classes of entities. For example, the classes of "good guys" and "bad guys" are disjoint, and any abductive proof that assumes "good guy" and "bad guy" of the same entity is inconsistent. To view an attack by guerrillas on regular army troops as an interesting act would require assuming the victims, i.e. the troops, were "good guys" and since the "good guys" are inconsistent with the military, no consistent explanation of the event in question in terms of "Interesting Act" is possible, and hence no template would be generated for such an incident.

In addition to proving and assuming literals, an important part of the abduction proof involves minimizing the extension of certain predicates through factoring. If $\exists x P(x)$ is an assumption, and $\exists y P(y)$ is a goal, then it is possible to *factor* the literals through unification, setting the resulting assumption cost to the minimum assumption cost of the two literals. This factoring operation entails the assumption that $x = y$, which amounts to assuming that individuals that share property P are identical. This factoring mechanism is the primary mechanism by which anaphora is resolved. Pronominal anaphora works differently, in that the structure of the text is taken into account in creating an ordered list of possible antecedents. The abductive reasoner will resolve the pronoun with the first object on the antecedent list that leads to a consistent proof.

Only a subset of the predicates in a domain should be minimized. For example, causation is a relation that holds among many events. Simply knowing that event e_1 causes e_2 and e_3 causes e_4 is rather poor grounds for assuming that e_1 and e_3 are the same and that e_2 and e_4 are the same. Apparently, causation is not one of the predicates that should be factored. However, predicates corresponding to natural kinds probably refer to specific entities, and are good candidates for minimization. Similarly, predicates relating event types to tokens should be minimized. If an article mentions a kidnapping twice, then it is often reasonable to assume that the same event is being described.

Clearly, assumption of a goal literal is not a sound operation, because the assumption might be inconsistent with the base theory. This means that every set of assumptions must be checked for internal consistency and consistency with the base theory. Moreover, using the factoring mechanism for anaphora resolution requires one to have a rich enough domain theory so that incorrect resolutions can be eliminated from consideration, because otherwise the system is strongly biased toward collapsing everything into a single individual or event. In general, consistency checking is a computationally expensive process, in the cases that it is even decidable. TACITUS therefore uses a restricted theory to check consistency of a set of assumptions so that the consistency check can be computed with comparatively little effort. Any assumption set is rejected as inconsistent if it meets any of the following criteria:

- $P(a)$ and $Q(a)$ are both assumptions, and a class hierarchy indicates that P and Q have disjoint extensions.
- $P(a)$ and $Q(a)$ are both assumptions, and P and Q are distinct predicates corresponding to proper names.
- Sets s_1 and s_2 are identified through factoring and have different cardinality.

This basic assumption and consistency checking mechanism drives the discourse processing in the current TACITUS system.

It may not be obvious that minimization of events and individuals of a given natural kind should lead to a correct interpretation of the text. After all, there is no a priori justification in the world for assuming that two individuals of a given type are the same. Strictly on the basis of probability, it is in fact highly unlikely. The minimization heuristic relies on the fact that one is interpreting a coherent text that conforms to the Gricean maxim of relevance. By assuming that a text is coherent, one can assume that the events and individuals mentioned are related in some systematic way. The abductive interpretation of the text makes these relations explicit as part of the process of explaining the truth of the sentences. Minimization of selected relations is one way of maximizing the connections of each sentence with the text that preceeds it.

The domain knowledge base is divided into a set of axioms, which are used for abductively proving the sentences from the text, and a class hierarchy, which is used for checking the consistency of the proofs. The axioms are divided into a core set of axioms describing the events in the domain that correspond to the incident types, and lexical axioms, which are meaning postulates that relate the predicate introduced by a lexical item to the core concepts of the domain.

The knowledge base includes approximately 550 axioms at the current stage of development. This breaks down into about 60 axioms expressing the core facts about the schemas of interest, 430 axioms relating lexical entries to these core schemas, and approximately 60 axioms for resolving compound nominals, of-relations, and possessives. The knowledge base also includes approximately 1100 locations, for which relevant axioms are introduced automatically at run-time.

3.4.2 An Example of Interpretation

To illustrate the basic principles of interpretation in TACITUS we refer to Message 100 from the MUC-3 development corpus:

```
   LIMA, 30 MAR 89 -- [TEXT] 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.
```

Interpreting the first sentence of this text requires making certain assumptions based on general world knowledge. For example, the knowledge base expresses the fact that dynamite is an explosive substance, a "substance object" compound nominal refers to an object that is composed of the substance, an object that is composed of an explosive substance is a bomb, and hitting a bomb results in an explosion, and explosions cause damage, and derailing is damage. By minimizing the extension of the *damage* predicate, we conclude that the hitting of the dynamite charge caused an explosion (which is a terrorist bombing incident), and the bombing caused the train to derail.

