[Proceedings of the *Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*, Colgate University, Hamilton, New York, August 14-16, 1985]

INTEGRATING SYNTAX AND SEMANTICS[1]

Steven L. Lytinen
Cognitive Systems, Inc.
234 Church St.
New Haven, CT 06510, USA

Abstract

Well-known examples such as Bar-Hillel's (1960) "The box is in the pen" illustrate that extensive semantic analysis is necessary to resolve ambiguities that must be resolved in machine translation. If one accepts the premise that semantics should be added to the analysis techniques used in machine translation, what is the way in which it should be added? This paper will argue for an *integrated* approach to semantic processing. That is, syntactic and semantic processing should take place at the same time, rather than in separate stages. However, although I will argue for the integration of syntactic and semantic analysis *processes,* I will also argue for the use of a separate body of syntactic *knowledge,* and for building a separate syntactic *representation* during the parsing process. This is in contrast to previous integrated parsers, which have relied almost exclusively on semantic representations to guide the parsing process, and which have not used a separate body of syntactic rules.

[1] This research was conducted at the Artificial Intelligence laboratory of Yale University, and was supported in part by the Advanced Research Projects Agency of the Department of Defense and monitored by the Office of Naval Research under contract No. N00014-82K-0149.

1 Introduction

It has long been realized by machine translation researchers that semantics[2] must be used to resolve some of the lexical and structural ambiguities that occur in natural language. Bar-Hillel's "box is in the pen" example (Bar-Hillel, 1960) illustrated that even simple English sentences could contain ambiguities that would require extensive semantic analysis to resolve. Interest has risen in recent yean to add semantic analysis to machine translation. Efforts have ranged from adding domain-specific semantic features to syntactic analysis, such as in METEO-TAUM (Chandioux, 1976), to analyzing the syntactic parse tree for logical relations between constituents, as in ARIANE (Boitet and Nedobejkine, 1981), to a full-blown semantic analysis of the input text such as Wilks' system (Wilks, 1973).

If one accepts the premise that semantics should be added to the analysis techniques used in machine translation, what is the way in which it should be added? This paper will argue for an *integrated* approach to semantic processing. By that, I mean that syntactic and semantic processing should take place at the same time, rather than in separate stages. I will argue that if syntactic and semantic processing are performed separately, with the results of a syntactic parse passed to a semantic interpreter, this must result in the inability to resolve many ambiguities during the syntactic analysis stage, thus dramatically increasing the number of syntactic interpretations that must be considered during the parse.

An integrated approach to parsing has been argued for before (e.g., (Riesbeck and Schank, 1976), (Schank and Birnbaum, 1980)). However, previous integrated parsers, such as ELI (Riesbeck and Schank, 1976) and Wilks' parser (Wilks, 1975) have been integrated in *representation,* as well as processing. That is, these parsers have not maintained separate syntactic representations of the input text apart from the text's semantic representation. This, I will argue, also leads to trouble, making it difficult to resolve syntactic ambiguities without requiring an inordinately large number of parsing rules.

To remedy the difficulties of syntax-first parsing and of previous integrated parsers, I will present an alternative approach to integrated parsing. This approach is implemented in a machine translation system called MOPTRANS, which parses short (1-3 sentences) newspaper stories about terrorism and crime, in English, Spanish, French, German, and Chinese. Translations are produced for these stories in English and/or German. Enough vocabulary, linguistic knowledge, and semantic knowledge have been encoded in the parser to enable it to parse 25-50 stories for each input language. The MOPTRANS system produces translations for all of the stories into English, and for some of the stories into German.

The MOPTRANS parser is integrated in the sense that syntactic and semantic processing take place at the same time. However, MOPTRANS does maintain a separate syntactic representation of the input text during parsing, and it uses a largely autonomous set of syntactic rules. Unlike syntax-first parsers, however, these syntactic rules are driven by the system's semantic analyzer. Thus, syntactic attachments are only considered when the semantics of the system judges the potential attachment to be semantically meaningful.

[2] By semantics, I mean the traditional linguistic concept of semantics, or knowledge about the meanings of words, as well as *pragmatics,* or knowledge about the world and about how language is used.

