TOPICS IN DISCOURSE ANALYSIS

by

James Edward Davidson

Technical Report 77-19

November 1977

Department of Computer Science
The University of British Columbia
Vancouver, British Columbia, V6T 1W5

Abstract

This thesis deals with the theory and analysis of connected English discourse.

The abstract theory of discourse, and its distinquishing characteristics, are discussed. Some problems in computer analysis of discourse are delineated; a method of analysis, based on a modified system of predictions, is introduced, and illustrated with examples from simple stories. A program embodying these concepts is described. Finally, possibilities for discourse, and its place in computational linquistics, are discussed, and directions for further work indicated.

- - - - - - - - - -

This work was supported by the National Research Council of Canada

TOPICS IN DISCOURSE ANALYSIS

by

James Edward Davidson

B.Sc., University of British Columbia, 1974

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in the Department

of

Computer Science

We accept this thesis as conforming

to the required standard

....

....

The University of British Columbia

September, 1976

Table of Contents

Table of Figures

Acknowledgement

I would like to thank my supervisor, Dr. Richard Rosenberg, for his patience and inspiration, and for his work in helping to make this thesis a reality. I would also like to thank Dr. Alan Mackworth, for his criticisms of preliminary drafts of this thesis. The financial assistance of the National Research Council is gratefully acknowledged.

0. Introduction

Natural language analysis, like most sciences, has proven amenable to the 'divide and conquer' philosophy: specify a small domain, and deal thoroughly with problems within that domain, while ignoring issues outside the domain.

This thesis follows a similar approach, except that in this case a different, and somewhat larger, target has been chosen -- the analysis of connected discourse.

Work with single sentences in isolation is recognized to be somewhat artificial, ignoring as it does the fact that language is just not processed out of context. It is the intent of this thesis to deal with some of the problems of connected discourse, in an effort to delineate exactly what 'context' is.

The first chapter is basically a survey of the previous work on discourse. The work of various linguists will be covered, and different theories of discourse considered.

In the second chapter, this work will be covered again, this time with a more pragmatic motivation: to discover the characteristics of discourse which make its analysis different from that of single sentences. A number of such characteristics will be described, together with their possible use in an analysis system.

The third deals with a somewhat peripheral area, psychology. I shall review the relevant work in the field, to provide a certain amount of validation for my approach.

In the fourth chapter, we return to the computational aspect of the problem. Previous work in computational linguistics is reviewed, and various issues of representation, control, etc., discussed.

The fifth, and most significant, chapter is a relatively complete specification of a model for discourse analysis. The various pieces of the model are explained, and the overall motivations (linguistic, psychological, and computational) discussed. A detailed example of analysis is presented, to illustrate the points made.

The final chapter consists of conclusions, and indications of possible future directions.

Like previous systems, the work here concerns a restricted domain. Problems of intention and belief are ignored completely. Speech acts are not considered. Questions of intension and extension are handled in a simplistic manner. The domain of analysis itself is heavily restricted, being confined to one area: written narrative discourse.

Why, then, study discourse?

I feel that it offers potential, in terms of effects which are simply not seen at the sentence level. Exactly what these effects are will become clearer as the thesis develops.

No attempt has been made here to explain in detail previous work in the field. Good reviews of computational linguistics can be found in Borgida (1975) and Cercone (1975).

It will be obvious that the work presented here draws heavily from many sources. In particular, Riesbeck's (1974; 1975) thesis was the inspiration for many of the ideas developed here; his influence will be obvious. Also, work by Schank, Winograd, and Marcus has contributed many insights. In the field of linguistics, the most significant impact has come from the work of Fillmore (1968), Chafe (1970), Grimes (1975) and Halliday (1967). Other authors are referenced, where appropriate, throughout the paper.

In many ways, this thesis deals with an approach to discourse, rather than a complete system. To quote a popular phrase, 'this paper raises more questions than it answers'. The system described here has been partially implemented, but much remains to be done. It is my hope that the reader will, at the end, have a clearer grasp of the issues involved.

I. Discourse

This chapter is an effort to characterize the general notion of discourse, in order to provide a basis for the work which follows.

It is divided into two sections; the first presents a broad overview of what we commonly call discourse, and the second focusses on one specific aspect of this -- narrative discourse.

I.1 Discourse in General

"Discourse" can be described as "connected communication of thought"[1]. In fact, it is more than this; a discourse is somehow more than just a string of connected sentences, and this fact must be taken into account for effective analysis.

Discourse has appeared under a number of names in previous work: 'rhetoric','composition','communication', and 'discourse' are all aspects of the same topic, which I shall simply call 'discourse'.

The primary motivation for discourse study is linguistic. Some linguists currently recognize that language cannot be studied effectively without going beyond the sentence level.

In earlier linguistics, from Bloomfield to Chomsky, the sentence was held to be the largest coherent unit of linguistic structure[2]. As Chomsky (1965, P. 3) stated:

> '[This study] will be concerned with ... the rules that specify the well-formed strings of minimal syntactically functioning units...' [where 'minimal syntactically functioning units' were later stated to be sentences].

The major reason for this was complexity; the traditional linguists saw discourse as a large, uncontrollable mass, not amenable to the precise mathematical formalization which they could apply so effectively to sentences.

The emphasis is now changing. As Sanders (1970, p. 73) remarked:

> 'It can thus be concluded that the only possible natural domain for a scientific theory about any language is the

---

[1] Funk and Wagnall's Standard College Dictionary, Toronto, Fitzhenry and Whiteside, 1974, p. 380

[2] For example, Bloomfield (Language, 1933, p. 170) defined the sentence as "an independent form, not included in any larger (complex) form". Beneviste (Problemes de linguistiques general, 1966, p. 128) stated "Un phrase ne peut donc pas servir d'integrant a un autre type d'unite"[A sentence cannot serve as an integrant of another type of unity]

infinite set of all possible discourses of that language.'

The main reason for the switch is that it is recognized (by some linguists, at least) that there are linguistic constructs which can only be seen at the supra-sentential level. That is, language is used for the purpose of communication, and any attempt to ignore this by imposing artificial constraints is doomed to failure.