The next sentence mentions a death in an explosion. Correct interpretation of this sentence requires association between the explicitly mentioned explosion and the explosion resulting from hitting the dynamite charge. Otherwise the system may conclude that two bombing events were involved. Minimization of exploding events results in the correct resolution.

Although this text does not contain a pronoun, the resolution of pronoun references is an important aspect of interpretaton. The descriptive content of pronouns is very limited, since it includes only number, gender, and animateness information. In general this descriptive content is insufficient alone to facilitate identification of the antecedent. In addition, syntactic relationships can rule out certain coreferentiality possibilities that would be consistent with the descriptive content of the pronoun, and make others more likely. Therefore, TACITUS uses a different method for resolving pronominal references than for NPs with noun heads.

Hobbs (1978) describes an algorithm for pronominal anaphora resolution based only on criteria of syntactic structure and matching of basic selectional constraints. A statistical study by Hobbs demonstrated that this algorithm correctly identifies pronominal antecedents 91.7% of the time in the texts he studied. TACITUS employs this algorithm to produce an ordered disjunction of coreference possibilities as part of the logical form. During abductive interpretation, the variable representing the referent of the prounoun is bound to the the first alternative on this list, and if this binding passes the consistency check, it is assumed to be the correct resolution. If not, successive bindings are chosen from progressively less likely alternatives as determined by the resolution algorithm, until a consistent interpretation is found. This resolution method thus allows the syntactic algorithm to be improved by the incorporation of pragmatic information, although no evaluation has yet been undertaken to quantify the success of this approach.

3.4.3 Problems for Future Research

A serious problem with this general approach to discourse processing is the combinatorics of the problem of minimizing the predicates while at the same time searching for the cheapest abductive proof. Each time the theorem prover attempts to prove a goal, it has three choices: it can prove the goal, assume it, or factor it with something it already assumed. It is easy to see that each choice point increases the size of the search space exponentially.

One approach to dealing with this combinatorial problem is to limit the choices by processing only sentences that pass a statistical relevance filter. Another strategy along these lines is careful selection of the predicates to be minimized. If predicates are related by a chain of entailments, only the most general predicates in the chain should be considered for minimization.

Another problem is that our approach requires a relatively rich knowledge base for consistency checking. When information outside the scope of the knowledge base is encountered, the minimization strategy is generally too agressive. The absence of information allows it to assume that almost anything is the same as anything else without contradiction. Knowledge base construction is, of course, part of the long-term effort we are addressing.

Finally, methods must be found for expanding the consistency checking to include various temporal and locative inconsistencies, as well as some other problems. For example, current consistency checking methods have trouble dealing with singular and plural entities properly, as well as collective anaphora.

One type of discourse problem that our current approach cannot handle is the resolution of anaphora depending on syntactic parallelism, because all information about parallel structure of phrases is lost by the time the abductive reasoning process operates. However, the actual number of texts in which this consideration is crucial for performing the MUC-3 task seems to be quite small, and therefore the shortcoming is not a severe handicap for this task.

Our current experience with the TACITUS system suggests that this simple but powerful method of abductive interpretation can be quite successful at handling many of the discourse problems that arise in this task.

In many cases, anaphora are correctly resolved, and correct causal relationships between actions are postulated. Although the system in its current state of implementation still makes mistakes, these mistakes can often be traced to inadequacies in the knowledge base. While the ultimate success of the general approach is still an open question, current experience suggests that there is still much room for improvement before inherent limitations are reached.

3.5 Template Generation

The task of the template generation component is to take the results of the abductive proofs in pragmatics and put them into a specified template form. For the most part, this is a matter of reading the information directly off the results produced by the pragmatics component. There are complications, however.

In general, the system generates one template for every interesting act that is assumed by pragmatics. But there are several exceptions. An interesting act can be both an ATTACK and a MURDER, and only the MURDER template would be produced. An interesting act of type MURDER might be divided into two templates, if it was found that some of the victims survived the attack. For example, the text

> A Colombian vessel fired automatic weapons at Luis Meneses Benavides and Julio Torres.
> Torres was killed and Meneses was wounded.

would result in one MURDER template and one ATTEMPTED MURDER template.

TACITUS does not employ any means of individuating events other than the general heuristic of finding a consistent interpretation with the minimal number of events of each type. There are a number of problems posed by the MUC-3 domain that require some extensions to this basic minimization strategy. In interpreting the final sentence of Message 100, TACITUS relies on its knowledge that any type of event can be described as an incident. Minimization of events can be done by resolving the "incident" to either the implicit explosion, the death of Flores, or the derailing, and thus associating the locative and temporal information contained in this sentence with one of the events we already know about. Since all of these events are essentially concurrent, the template generation process can correctly fill the template, no matter which event is chosen as the resolution of "incident."