This paper will not include a discussion of MOPTRANS' semantic analyzer. For a detailed description, see (Lytinen, 1984). Instead, this paper will focus on the way in which the semantics of the system is integrated with syntactic processing, and why this integration is desirable.

2 Why Syntax Needs Semantics

Consider the following sentences:

The cleaners dry-cleaned the coat that Mary found at the rummage sale for $10.

The cleaners dry-cleaned the coat that Mary found in the garbage for $10.

On the basis of syntactic information alone, these two sentences are both ambiguous. In both sentences, the prepositional phrase "for $10" could be attached to either verb, or to the last NP ("the rummage sale" or "the garbage"). However, due to the semantics of the sentences, their syntactic structures are not the same. In the first example, the preferred attachment of "for $10" is to the verb "found," although it could be argued that the sentence is still ambiguous, even after semantic interpretation. However, in the second example, the attachment of "for $10" is definitely to the verb "dry-cleaned."

Examples like these illustrate that semantics must be used in order to resolve some syntactic ambiguities. It is very difficult to write syntactic rules to resolve the ambiguity in these examples. Obviously the syntactic structure of these sentences gives no clue, since the structure up to the point of the ambiguities is the same. There is no way that selectional restriction rules could help, either, since the same verbs, "dry-cleaned" and "found," appear in both sentences. The ambiguity can be resolved only after inferring the different meanings of the word "found" in these two sentences. In the first sentence, since finding an article at a rummage sale usually entails purchasing the article, "for $10" can, and does, attach itself to this meaning of "found," However, in the second sentence, since finding an article in the garbage does not involve any sort of purchase, "for $10" cannot be attached to "found." Thus we see that the syntactic ambiguities in these two sentences cannot be resolved until a semantic analysis is performed on them which enables the inference to be made that "found" in the first sentence refers to a purchase.

If one accepts the premise that all syntactic ambiguities should be resolved during syntactic analysis, then examples like these argue for the integration of syntactic and semantic processing, since it is not possible to resolve these ambiguities if semantic analysis is performed afterwards. However, the argument could be made that syntactic analysis need not resolve all syntactic ambiguities. Unresolvable ambiguities could be passed on to the semantic analysis stage, where they would be resolved after semantic analysis provided the necessary information.

If syntactic ambiguities are not resolved immediately, though, there is a computational price to pay. This is because sometimes an unresolved syntactic ambiguity can affect the remainder of the syntactic analysis. Consider the following sentence:

The cleaners dry-cleaned the coat that Mary found in the garbage for $10 while she was away in New York.

If semantics is used immediately to resolve the attachment of "for $10" to the verb "dry-cleaned," then the clause "while she was away in New York" must also be attached to "dry-cleaned," since the clause beginning with "found" is already closed by the attachment of "for $100" to a constituent prior to "found". However, if the attachment of "for $10" is not resolved immediately, then the syntactic analyzer must consider other attachments of this clause. Since "for $10" could possibly attach to "found," this means that the clause could also be attached there. This "artificial" ambiguity cannot be resolved syntactically, either. Thus, a syntactic parser would find this sentence to be 3-way ambiguous. The additional ambiguities could be completely avoided if semantics were used immediately to resolve the original ambiguity.

Carrying forward ambiguities in syntactic analysis that could be resolved in an integrated parser can cause a combinatorial explosion in the number of syntactic ambiguities that must be considered as the parse continues. For example, consider the following sentence:

> The stock cars raced by the spectators crowded into the stands at over 200 mph on the track at Indy.

This sentence is highly ambiguous syntactically, but if semantic processing proceeds in parallel with syntactic processing, many of the ambiguities can be resolved along the way, reducing drastically the number of possible interpretations that must be considered. Let us compare the complexity of a left-to-right parse of this sentence, with and without the use of semantics to resolve ambiguities. The verbs "raced" and "crowded" could both be either past active or past participle. Syntactically, these ambiguities cannot be resolved. Thus, the part of the sentence up to "into the stands" has 4 possible syntactic interpretations: one in which "raced" is past active, one in which "crowded" is past active, one in which both "raced" and "crowded" begin reduced relative clauses which modify "cars," and one in which the second relative clause modifies "spectators" instead. However, semantics can provide the information that "spectators" are not likely to race stock cars. Therefore, semantic information can determine that "raced" must be a past active verb. This determination eliminates all but one of the 4 interpretations.