Interestingly, a number of uses of discourse study can easily be seen. First, there is the possibility of learning how to _produce_ effective discourse. A common flaw in literature (both fictional and non-fictional) is the failure to recognize and make use of inherent structural features which increase readability. In fact, the journal _College Composition and Communication_ conducted a symposium on the paragraph (Becker, 1966; Christensen, 1966; Rodgers, 1966) addressed to precisely this problem. Obviously, a theory of discourse will not provide an immediate solution, but it will constitute a step.

Second, translation might be more easily done. One of the obvious desiderata to all who watched the Machine Translation programs of earlier years was the failure to take into account more global 'context' mechanisms. It seems that machines and people alike would benefit from the ability to effectively characterize a discourse.

Related to this is the work on abstraction and content analysis. (see, for example, Holsti (1969), or Carney (1972)). For obvious reasons, it is desirable to be able to extract the 'central thread' from a text, to store it for whatever purpose. Again, this seems difficult without a coherent theory of what a text _is_.

Given the possible uses for discourse analysis, what work has been done so far? Surprisingly (to me at least), there is a fair amount, although much of it falls into fields peripheral to linguistics.

One of the standard fields of study, dating back to Aristotle, is _rhetoric_. It is still taught today, as a method of enabling the writer to express himself more powerfully. Perhaps it should be criticized for being prescriptive rather than descriptive, but it is nonetheless an important field (note that it comes close to satisfying the first advantage mentioned earlier, of production of effective discourses).

Literary criticism is another field which has taken on aspects of discourse analysis. In order to successfully review a work, the critic must have in his own mind a fundamental idea of what it says, and more importantly, whether it says it effectively. Although there is little here in the way of written guidelines, the field has produced some interesting examples of discourse structure.

The remainder of the work on discourse has come from within the fields of linguistics, although even here it seems scattered, and out of the mainstream.

Discourse in General

One such example is Longacre (1970), who worked under Pike in the field of tagmemic grammars. The work is too complex to be discussed here, but he did at least provide a reasonable taxonomy of types: narrative, procedural, expository, hortatory, dramatic, activity, and epistolary.

Kinneavy (1971) provided what is probably the single most comprehensive basis for work in discourse. He identified the categories of reference, literature, persuasion, and expression (similar to Longacre's taxonomy), but more importantly, presented a reasonably coherent theory of a discourse. He saw the encoder (speaker), the decoder (hearer) and reality, as providing the vertices of a triangle, of which the signal (discourse) is the inside.

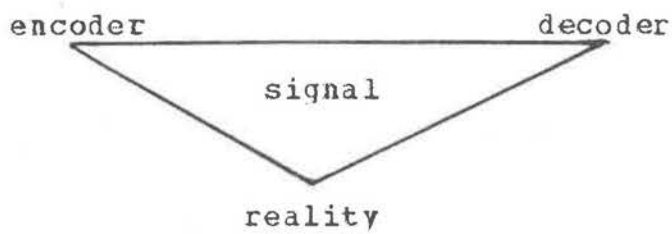


Figure 1 - A Schema for Discourse

Thus, the <u>function</u> of the discourse is formalized: to communicate from the encoder to the decoder some signal, which bears an undetermined relationship to the real world. Kinneavy developed in detail each of the four points mentioned, and illustrated different examples of the discourse situation; space precludes my saying more here.

Within this one framework, we can see how to deal with some of the more advanced notions of current linguistics: speaker/hearer relationship, speech acts, purpose, communication situation, etc.

Interestingly, various peripheral disciplines have emphasized different aspects of this structure. For example. Shannon, in his work on information theory, paid less attention to the possible relationships to reality, but added a new component, noise, to the signal. Morris (1946), a sign theorist, distinguished the actual reality (denotatum) from what it means to the interpretant (significatum). More detail on this matter is given in Kinneavy's original work (pp. 18-26).

Hausenblas (1964) of the Czech school, developed another method of classifying discourse. He identified three dimensions along which a discourse can vary: simple/complex; dependent/-independent (of situation); and continuous/discontinuous. These components seem to characterize the <u>style</u> of the discourse, rather than its content (cf. Longacre).

Thus, we have been able to identify the function (Kinneavy), content (Longacre), and style (Hausenblas), as potential descriptions of discourse.

Within the general field of discourse, two linguists -- Halliday (1967; 1970; Halliday and Hasan, 1976), and Grimes

(1971; 1972; 1975) -- have completed a tremendous amount of work. Their influence will be seen throughout this paper.

Discourse in its entirety is too broad a field to be covered in a study such as this. Since we want ultimately to design a functioning model for discourse analysis, a smaller domain must be selected, with more specific rules of structure and function.

In determining what kinds of discourse to study, one question concerns the medium: is the discourse to be spoken or written? We will deal with the latter because it is more 'complete', in the sense that everything is inside the discourse. Spoken discourse (i.e. conversation) makes extensive use of extralinguistic information: the speaker/hearer relationship, objects in the environment, etc., as well as such medium-dependent features as intonation. Such is not, in general, the case in written discourse (although there can be exceptions). Also, written discourse generally adheres more closely to standard notions of grammaticality; there is less use made of sentence fragments, ejaculations, ellipses, elisions, etc. Accordingly, the study will be confined to written discourse.

The question of type (in Longacre's classification) must also be decided. I have chosen narrative, for a number of reasons. First, it is well structured; narratives have an inherent organization, as has been observed by a number of researchers. Second, there already exists some work in the field; the same researchers mentioned above have investigated the problem thoroughly, and presented a few answers. Third, narrative seems to lend itself best to compactification; one can deal with very short stories, and still not have the feeling of having removed the significant aspects.

This section has been an effort to characterize discourse, and indicate the general truths which hold about it. The next section deals with one particular aspect of this, narrative discourse.

I.2 Narrative Discourse

Throughout this section, 'discourse', unless otherwise specified, will refer to narrative discourse. By 'narrative' in this case, I shall mean very short (i.e. one-paragraph) stories. In keeping with the best tradition of Eugene Charniak and Roger Schank, the examples I will deal with later will be of a fairly simple nature, at the level of perhaps a 9-year old.