For each interesting act, a cluster of contemporaneous and causally related events from the text is formulated. Any temporal or locative information that is associated with any of these events, or the agents and objects participating in the events, is used to fill the DATE and LOCATION slots of the respective templates. Each slot is then filled by looking at the arguments of the relevant predicates; if any of these arguments represent sets, the sets are expanded into their constituents for the slot fills.

Events in the MUC-3 domain have multiple agents, objects, and instruments. Therefore, two events of the same type with different agents and objects can consistently be collapsed into a single event with multiple agents and objects. The reliable individuating criteria are time and location. However, the temporal reasoning necessary to determine accurately whether two intervals or locations are different can be quite complex, and do not fit within the class hierarchy or simple consistency checking mechanisms that are employed by the abductive theorem prover. Therefore, these sorts of inconsistencies are not detected during pragmatics processing and must be left for the final template filling phase. At this stage of development, there is no opportunity to backtrack to earlier stages of processing if the consistency checking mechanism is not powerful enough to detect the relevant inconsistencies. The incorporation of locative and temporal consistency checking into the abductive proof process is a current topic of investigation.

For string fills, proper names are preferred, if any are known, and if not, the longest description from all the coreferential variables denoting that entity is used, excluding certain uninformative descriptors such as "casualties."

In a final pass, analysis eliminates from consideration templates that do not pass certain coherence or relevance filters. For example, any template that has a "bad guy" as the object of an attack is rejected, since this is probably a result of an error in solving some pragmatics problem. Templates for events that take place in the distant past are rejected, as well as events that take place repeatedly or over vague time spans, such as "in the last three weeks". Finally, templates for events that take place in irrelevant countries are eliminated. This final filter, unfortunately, can entirely eliminate otherwise correct templates for which the location of the incident is incorrectly identified. This was responsible for several costly mistakes made by TACITUS in the MUC-3 evaluation.

4 Causes of Failures

It is difficult to evaluate the interpretation and template generation components individually. However, we have examined the first twenty messages of TST2 in detail and attempted to pinpoint the reason for each missing or incorrect entry in a template.

There were 269 such mistakes, due to problems in 41 sentences. We have classified them into a number of categories, and the results for the principal causes are as follows.

Reason	Mistakes	Sentences
Simple Axiom Missing	49	9
Unknown Words	38	3
Combinatorics	28	3
Parsing Problems	26	5
Unconstrained Factoring	25	3
Lexicon Error	24	2
Syntax-Pragmatics Mismatch in Logical Form	22	5
Complex Axioms or Theory Missing	14	5
Relevance Filter Missed Sentence	11	2
Underconstrained Axiom	8	3

An example of a missing simple axiom is that "bishop" is a profession. An example of a missing complex axiom or theory is whatever it is that one must know to infer the perpetrator from the fact that a flag of a terrorist organization was left at the site of a bombing. An underconstrained axiom is one that allows, for example, "damage to the economy" to be taken as a terrorist incident. Unconstrained factoring is described above in Section 3.4.1. An example of a lexicon error would be a possibly intransitive verb that was not correctly specified as intransitive. The syntax-pragmatics mismatches in logical form were representation decisions (generally recent) that did not get reflected in either the syntax or pragmatics components. "Combinatorics" simply means that the theorem-prover timed out; that this number was so low was a pleasant surprise for us.

In these results two incorrect lexical entries and problems in handling three unknown words were responsible for 23% of the mistakes. This illustrates the discontinuous nature of the mapping from processing to evaluation. A difference of δ in how a text is processed can result in a difference of considerably more than ϵ in score. The lesson is that the scores cannot be used by themselves to evaluate a system. One must analyze its performance at a deeper, more detailed level, as we have tried to do here.

5 Summary: What Was and Was Not Successful

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-2 and MUC-3, we have 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.

In template-generation, we feel our basic framework is adequate, but a great many details must be added.

The module we would most like to rewrite is in fact not now a module but should be. It consists of the various treatments of subcategorization, selectional constraints, generation of canonical predicate-argument relations, and the sort hierarchy in pragmatics. At the present time, due to various historical accidents and compromises, these are all effectively separate. The new module would give a unified treatment to this whole set of phenomena.

6 Usability for Other Applications

In the preprocessor, the spelling corrector and the morphological word assignment component are usable in other applications without change.

The methods used in the relevance filter are usable in other applications, but, of course, the particular statistical model and set of keywords are not.