As the parse continues, "at 200 mph" could be attached in many ways without considering semantics: to "cars," "raced," "spectators," "crowded," or "stands." Because of the combination of the possible PP attachments and the possible interpretations of "raced" and "crowded" as active or passive, a syntax-first parser would be faced with 13 possible parses of the sentence up to this point.[3] However, with semantics, there are only three possible attachments: to "raced," "crowded," or "stands." Since semantics could supply the additional information that spectators cannot be on the track in a race, two of these choices can be eliminated, leaving "raced" as the only possible attachment.

The number of possible attachments for the next two PP's continues to grow combinatorially without the use of semantics. These PP's could conceivably be attached to all 5 of the constituents to which the previous PP could attach, as well as to "200 mph." Given 6 different possible attachments, 13 possible parses thus far, and 2 prepositional phrases, we have 156 (6 X 13 X 2) potential interpretations to consider. Selectional restrictions could probably eliminate some of these attachments, and some attachments are not possible in some of the 13 interpretations, due to the closure of some constituents by

---

[3] There would be 20 possible parses, due to the 4 possible interpretations from before, multiplied by the 5 possible attachments of the prepositional phrase, but 7 of these parses are not possible due to the fact that some constituents are closed by previous attachments. For example, if "crowded" is interpreted to be active, "on the track" cannot be attached to anything before "crowded."

previous attachments, but we can see that the number of combinations becomes quite large. However, since semantics would have been able to eliminate all but one of the 13 interpretations thus far, the last 2 PP's could only conceivably be attached in two places, "raced" or "200 mph." This is because the attachment of "on the track" to "raced" eliminated the possibilities of attachment anywhere else.

As this example demonstrates, the price for separating syntactic and semantic processing can be quite expensive computationally. Unresolved syntactic ambiguities can build on each other, resulting in the need to consider many syntactic attachments which would be eliminated if semantic processing were done in parallel. Therefore, it seems that semantic and syntactic processing should indeed be integrated, to control the combinatorial explosion that can take place in syntax-first parsing.

3 Why Semantics Needs Syntax

Syntactic and semantic processing have been integrated in many previous parsers; for example, ELI (Riesbeck and Schank, 1976), and Wilks' parser (Wilks, 1975)). However, in these previous parsers, the assumption has been made that a full-blown, separate syntactic analysis is not needed in order to build a semantic representation of text. Instead, many past conceptual analyzers have relied on "local" syntactic checks for the syntactic information needed.

To explain what I mean by local syntactic checks, let us consider some of the syntactic rules which were used in the Conceptual Analyzer (CA) (Birnbaum and Selfridge, 1979), a descendant of ELI. CA's parsing rules were encoded in the form of *requests,* which were test-action pairs stored mainly in the parser's lexicon. A request could be in one of two states: active or inactive. A request was activated when the parser encountered a word whose dictionary entry contained that request. Once active, a request stayed active until it *fired,* or was executed; or until it was explicitly deactivated by another request. A request fired if it was in the active state and the conditions of active memory satisfied the test portion of the request's test-action pair.

Requests were responsible for making most of the decisions that took place during parsing, including the resolution of syntactic ambiguities. For example, consider the following sentence, which was parsed by CA:

A small plane stuffed with 1500 pounds of marijuana crashed.

The word "stuffed" can function as either a past participle or a past active verb. In this context, it functions as a past participle, signaling the beginning of a reduced relative clause.

To resolve this ambiguity, CA used 3 requests. One looked for the presence of a form of "to be" to the left of "stuffed." If it was found, then "stuffed" was passive, and a representation was built of "stuffed" with the NP to the left of "stuffed" (in this case "plane") assigned to be the OBJECT being stuffed. A second request looked for the word "with" appearing after "stuffed." If it was found, "stuffed" was again treated as passive, and again the NP to the left was the OBJECT being stuffed. This request, if it fired, also activated another request which looked for another verb further on in the sentence, marking the end of the relative clause. Finally, the third request looked for something which was of the semantic class CONTAINER to the right of the word "stuffed." If this request fired, the CONTAINER was the OBJECT being "stuffed," and the NP to the left of "stuffed" was the ACTOR. This request fired for sentences like "John stuffed the plane with marijuana."