One example might be:

> Mary was making dinner. She told John to put the casserole in the oven, and gave the salad to Peter. Then she took the dessert out of the fridge. Finally, it was ready; everybody sat down at the table.

An example of this sort avoids most of the esoteric and complex syntactic constructs which, while of interest to some linguists, are irrelevant to our stated goal of discourse analysis. At the same time, it preserves the essential features of a discourse: the cohesiveness, the temporal continuity, the use of characters, etc., so that any principles derived here will hopefully be generalizeable.

Given these examples of discourse, what kinds of general statements can be made about them? Probably the most important and noticeable aspect of a narrative is its temporal connectivity. Generally, there is a central thread of time running through the story. This tends to serve as the background, upon which other elements are hung. In some languages, in fact, the style of discourse is sufficently constrained that the telling of an event must take the same order as the event itself.

Grimes (1971) identified several different _components_ of discourse. The first are _events_. These are the actions of a story, usually constituting its most important part. They provide the sequencing of a discourse, and are incorporated into its central time line. We can describe the sequencing of events as being either tight (i.e. contiguous in time) or loose (i.e. containing lapses). (Interestingly, this distinction can be traced back to the ancient Greek, and the development of the perfect tense.) Events tend to be clustered -- that is, gathered into small islands of contiguous groups, with gaps between the islands.

The next class is that of _participant_. Intuitively, a discourse contains a 'cast' of characters; this may vary in size from one to thousands (although usually no more than 20 or so will be developed). Participants are related to events and other participants. The continued use of a restricted set of participants provides a certain amount of cohesiveness in a discourse.

Most discourses will have a _setting_, locating the events both spatially and temporally. The description of the setting may be brief, or it may be omitted entirely, with the reader left to make his own assumptions.

There is generally a fair amount of peripheral information which may loosely be titled _background_. This is usually explanatory information, designed to convey to the reader something he would not otherwise have known. Interestingly, the amount of background information used can be an indication of the development of the narrative. If the author has kept the background information to a minimum, he is probably implicitly asserting that either:

i) the situation being discussed is quite simple;
ii) the reader is already relatively familiar with the various parts.

Grimes refers to another class, _evaluations_, as instances in which the author conveys his own feelings about an event or character. This is sometimes done explicitly; more often, it is done implicitly, through the connotative meaning of the various words chosen (compare 'stubborn' and 'tenacious').

The last, although still very important, class is collateral: a catch-all group containing information about the discourse itself -- essentially, a meta-discourse. This can appear in a number of ways. One is foreshadowing, the hinting of events to come. Another is alternative: the mention of events which did not happen, used to emphasize those that did. Yet another (more common in expository than narrative discourse) is summary: a reiteration of the contents of the discourse. Collateral will be discussed in more detail in the next section.

This gives us a reasonable feeling for what a discourse is. One part not discussed yet is coherency -- what it is that makes a discourse adhere together, rather than remaining a collection of isolated units. This will be dealt with more fully in the next chapter, but a certain amount can be said now.

One of the foremost cohesive methods is reference. This is related the question of participants, mentioned above; if a previously identified participant must be referred to again, an 'abbreviated' form can be used. Essentially, the problem of reference can be broken into two subgoals:

i) establishing the identity of the referent;
ii) maintaining this binding.

The devices used to do this (pronominalization, definite noun phrases, demonstratives, etc.) are indicative of the specificity of the reference, and, like background, carry a lot of implicit information about the salience or newness of the referent.

Phillips (1975) also develops some ideas for coherency. Besides anaphora, one method he sees is causality; the fact of one event directly causing another seems to unify the text. Schank (1974a) developed this further, defining a complete taxonomy of types of causation, and showing how they contribute to overall unity.

Another method is spatio-temporal connectivity. That is, events must appear to be connected in space and time, and to follow naturally in some way.

A last cohesive aspect, which will be covered in considerably more detail in the next chapter, is theme. This is essentially the 'topic' of the discourse, and it may take a tree-like organization (Lakoff, 1972).

These factors all seem to apply to narrative discourse, but at a 'local' level. That is, what we have been dealing with is the microstructure of the discourse. In the next section, we will look at various efforts which have been made to characterize the macrostructure (i.e., the global organization).

I.2.1 Text Grammars

'Text grammar' is a loose term, used to describe a series of efforts by Propp (1958), Kummer (1972), van Dijk (1972), and Rumelhart (1975). These have endeavoured to provide a 'grammar' for stories, in the sense that a set of (often context-free) rules can be given, with which it is theoretically possible to generate the stories (or, in our case, to 'parse' them).

Propp provided the earliest, and perhaps least structured, work. His conclusions were based on an analysis of about one hundred Russian folktales. Unfortunately, the results were somewhat fuzzy; as Lakoff (1972) pointed out, his categorization of events as belonging to specific classes is highly suspect. Nonetheless, this provided a start on the problem

Kummer and van Dijk, working within the transformational grammar paradigm, produced formal, well-specified systems. Unfortunately, their models suffered, to a certain extent, from the same flaws as other work in transformational grammar -- counterintuitiveness and lack of perspicuity. That is, the mathematical formulae may reflect superficial properties of the discourse, but they do not correspond in any easily understood way to our underlying intuitions about discourse. Thus, the models should probably be dismissed as a serious but somewhat impractical endeavour -- although their study could lead to certain insights about the nature of discourse.

Of all the systems, that by Rumelhart would seem to be (not surprisingly) the one best suited to the artificial intelligence approach. Rumelhart developed a system that was based to a large degree on intuition -- but which embodied most of the concepts associated with discourse. An example of his grammar is:

```
story   --> setting + episode
setting  --> (state)*
episode  --> event + reaction
```

The problem with his system is, literally, that we are left hanging. The grammar is quite imprecise, and the 'primitives' (i.e. terminals) are expressed at such a high level that it is difficult to see how they are represented in actuality. For example, 'state' is left as an unanalyzed (and unanalyzable) primitive. This creates problems in analysis of discourse, since presumably a 'state' could be manifest in a number of different ways, and the search for it could prove explosive.

I feel, however, that this is the right direction to follow. The nature of the system is such that it could hopefully be extended to a more concrete level.