In the syntactic analysis component, the grammar and parsing programs and the vast majority of the core lexicon are usable without change in another application. Only about five or six grammar rules are particular to this domain, encoding the structure of the heading, interview conventions, "[words indistinct]", and so on. The logical form produced is application-independent.

The theorem prover on which the pragmatics component is based is application-independent. All of the enhancements we have made in our MUC-3 effort would have benefited our MUC-2 effort as well.

In the knowledge base, only about 20 core axioms carried over from the opreps domain to the terrorist domain. Since most of the current set of axioms is geared toward the particular MUC-3 task, there would very probably not be much more of a carry-over to a new domain.

The extent to which the template-generation component would carry over to a new application depends on the extent to which the same baroque requirements are imposed on the output.

7 Preparing for the Evaluation

7.1 Level of Effort

Between the preliminary MUC-3 workshop in February and the final evaluation in May, approximately 800 person-hours were spent on the project. This breaks down into subtasks approximately as follows.

Preprocessor, system development, testing:	180 hours
Development of parsing algorithms:	180 hours
Grammar development:	220 hours
Pragmatics and template-generation:	220 hours

7.2 The Limiting Factor

Time.

7.3 Training

The amount of the training corpus that was used varied with the component. For the relevance filter, all 1400 available messages were used. For the lexicon, every word in the first 600 and last 200 messages and in the TST1 corpus were entered. For the remaining messages, those words occurring more than once and all non-nouns were entered.

For syntax and pragmatics, we were able only to focus on the first 100 messages in the development corpus.

Tests were run almost entirely on the first 100 messages because those were the only ones for which a reliable key existed and because concentrating on those would give us a stable measure of progress.

The system improved over time. On the February TST1 run, our recall was 14% and our precision was 68% on Matched and Missing Templates. At the end of March, on the first 100 messages in the development set, our recall was 22% and our precision was 63%. At the time of the TST2 evaluation, on the first 100 messages in the development set, our recall was 37% and our precision was 64%. We have thus been able to improve recall with no significant sacrifice in precision.

8 What Was Learned About Evaluation

On the one hand, the mapping from texts to templates is discontinuous in the extreme. One mishandled semicolon can cost 4% in recall in the overall score, for example. Therefore, the numerical results of this evaluation must be taken with a grain of salt. Things can be learned about the various systems only by a deeper analysis of their performance. On the other hand, the task is difficult enough to provide a real challenge, so that pushing recall and precision both into the 70s or 80s will require the system to do virtually everything right.

Leading up to MUC-3 there were a great many difficulties to be worked out, diverting the attention of researchers from research to the mechanics of evaluation. It is to be hoped that most of these problems have been settled and that for MUC-4 they will constitute less of a drain on researchers' time.

We feel the task of the MUC-3 evaluation—automatic database entry from complex text—is both challenging and feasible in the relatively short term.

Acknowledgements

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] 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.

[2] 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.

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

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

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

Appendix

TST1-MUC3-0099

LIMA, 25 OCT 89 (EFE) -- [TEXT] POLICE HAVE REPORTED THAT TERRORISTS TONIGHT BOMBED THE EMBASSIES OF THE PRC AND THE SOVIET UNION. THE BOMBS CAUSED DAMAGE BUT NO INJURIES.

A CAR-BOMB EXPLODED IN FRONT OF THE PRC EMBASSY, WHICH IS IN THE LIMA RESIDENTIAL DISTRICT OF SAN ISIDRO. 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.

POLICE SAID THE ATTACKS WERE CARRIED OUT ALMOST SIMULTANEOUSLY AND THAT THE BOMBS BROKE WINDOWS AND DESTROYED THE TWO VEHICLES.

NO ONE HAS CLAIMED RESPONSIBILITY FOR THE ATTACKS SO FAR. 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. THE SOURCES ALSO SAID THAT THE SHINING PATH HAS ATTACKED SOVIET INTERESTS IN PERU IN THE PAST.

IN JULY 1989 THE SHINING PATH BOMBED A BUS CARRYING NEARLY 50 SOVIET MARINES INTO THE PORT OF EL CALLAO. FIFTEEN SOVIET MARINES WERE WOUNDED.

SOME 3 YEARS AGO TWO MARINES DIED FOLLOWING A SHINING PATH BOMBING OF A MARKET USED BY SOVIET MARINES.

IN ANOTHER INCIDENT 3 YEARS AGO, A SHINING PATH MILITANT WAS KILLED BY SOVIET EMBASSY GUARDS INSIDE THE EMBASSY COMPOUND. THE TERRORIST WAS CARRYING DYNAMITE.

THE ATTACKS TODAY COME AFTER SHINING PATH ATTACKS DURING WHICH LEAST 10 BUSES WERE BURNED THROUGHOUT LIMA ON 24 OCT.