These 3 requests used "local" syntactic information in order to disambiguate the word "stuffed." By this, I mean that only words in the immediate neighborhood of "stuffed" were checked for particular syntactic properties, or for their presence or absence. If a form of "to be" appeared directly before "stuffed," then "stuffed" was assumed to be passive, but not part of a relative clause. If the preposition "with" appeared directly after "stuffed," then "stuffed" was part of an unmarked relative clause. If a noun group followed "stuffed" which could function as its direct object, then "stuffed" was a past active verb.

The advantage of using only local syntactic checks in requests was that it was not necessary for the parser to keep track of a separate syntactic analysis. Syntactic ambiguities were resolved by examining short-term memory to see what *semantic* constituents had been built, and by examining the order in which these semantic constituents had appeared in the sentence. However, it is not always the case that local checks are enough. Consider the following examples:

The soldier called to his sergeant.
I saw the soldier called to his sergeant.

The slave boy traded for a sack of grain.
I saw the slave boy traded for a sack of grain.

In these cases, the appearance of a preposition after the verbs "called" and "traded" does not guarantee that the verbs are passive. This is because both verbs can be used either transitively or intransitively. Instead, the information that must be used to determine whether the verbs are active or passive is whether or not there is another verb in the sentence which functions as the main verb.

The requests needed to handle these examples would be more complex. First, a request would be required which looked to the left to see if another verb was already on the active list. If so, then "called" would have to be unmarked passive. But the absence of a verb would not guarantee that "called" was active, since the main verb of the sentence could also come after "called," as in the following example:

The soldier called to his sergeant was reprimanded.

Two requests would be required, one looking back for the main verb of the sentence, and one looking forward for the main verb. These two requests would be in addition to the requests that were used for "stuffed."

Even with these additional requests, however, many sentences could still not be handled:

The soldier called to the sergeant shot in the arm.
The soldier called to the sergeant shot three enemy troops.

In these examples, the verb which appears later in the sentence is also syntactically ambiguous. So the appearance of a verb after "called" does not always guarantee that "called" is a past participle.

To handle examples like these, the requests above would have to be made still more complicated. An additional request under "called" would have to look for a verb which could either be past active or past participle. If such a verb was found, then special requests would have to be activated which would look for the appropriate clues around the second verb to determine whe ther it was active or passive, thus also determining if the first verb was active or passive. In short, the number of requests which would be required and the complexity of these requests would become very great.

In general, then, it appears that some syntactic ambiguities cannot be resolved without great difficulty by local syntactic checks. This is because the resolution of syntactic ambiguities sometimes requires more global knowledge about the syntax of a sentence, such as whether a particular verb functions as the main clause verb. Information like this cannot be determined so easily by rules which examine only immediate context. Thus, although we would like for syntactic and semantic processing to be integrated, as it was in ELI and CA. it seems that a separate syntactic representation must still be built during the analysis process in order to resolve these ambiguities.

4 A Parser Which Satisfies Both Constraints

I shall now describe the MOPTRANS parser, and demonstrate how it overcomes the difficulties that I have outlined in the last two sections. The MOPTRANS parser is an integrated parser, in the sense that syntactic and semantic processing take place in tandem. However, it is different from previous integrated parsers, in that it uses a largely autonomous set of syntactic rules, and a syntactic representation of the input text is built during parsing. MOPTRANS uses PARSIFAL-like parsing rules (Marcus, 1978), which specify how sequences of syntactic constituents in the input text can be attached to each other. Also like Marcus' parser, the MOPTRANS parser does not always account for every new constituent immediately, as is the case in an ATN parser (Woods, 1970). If no syntactic pattern is matched by the input, the parser continues reading on until a rule does match.

Unlike PARSIFAL and other syntactic parsers, syntax rules in MOPTRANS are only considered and applied if the syntactic attachments that they make are judged by the parser's semantic analyzer to be semantically appropriate. In this way, syntactic and semantic processing are completely integrated. To make this more clear, let us consider a simple example, and how it would be parsed by the MOPTRANS parser

John gave Mary a book.