In general, text grammars would seem to be a viable notion, if developed in the right way.

This concludes the chapter on discourse. In the next, the problem of <u>analysis</u> of discourse is re-examined.

II. Relevant Features of Discourse

In this section we will review the characteristics of discourse mentioned earlier, this time considering their applicability to a theory of analysis.

In many cases, the examples used will be individual sentences; this is done for reasons of space. The properties described here are all extensible to and applicable at the discourse level.

Essentially, discourse is, as mentioned, more than just a group of sentences, and our model must reflect that. That is, any system that analyzed a discourse by merely analyzing its component sentences would be a failure, having missed the primary characteristic.

In considering discourse analysis, two questions come to mind:

i) what features of discourse make it easier (or perhaps harder) to handle than single sentences in isolation?

ii) what form should the final result of the analysis take; i.e. how can the meaning of a discourse be represented?

About the latter I shall have little to say, and even that will be postponed until a later chapter.

The former can be discussed here. What we want to characterize is the additional information available in a discourse which is not found at the level of single sentences. More importantly, we must show how this can be used to aid in analysis.

It has been recognized for some time in computational linguistics work that in order to be properly analyzed, a text must be, in some sense, understood -- i.e. the analysis system must have fairly extensive knowledge, about both the subject domain and the discourse medium. For this reason, the somewhat simplistic work in early computational linguistics failed. What we need here is a way to get a handle on the information necesary to correctly analyze discourse; we know that syntax alone is not enough.

Several potentially helpful features can be identified; I shall deal with them one at a time. Some of these were described in the previous chapter, some have been mentioned in other computational linguistics work, and some are totally new.

For each feature, two things need be expressed:

i) the surface manifestation of the feature -- i.e. the form it takes in a discourse;

ii) the organizational information contained in the feature -- i.e. what it tells us about the structure of a discourse.

II.1 Staging

This is described (Grimes, 1975) as the manner in which the speaker organizes the information for the hearer's benefit. The most important aspect of this is the <u>theme</u>: the 'point of departure' for the speaker, with respect to which he organizes the discourse (see Halliday, 1967, for a comprehensive discussion on this). More specifically, theme is the (abstract) concept identified; <u>topic</u> is the surface form used to signal it.

Theme occurs at the sentence, paragraph, and discourse levels (see Phillips, mentioned earlier). It can most easily be identified at the sentence level, and that one will be dealt with here, although sentences can easily be related to the discourse level.

In surface structure, theme may be marked or unmarked. If it is unmarked (as in most cases), the first nominal concept in the sentence constitutes the theme.

The horse ran from the corral.

In this case, the speaker is treating 'the horse' as his point of departure.

The theme can be more precisely identified if it is marked; this can be done in a number of ways.

<u>Fronting</u> is the rearrangement of the sentence to put the desired unit into initial position. The most common manifestation of this in English is the passive voice.

The city was razed by the earthquake.

Here the rearrangement has served to emphasize that the city, and not the earthquake, is what's being talked about.

Other methods of fronting exist:

Home is where I want to be.

The effects are the same.

<u>Partitioning</u> is another method of reordering the sentence to highlight the theme. One method of this is extraposition (clefting):

It's Bill who took the car.

Another form is pseudo-cleft:

What I want for Christmas is a new bike.

Note that in the first case, 'Bill' would have been the theme even in the unmarked form; the marking merely adds emphasis. In the second example, the pseudo-clefting emphasizes the uniqueness of the theme -- i.e., 'I want a bike (and nothing else will do).'.

<u>Embedding</u> can also serve to emphasize a concept. Compare:

John opened the door with the key.
John used the key to open the door.

In the second case, the failure to use the standard instrumental case indicates that the key is important.

Theme can be marked in a number of other ways; often the marking is hard to recognize. Some languages have more rigid rules; often the theme will have inflectional features, and there may be different features for the sentence theme and paragraph theme.

The ability to recognize the theme, even if it is only at the sentence level, is quite beneficial in analysis. It essentially conveys the <u>orientation</u> of the sentence: where the speaker is starting out from. This can be used to control such processes as foregrounding, etc.

II.2 <u>Cohesion</u>

Halliday (1967; Halliday and Hasan, 1976) and Grimes (1975) identify this as the manner in which incoming information relates to the information previously given. There are a number of aspects of cohesion, of which two are relevant: information blocking, and the specification of reference.

II.2.1 Information Blocks

Information blocks are unified chunks of information, which are thrust upon the hearer one unit at a time. Their significant aspect is their length, which essentially indicates how much information the hearer is to absorb, and thus how important that information is (this has been referred to as the 'rate of information injection'(Grimes, 1975, pp. 274-275)).

Information blocks are signalled in speaking, through intonation. In writing, punctuation (especially commas) and clause separation are the usual methods.

For example, compare:

> The beaver, a timid animal, is rarely seen in its natural environment.
>
> The timid beaver is rarely seen in its natural environment.

In the first example, the use of the appositive clause creates a new information block, and serves to emphasize the beaver's timidity. In the second example, the adjective has been absorbed directly into the longer block, thus down-playing its importance.

Information blocks also capture the 'given/new' distinction. Some blocks may contain only new information, others both given and new. In the latter case, the given information generally constitutes the first part of the block, with the new information following. 'Given' is often confused with 'theme'; as Halliday explains it, given information is what has been talked about before, theme is what is being talked about now (see Chafe, 1974, for more comment on this).

The discovery of an information block can be valuable in discourse analysis, since it generally indicates how salient the information contained in it is. In particular, longer blocks generally contain less important information than shorter ones -- i.e., their rate of information injection is lower.

II.2.2 Reference

As mentioned previously, reference is one of the stronger cohesive elements (empirical verification of this will appear in the next chapter). This fact is often overlooked in computational linguistics, where the common practice is to merely establish the (extensional) identity of the referent.

The important aspect of reference is its <u>specificity</u>. The various forms of referring (pronouns, demonstratives, inclusive nouns, definite noun phrases, etc.), are all specific to a greater or lesser degree. The specificity indicates the extent to which the speaker feels it necessary to 'point out' the referent -- i.e. how strongly he thinks it is present in the hearer's memory.

The form of reference we are dealing with here does not connect very closely with Charniak's (1972) work on reference <u>resolution</u>. The information carried in our case does not usually help in resolution (although it may), but serves rather to indicate the salience of the referent. Compare:

> the left arm of the chair

the arm

it

as referring expressions. Each of these has implicit in it assumptions about the 'presence' of the referent in the memory of the hearer. Again, see Chafe (1974) and McDermott (1976) for comment on this.

Other forms of cohesion exist, including: substitution, ellipsis, lexical cohesion, and conjunction. Space precludes dealing with all of these here; they are covered more fully in Halliday and Hasan (1976).

II.3 Collateral

This was covered in the last chapter (pp. 12-13); it is, as the name implies, peripheral information, outside the actual content of the text, whose purpose is to emphasize parts of the text.

A number of types were mentioned, including foreshadowing, alternative, and summary. Of these, the use of alternatives is probably the easiest to recognize. This is indicated, usually, by the presence of a negated clause (or negating conjunction):

> We might have been killed, but the plane landed safely.

The effect here is to stress the positive aspect, i.e. 'we are still alive'.

Summary is a form rarely seen in narrative discourse, but it is occasionally found in children's stories:

> This is a story about Goldilocks and the three bears. ...

Even though it may appear at the start of the story, this is an example of summary.

The benefits of collateral are immense; essentially, it provides gratuitous information about how to organize the discourse. The task of summarizing a story is obviously greatly simplified if the author does it for us.

II.4 Text structure

As mentioned earlier (I.2.1), the best work on text structure seems to be that of Rumelhart. In theory, we should be able to develop a set of rules which would enable us to generate or parse a story. The problem, as indicated, lies with the gap between the top and the bottom -- i.e. between the terminals of the grammars and the actual surface forms of the discourse.

The main possibility would seem to lie with developing a restricted grammar to handle certain sets of stories. For example, one rule might be:

```
setting --> temporal location |
            spatial location |
            (character)* |
            continuing event
```

where there is an implicit inclusive or between the branches of the rule. Thus, a sentence like

> Billy was playing in his yard.

fulfils three of the four possibilities.

Note that the 'inclusive or' cannot easily be handled with the standard context-free rules. What is needed is a more expressive control structure -- not only context-sensitive but powerful and intuitive. This particular problem will not be discussed here.

A grammar of the type described could probably be made to work, if the input set of stories were sufficiently constrained. The current uses of 'frames' and 'scripts' in artificial intelligence would seem to fall into this class.

II.5 Context

'Context' is a somewhat nebulous term, used to describe the 'surroundings' of the current input. The word derives from the Latin for 'weave together', and this is as good a definition as any. We can identify several direct contextual effects:

i) change of word sense;
ii) change of importance -- i.e. what this component means to the overall discourse;
iii) reference, ellipsis, etc.

Sentences in isolation have different meanings from those same sentences in context. Compare these examples from Reisbeck (1974):

> John and Mary were racing. They were afraid of being beaten.
> John and Mary were running. They were afraid of being beaten.

The meaning of the second sentence changes completely, depending on what precedes it -- even a change to a single word affects the context.

In working with context, it is tempting to say that we need merely save all the information in a discourse, so that it is available when we need it. This approach, however, quickly becomes explosive; what is needed instead is a well-specified indication of which information to save, why to save it, and how to use it later (see McCalla, 1976, for a discussion on this).

Many of the benefits of context are manifest in the theme (see 'staging', above), but two others are relevant here.

II.5.1 Foregrounding

The first is what Chafe (1970; 1974) has referred to as foregrounding. At any given time, certain concepts are 'on stage', in that they are in that they are in the consciousness of the hearer. Concepts are brought onstage by being referred to, and can be kept there via repeated reference. They seem to leave the stage simply fading away over time, although this can be affected by various aspects of the intervening discourse.

An example might be

> I just found a book belonging to Peter. I wonder where he's living now.

Here the 'he' clearly refers to 'Peter', as the only suitable foregrounded concept ('book' is also foregrounded, but cannot match the pronoun).

Complex rules can be derived, both for the foregrounding and 'unforegrounding' (backgrounding?) of a concept. Chafe (1974) presents a comprehensive discussion.

II.5.2 Frames

The other use of context which is easily identifiable falls under the notion of frames (Minsky, 1974) (here used in the sense of a general situation, rather than a plot-oriented script). For instance, the text

> John and Peter were playing baseball when the bat cracked.

is perfectly coherent because we know that the baseball frame has a 'slot' for bat.

In some sense, this is merely another aspect of foregrounding, but I feel it is better to identify it in its own class. Basically, the function of frames (context clusters, beta-structures, schemata,...) is to provide the understander with knowledge in useful-sized chunks -- i.e. to partition the memory, in an effective way. They also provide power for word-sense disambiguation, and possible control of inferencing. Frames can become arbitrarily complex, almost to the point of text grammars, or arbitrarily simple, almost to the point of foregrounding.

In order to use the notion of frames effectively, we need a reasonable idea of how and when frame are activated and deactivated. This is a common problem in current A.I. work, and I have no easy solution. In the work here, I will use a simplistic mechanism, in which a frame is activated by reference to the appropriate word or concept, and deactivated (i.e. 'pushed out') when a replacement frame is activated at the same level.

There is obviously much more to context than what is described here; for the moment, however, this will have to do.

II.6 Real-world Knowledge

One point that has not been heavily stressed so far is the semantics of the discourse -- the actual content. Lest I give the impression that form and style are all that are needed, I will present a brief summary here of the use of real-world knowledge.

Perhaps the best example of this is Rieger's (1975) verb-driven inference program, which makes semantic deductions based on the input. The system makes forward inferences in an uncontrolled manner, trying to deduce all possible facts from a given input.

The system in its original form is explosive, but with some effort we should be able to provide enough coherent direction that the process will be more controlled. Of Rieger's sixteen inference classes, four (specification, function, cause, and result) are valuable in discourse. This reflects our previously-stated notion that causality is a major cohesive factor in a discourse.