As with previous integrated parsers, MOPTRANS' dictionary definitions contain information about what semantic representation the parser should build when it encounters a particular word. Thus, "John" causes the representation PERSON to appear in the parser's active memory. At the same time, since "John" is a proper noun, the syntactic class NP is also activated.

MOPTRANS' definition of the word "gave" builds the Conceptual Dependency representation (Schank, 1972) ATRANS (transfer of possession or control). At this point, MOPTRANS considers the two semantic representations in active memory, PERSON and ATRANS. The semantic analyzer tries to combine these representations in whatever way it can. It concludes that the PERSON could be either the ACTOR or the RECIPIENT of the ATRANS, since the constraints on these roles are that they must be ANIMATE. It also concludes that the PERSON could be the OBJECT of the ATRANS (that is, the thing whose control or possession is being transferred). However, since this role is expected to be a PHYSICAL-OBJECT rather than an ANIMATE, the match is not as good as with the

ACTOR or RECIPIENT roles.[4]

This is the point at which the MOPTRANS parser utilizes its syntactic rules. Semantics has determined that 2 possible attachments are preferred. Now the parser examines its syntactic rules to see if any of them could yield either of these attachments. Indeed, the parser's Subject Rule will assign the PERSON to be the ACTOR of the ATRANS. The Subject Rule looks like this:

```
Subject Rule

Syntactic pattern:       NP, V (active)
Additional restrictions: NP is not already attached syntactically
Syntactic assignment:    NP is SUBJECT of V.  V  is indicative (V-IND)
Semantic action:         NP is ACTOR of V (or another slot, if specified
                           by V)
Result:                  V-IND
```

This rule applies when an NP is followed by a V, and when the NP can fill the ACTOR slot of the semantic representation of the V. The NP is marked as the SUBJECT of the V, and the V is marked as indicative (V-IND). As dictated by the RESULT of the rule, the V-IND is left in active memory, but the NP is removed, since its role as subject prevents subsequent attachments to it such as PP attachments. At the same time as these syntactic assignments, the semantic representation of the NP "John" is placed in the ACTOR slot of the ATRANS representing the verb.

The rest of the sentence is parsed in a similar fashion. To determine how "Mary" should be attached to "gave," semantics is asked for its preference. It determines that the RECIPIENT slot of the ATRANS is the best attachment.[5] Syntax is consulted to see if any syntactic rules could make this attachment. This time, the Dative Movement rule is found:

```
Dative Movement Rule

Syntactic pattern:       V-IND, NP
Additional restrictions: V-IND allows dative movement
Syntactic assignment:    NP is (syntactic) INDIRECT OBJECT of V-IND
Semantic action:         NP is (semantic) RECIPIENT of V-IND (or another
                          slot, if specified by V-IND)
Result:                  V-IND, NP
```

When applied, this rule assigns "Mary" as the indirect object of "gave," and places the PERSON concept which represents "Mary" into the RECIPIENT slot of the ATRANS.

The final NP in the sentence, "the book," is attached to "gave" in a similar way. Semantics is asked to determine the best attachment of "book," which is represented as a PHYSICAL-OBJECT, to other concepts in active memory, which at this point contains the ATRANS as well as the person representing "Mary." Semantics determines that the best

[4] The way in which the semantic analyzer reaches these conclusions will not be discussed in this paper. For more details, see (Lytinen, 1984).

[5] Just as earlier, "Mary" could either be the ACTOR and RECIPIENT of the ATRANS, but "John" has already been assigned as the ACTOR.

attachment is to the OBJECT role of the ATRANS. The syntactic rule which can perform this attachment is the Direct Object rule, which is similar in form to the Dative Movement rule above. This rule is applied, yielding the final semantic representation (ATRANS ACTOR PERSON OBJECT PHYSICAL-OBJECT RECIPIENT PERSON), and the syntactic markings of "John" as the subject of "gave," "book" as its direct object, and "Mary" as its indirect object.

One important thing to note about the parsing process on this sentence is that although the Direct Object Rule could have applied syntactically when "Mary" was found after the verb, it was never even considered. This is because the semantic analyzer preferred to place "Mary" in the RECIPIENT slot of the ATRANS. Since a syntactic rule was found which accomodated this attachment, namely the Dative Movement rule, the parser never tried to apply the Direct Object rule.