For example, given the sentence:

John hit Mary.

our system might make the following 'predictions':

cause:
John was angry at Mary.

result:
Mary hit John back.
Mary started crying.

specification:
John hit Mary with his hand.
John hit Mary with a hammer.

function:
(none applicable)

Thus, the semantic inference provides us with, in some sense, the ability to 'understand' the story. From this, we can predict what comes next, and the work involved will be simplified if the event does in fact happen.

One factor which seems to control the explosion of inferences is an interesting 'windowing' effect: the inference mechanism can use only the current part of the discourse, and not previous input, as a basis. For example, if after the 'John hit Mary' sentence, we find

Mary hit John back.

the old set of inferences (about how and why John hit Mary) would die off, since the sentence upon which they are based is no longer 'in the window'. A new set of inferences is spawned instead, based on the concept of Mary hitting John.

Another method of inference control was suggested by Rosenschein (1975). His approach viewed inferencing as an operation dependent upon a whole <u>set</u> of facts. Under this assumption, the system was designed to find the 'least possible pattern' (i.e., minimal extension to the set). The requirements for this inference were that it:

- i) cover the input set of facts (i.e. make maximum possible use of the given information);
- ii) be independent of the current facts (i.e. not assert something already known);
- iii) be minimal (i.e. make the fewest possible assertions).

There are obvious flaws in this approach, but it merits inspection.

Thus far, we have identified half a dozen features of discourse which are available to assist in analysis. There are others, notably presupposition and diction (lexical selection).

I will refer to these as <u>information sources</u> (I.S.'s). Several characteristics of these sources can be identified:

- i) They are <u>weak</u>; the amount of information they carry is not as strong as the standard areas of syntax and semantics.
- ii) They may have nothing to say; the module for collateral, for example, may lie dormant for long periods.
- iii) They are <u>gratuitous</u>; the information is there anyway, so a proper analysis system must use it.

Given these conditions, how can we incorporate the I.S.'s into an analysis system to make the most effective use of them.

What we would like theoretically is a clean modular system, with each module suggesting things whenever it recognizes its own need. This brings us to the standard A.I. problem of interacting sources of knowledge. A possible control system will be outlined in chapter V.

As an aside, the features discussed here are significant, because they draw attention to the form/content dualism in language. That is, the speaker (writer), in preparing his message, is concerned with two points:

i) what to say;
ii) how to say it.

where (ii) describes the _form_ of the discourse (i.e., the manner in which it is structured), and often contains a lot of information relevant to the meaning of the discourse. Halliday (1970) recognized the difference, referring to content as _ideational_ meaning, and to form as _interpersonal_ and _textual_. Unfortunately, the trend in computational linguistics work has been to downgrade the importance of this aspect.

Thus we have outlined the theoretical constraints of a discourse analysis system. The next section deals with a peripherally related area, psychology, but the following one (at last) presents a step towards a solution.

III. Psychology and Natural Language

This chapter represents a minor, but necessary, digression from our central topic of discourse analysis. The intent here is to present some of the background work in psychology, which has had such a pervasive effect on Artificial Intelligence. It is by no means a complete survey, nor would I wish it to be one. Rather, I hope to extract some general themes -- ground rules, as it were -- which can be used for guidance in later implementation.

In discussing work in psychology, I will not elaborate upon the methods used in various experiments. Rather, the conclusions reached by each experimenter will be presented, and the results interpreted in light of the goal of language analysis. Through this, I hope to build a simple model of the language understanding process.

The work in psychology which is relevant to language analysis can be effectively categorized into five classes:

i) information theory and related work;
ii) memory for sentences;
iii) memory for discourse;
iv) organizational schemata;
v) other work.

Each of these will be dealt with separately. Note that the form of memory in general will not be discussed; this is too broad a topic, and a number of comprehensive references already exist (see, for example, Tulving and Donaldson (1972), Lindsay and Norman (1972), Anderson and Bower (1973), Norman and Rumelhart (1975), Cofer (1976), and Norman (1976)).

Much of the early work in the psychology of language involved list-learning experiments, in the style established by Ebbinghaus (1885). As such, it is not generally relevant to our specific purposes. In the more recent work, however, a number of interesting results have been produced.

III.1 Information Theory

One of the earliest works was Miller's (1956a) seminal paper on coding processes, or 'chunking'. Miller's assertion was that units of arbitrary complexity could be retained in short-term memory if they could be 'chunked' -- i.e. converted to organized units. He found that subjects could retain a relatively constant number of chunks in semantic memory, regardless of the complexity of the chunks (see Simon (1972) for some comment on this). The implications for language processing are obvious. Intuitively, words entering into short term memory remain there as individual units, until chunked into larger groups (e.g. a phrase); these larger groups, in turn, may later be combined into still larger units. The interesting effect, of course, comes when the upper limit of short term memory (in Miller's case, 7) is approached. At this time, the input *must* be organized in some manner, or else information will be lost from short term memory.

Related to this are the 'perceptual strategies' of Bever (1970), who discovered:

i) psychologically, the clause is the main element of the sentence -- subjects tend to group smaller elements up to the clause level;

ii) subjects tend to perceive clauses in a basic S-V-O order; hence transformations such as passivization delay processing, since they force a reanalysis;

iii) subjects treat the first clause of the sentence as the main one; any shift in the ordering causes a delay in processing.

Interestingly, these effects can be largely explained within the framework of the chunking hypothesis; the strategies mentioned are part of an effort to prevent information overload, by making as much use as possible of the syntactic structure of the input.

We are thus faced with a model of a hearer who is processing information as rapidly as possible, to remain within the constraints of his memory system. Intuitively, he does this with the aid of the highly complex structural features of natural language. A few were mentioned in the last section, in dealing with cohesion, staging, collateral, etc. A more general approach will be taken here.

A starting point for such an investigation is provided by the incredible <u>redundancy</u> of natural language. Shannon (1952), in his pioneering paper on information theory, stated that the entropy of English, at the level of single letters, is roughly 1.2 bits, rather than the 4 bits or so that might be expected from random words. This redundancy would seem to be a result of the form/content dualism; if content alone were the determining factor, redundancy should be eliminated. Obviously, this finding cannot be applied mechanically, but the principle would seem to be sound: at any point in the discourse, we can predict, to a greater or lesser degree, what will follow. Using the inherent structure of language, we are able to 'guess' intelligently.

In A.I. terms, this is a case of the standard interaction between top-down and bottom-up information (about which more will be said later). Basically, analysis proceeds in a bottom-up mode, until some guesses and predictions can be made, at which point it shifts to top-down. This interchange goes on at several levels at once.

Thus, the characteristics would seem to be as follows: analysis can proceed in bottom-up mode, as long as it does not exceed the limits of short term memory, At each point, however, we are -- consciously or unconsciously -- making predictions about what is to come next. The problem of an effective natural language program, then, would be to make these predictions in a powerful and non-explosive manner. Several sources of information are available for this purpose; the ones mentioned in the previous chapter provide a start.

Interestingly, a model for language analysis embodying these concepts already exists. Marcus (1974; 1975) has implemented a system which satisfies precisely these constraints. His 'wait-and see' system is allowed a limited lookahead (based on predictions from the input so far), after which time it must make a decision.

The original prediction system, from which my work has been drawn, was that of Riesbeck (1974). His system, however, was designed with different goals in mind. Further comparison will be provided in the next chapter.

III.2 Memory for Sentences

Another very popular (and relevant) domain of psychological work concerns the memory forms used to encode sentences, and the processes used to construct these forms.

The surge of transformational grammars in the early 1960s resulted in much effort, despite Chomsky's protests, to prove that a sentence was stored internally in its base form (i.e. deep structure). This assumption, which came to be known as the derivational theory of complexity, has implicit in it the assumption that a sentence requiring more transformations (i.e. passive, negative) will take longer to comprehend than one requiring fewer transformations (see Fodor and Garrett, 1966, or Ammon, 1968). For a time, this outlook seemed valid (for example Miller and McKean, 1964, found that passive and negative transformations have an additive effect on cognitive complexity). However, other results soon began to contradict this theory, and it was soon abandoned.

Interestingly, at about this time, Chomsky (1971; originally published in 1968) retreated from his earlier position, and admitted that the surface structure of a sentence does have an effect on meaning -- i.e., that the deep structure does not adequately represent the meaning of the sentence.

One interesting result produced in this paradigm came from the work of Marks and Miller (1964), who were investigating the effects of syntax and semantics on sentence comprehension. The derivational theory of complexity would predict that the only controlling variable would be syntax, whereas some of the more modern theories would put the emphasis on semantics. Marks and Miller found that either syntax or semantics alone worked about equally well (in enabling subjects to understand sentences), but that the two together produced much better results. This indicates that subjects use whatever information is available to aid them in the comprehension process.

The question now becomes "what do people store as a result of understanding sentences?". A number of experiments were run to test this.

The first, and perhaps most important, of these was that by Sachs (1967), who discovered that subjects were able to recall and recognize the meaning of sentences quite well, but were unable to detect changes in surface syntactic form. This would indicate that what is stored is some sort of propositional meaning representation, rather that a syntactically oriented structure.

Bransford and Franks (1971; Bransford, Barclay, and Franks, 1972) carried this one step further, and discovered that subjects were unable even to distinguish the separation of individual sentences. In their experiment, the subjects were given a series of short sentences, and then asked to recognize various combinations of these. In general, there was a marked tendency for subjects to 'coalesce' the meaning, and to 'remember' more holistic units than had actually been observed. Independent and rigourous verification of this work was provided by Johnson-Laird (1970), Bock & Brewer (1974), and Griggs (1974).

Kintsch et al (1976) developed this to an even greater degree. After telling the subjects a short (one-paragraph) story, they discovered that the subjects were unable to distinguish between

i) information explicitly given in the text;
ii) information inferred from the text;
iii) information previously known by the subject.

Thus, it seems that the information had become totally <u>integrated</u> into the subjects' memory, to the point that it could no longer be distinguished as a separable text.

There is other evidence in this regard; space precludes a more exhaustive analysis. Essentially, the various experiments mentioned seem to point to one conclusion: what is stored after the analysis of a sentence is not any sort of syntactic structure, surface or deep, but rather some abstract semantic form, which is fully integrated into the subject's memory.

III.3 Memory for Discourse

With the domain of the sentence having been, in some way, disposed of, it is productive to turn our attention to the more complex area of psychology and discourse. As Frederiksen (1976, p. 1) points out,

> 'Most of the knowledge which humans acquire in a lifetime derives ... from organized information units which possess a high degree of structure'

Thus, it is imperative that any psychological research in language examine the problem of discourse comprehension.

A number of obvious questions can be raised here:

i) What is the form of the memory structures resulting from discourse?
ii) How are these structures built?
iii) Why are certain parts of a discourse remembered better than others?

These probably cannot be answered separately, if indeed they can be answered at all.

In general, the facts discovered in the previous section can serve as a starting point. It seems that subjects do not store a discourse (i.e., an expository or narrative paragraph) as a series of isolated sentences, or even as a series of connected sentences, but rather as some tightly interconnected unit.

Fillenbaum (1971) studied the effects of conjunction, and observed that subjects tend to retain the conjunction when it is salient to the story (for example, 'and' has more cohesive power when used in a temporal rather than a conjunctive sense). This, then, is evidence for the coherence mentioned earlier.

Clark and Clark (1968) also found that temporal ordering and causality play a significant part in the 'memorability' of sentences.

Lesgold (1972) performed similar experiments on coherency, and found that pronominalization is important in determining the unity of the text. Basically, use of pronouns to refer to previously-mentioned objects resulted in better recall of stories than use of definite noun phrases or other referring devices.

The variables mentioned so far deal with what we previously called the microstructure of the text -- the manner in which individual pieces are connected together. The work to this point has confirmed the assumptions made by linguists, such as Phillips, Grimes, and Schank, concerning connectivity (pp. 12-14).

We will deal next with the macrostructure -- the overall unity which makes a text cohesive.

One of the experiments in this area was performed by de Villiers (1974). He found that, when given a text, subjects recalled individual sentences according to their relative salience to the text (where 'salience' was subjectively determined). When given the same sentences in isolation, subjects tended to recall them based on their individual concreteness (i.e. how much of an image could be created). Interestingly, Sulin and Dooling (1974) and Meyer (1975) produced the same results in different ways.