The MOPTRANS parser is able to resolve syntactic ambiguities that proved difficult for past integrated parsers. For the sentence discussed earlier, "I saw the soldier called to his sergeant," MOPTRANS has no trouble determining that "called" is an unmarked passive, because according to its syntax rules, another indicative verb at this point is not possible. The rule which is applied instead is the Unmarked Passive rule:

```
Unmarked Passive Rule

Syntactic pattern:         NP, VPP
Additional restrictions:  none
Syntactic assignment:      NP is (syntactic) SUBJECT of VPP, VPP is PASSIVE.
                           VPP is a RELATIVE CLAUSE of NP
Semantic action:           NP is (semantic) OBJECT of S (or another
                              slot,   if specified by VPP)
Result:                    NP, VPP
```

"Called" is represented by the Conceptual Dependency primitive MTRANS, which is used to represent any form of communication. Since "soldier" can be attached as either the ACTOR or the OBJECT of an MTRANS, semantics would be happy with either of these attachments. However, the Subject Rule cannot apply at this point, since "soldier" is already attached as the syntactic direct object of "saw." Thus, this restriction on the Subject Rule prevents this attachment from being made. Instead, the Unmarked Passive Rule applies, since it semantically attaches "soldier" as the OBJECT of the MTRANS, and since "called" is marked as potentially being a past participle (VPP).

Unlike syntax-first parsers, the MOPTRANS parser can immediately resolve syntactic ambiguities on the basis of semantic analysis, thereby cutting down on the number of syntactic attachments that it must consider. We have already seen this in the example, "John gave Mary the book," in which the parser does not even consider if "Mary" is the direct object of "gave." Let us return now to two examples discussed earlier:

The cleaners dry-cleaned the coat that Mary found at the rummage sale for $10.

The cleaners dry-cleaned the coat that Mary found in the garbage for $10.

MOPTRANS parses the relative clause "that Mary found" with the following rule:

```
Clause Rule for Gap After the Verb  (CGAV Rule)

Syntactic pattern:        NP, RP (relative pronoun) (optional), V-IND
Additional restrictions: V-IND is not followed by an NP
Syntactic assignment:     V-IND is a RELATIVE CLAUSE of NP
Semantic action:          NP is the semantic OBJECT of the V-IND
Result:                   NP, V-IND (changed to CLAUSE-VERB)
```

The Subject Rule assigns "Mary" to be the subject of "found," since "Mary" is not yet attached syntactically to anything before it. Then, since no NP follows "found," and since the attachment of "coat" (a PHYSICAL-OBJECT) as the OBJECT of the ATRANS is semantically acceptable, the CGAV rule applies, assigning "that Mary found" as a relative clause.

When the parser reaches "for $10," the first S, "The cleaners dry-cleaned the coat," as well as the relative clause, are both still in active memory. The NP "$10" is represented as MONEY. The preposition "for" also has a semantic representation, which describes the possible semantic roles that a PP beginning with "for" can fill. One of these roles is called IN-EXCHANGE-FOR. "Dry-cleaned" is represented by the concept PROFESSIONAL-SERVICE, which expects to have its IN-EXCHANGE-FOR role filled with MONEY, since most professional services are done for money. ATRANS, on the other hand, does not explicitly expect an IN-EXCHANGE-FOR role. Thus, semantics prefers to attach the PP "for $10" to PROFESSIONAL-SERVICE and the verb "dry-cleaned."

In the second example, on the other hand, when the PP "at the rummage sale" is attached to "found," this triggers an inference rule that the ATRANS representing "found" must actually be the concept BUY, since "rummage sale" is a likely setting for this action. BUY, like PROFESSIONAL-SERVICE, expects the role IN-EXCHANGE-FOR to be filled with MONEY. Thus, semantics has no preference as to which verb to attach "for $10" to. To resolve the ambiguity, a syntactic recency preference is used, thereby attaching "for $10" to "found."

Because of this resolution of ambiguity, the MOPTRANS parser does not have to consider ambiguities further on in the sentence that it might otherwise have to. For example, in the sentence, "The cleaners dry-cleaned the coat Mary found in the garbage for $10 while she was away in New York," the PP attachment rule which MOPTRANS uses removes the relative clause "that Mary found in the garbage" from active memory, since the PP attaches to something before this clause. Therefore, when the parser reads the clause "while she was away in New York," there is only one possible verb, "dry-cleaned," to which this clause can be attached.

5 Conclusion

In this paper I have argued that semantic and syntactic analysis should be integrated. By this, I mean that syntactic and semantic processing must proceed at the same time, relying on each other to provide information necessary to resolve both syntactic and semantic ambiguities. Non-integrated, syntax-first parsers must leave some syntactic ambiguities unresolved until the semantic analysis stage. This can result in a highly inefficient syntactic analysis, because the failure to resolve one syntactic ambiguity can lead to other, "artificial" syntactic ambiguities which would not have to be considered had the original ambiguity been resolved with semantics. These new ambiguities may also be unresolvable using only syntax. If several of these ambiguities are encountered in one sentence, the combinatorics of the situation can get out of hand.

Previous integrated parsers have avoided these inefficiencies, but have suffered from problems of their own. Because of the lack of a separate representation of the input text's syntactic structure, it is difficult to write "local" syntax-checking rules to resolve some types of syntactic ambiguities. Attempts to resolve these ambiguities results in a proliferation of rules.

To solve both of these problems at the same time, the MOPTRANS parser is integrated, in that syntactic and semantic processing proceed in parallel, but MOPTRANS has a separate body of syntactic knowledge, and builds % representation of the syntactic structure of input sentences. This enables it to use semantics to resolve syntactic ambiguities, and to easily resolve ambiguities that cause difficulties for local syntax-checking rules.

The MOPTRANS parser relies heavily on its semantic analyzer during the parsing process. Therefore, its ability to parse is only as good as its semantic theory. Obviously no semantic theory presently exists which can allow for correct semantic analysis of arbitrary texts, or even for a broad domain of texts. However, in limited domains, this approach could prove to be more successful. In any case, given the present desire to use semantics in machine translation systems, an integrated approach to the use of semantics with syntax appears to be advantageous to the approach of syntax-first analysis followed by a semantic interpreter.

References

Bar-Hillel, Y. (1960). The Present Status of Automatic Translation of Languages. *Advances in Computers, 1,* 91-163.

Birnbaum, L., and Selfridge, M. (1979, October). *Problems in Conceptual Analysis of Natural Language.* Technical Report 168, Yale University Department of Computer Science.

Boitet, C., and Nedobejkine N. (1981). Recent Developments in Russian-French Machine Translation at Grenoble. *Linguistics,* (19)}, 199-271.

Chandioux, J. (1976). METEO: An Operational System for the Translation of Public Weather Forecasts, pages 27-36. *FBIS Seminar on Machine Translation.* American Journal of Computational Linguistics, Microfiche 46.

Lytinen, S. (1984, November). *The Organization of Knowledge in a Multi-lingual, Integrated Parser.* PhD thesis, Yale University, Department of Computer Science.

Marcus, M. (1978, February). *A Theory of Syntactic Recognition for Natural Language.* PhD thesis, Massachusetts Institute of Technology.

Riesbeck, C., and Schank, R.C. (1976, October). *Comprehension by Computer: Expectation-based Analysis of Sentences in Context.* Technical Report 78, Yale University Department of Computer Science.

Schank, R.C. (1972). Conceptual Dependency: A Theory of Natural Language Understanding. *Cognitive Psychology, 3(4),* 552-631.

Schank, R.C., and Birnbaum, L. (1980). *Memory, Meaning, and Syntax.* Technical Report 189, Yale University Department of Computer Science.

Wilks, Y. (1973). An Artificial Intelligence Approach to Machine Translation. In Schank, R., and Colby, K. (Ed.), *Computer Models of Thought and Language,* San Francisco: W.H. Freeman and Co.

Wilks, Y. (1975). A Preferential, Pattern-matching, Semantics for Natural Language Understanding. *Artificial Intelligence, 6(1), .*

Woods, W. (1970). Transition Network Grammars for Natural Language Analysis. *Communications of the ACM, 13(10), .*