Perhaps this can be construed as the first evidence for the hypothesis that a text is more than the sum of its component sentences. What remains, of course, is to get a better grasp of the notion of 'salience'.

A number of experimenters have worked on this in recent years. Probably the most precise was Kintsch (1974, 1975, 1976), who devised a propositional system of representation for meaning. The propositions could be embedded; i.e. the ones at the top level provided the central thread of the story, the next level down described the top level, and so on. Whether or not his system is correct (and there is much dispute), it at least provides one method of judging importance: the propositions higher in the tree are judged to be the most important. In his experiments, Kintsch found the expected results: higher-level propositions are remembered better. This confirms the theory that importance affects memorability.

These results were derived independently by Meyer (1975). Working from a model proposed by Grimes (but very similar to Kintsch's), Meyer produced the same results: higher-level propositions are recalled better.

Schank (1974b) developed an interesting model of the process of paragraph-understanding, which supported the same conclusions, although no experimental verification was provided.

Frederiksen (1975; 1976) devised a complex system of text representation, based on Halliday's work. Essentially, he identified six levels of information: concept, relational triples, system, proposition, relational system, dependency system. The dependency system is the highest level, embodying logical, temporal, and causal relations; it is similar to the top levels of both Meyer's and Kintsch's systems. Frederiksen again found that information at the highest level is recalled best. Interestingly, he also derived two other results:

- i) that subjects perform a certain amount of semantic and inferential processing at input time (rather than waiting until recall time);
- ii) that information already in the system affects the acquisition of later information; Frederiksen identified three acquisition methods: (1) selective processing (2) slot filling (3) superpropositional inferences.

Much evaluation of this work remains to be done.

Interestingly, the first experiment in the field was the only one which failed to produce the expected results. Crothers (1972) delineated an experimental method which has been used by all succeeding researchers:

- i) formulate a linguistic description of the structure of prose;
- ii) conduct recall experiments, analyzing both the passage and the recalls of it according to the theory developed in the first stage;
- iii) derive the empirical relation between structure and recall;
- iv) design a process model of memory to account for the features discovered.

Crothers' own experiments, however, produced results which contradicted his expectations -- primary subtrees (i.e. the top level) were not recalled any better than other levels. Crothers offered some explanations for his findings, and Meyer (1975) gives others. At any rate, it is still to be accounted for.

Perhaps this part can now be summarized. From the evidence given, it seems that text is stored in memory in a unified, holistic manner. Subjects are aware of this unity, and in fact it has a great deal of effect on their understanding. In particular, different parts of a text tend to be recalled to different degrees, depending on their importance.

III.4 Organizational Schemata

A related, though earlier, piece of work was Bartlett's (1932) study, Remembering. Bartlett used a long process of 'serial reproduction' (i.e., having stories told and retold through a chain of people), to observe the form that a text structure takes when allowed to adapt itself. The material he used came from a little-known Eskimo folktale, The War of the Ghosts. Bartlett observed several effects

i) omission
  -detail which was not salient to the story disappeared fairly quickly;

ii) rationalization
  -the story, being mythical, had a number of peculiar and inexplicable occurrences; subjects soon modified these, or invented events to account for them;

iii) transformation of detail
  -where names, locations, etc., were unfamiliar, subjects changed them into something more recognizable;

iv) reordering of events
  - the events tended to be rearranged, so that more important ones were given more prominence in the story;

v) bias toward the concrete
  - abstract concepts tended to be replaced by more concrete (i.e. more imagistic) ones.

In general, Bartlett observed that stories were radically modified, undergoing vast changes in the transition between hearing and telling. From this, he concluded that memory is an active rather than passive process -- i.e. it continually reorganizes its own contents. He borrowed the term 'schema', and described this as:

> 'an active organization of past reactions, or past experiences, which must always by supposed to be operating in any well-adapted organic response' (Bartlett, 1932, p. 201)

These schemata influence both our perception and recall, so that remembering, rather than being mere retrieval, takes on a constructivist nature.

Much of Bartlett's work is open to criticism, but the general tenor is probably valid: memory should be viewed as an active, self-organizing system, which mediates and rearranges its own contents.

III.5 Other Work

There still remain a few interesting results, which do not fit into the categories discussed so far.

Hunt and Poltrock (1974), working within the 'information-processing' approach, provided evidence for the existence of separate 'buffers' (memories). These were described as short-, intermediate-, and long-term memory. This work served essentially to provide the new paradigm with the same basis as the older, associative approach.

Kintsch (1975) derived some results which have direct bearing on A.I. work. He found that as the number of nominal concepts (i.e. things) in a text increased, the rate of processing was degraded. This corresponds to the standard A.I. notion of semantic memory search, wherein a greater number of nodes results in slower processing. In also relates to the role of participants in a discourse; a small number of participants, used over and over, is intuitively more cohesive than a larger cast.

This chapter has been a review of psychological work, to extract some common themes. In conclusion, we can identify a number of findings from psychology which are of relevance to work in natural language:

- i) People seem to process language with a 'chunking' approach; the input is organized into coherent units as soon as enough information is available.
- ii) Understanding is predictive; at each stage, we are expecting something to follow.
- iii) People use whatever information is available -- be it syntactic, semantic, or pragmatic -- to understand language.
- iv) What is remembered from a discourse is not any syntactic structure, surface or deep, but rather some highly abstracted meaning.
- v) Certain features tend to unify a discourse, including causal and temporal connectivity.
- vi) Some parts of a discourse -- notably the most salient -- are remembered better than others.
- vii) Memory is an active process, rather than a passive receptacle.

These conclusions should be kept in mind as we deal with the proposed model for analysis, discussed in the next two chapters.

IV. Computational Prerequisites

This chapter is an effort to establish the groundwork for a complete model for discourse analysis, to be presented in the next chapter. To achieve this, I shall develop some of the computational requirements of a discourse system, and present some potential solutions.

The chapter is divided into two parts. In the first, I discuss the question of representation, the various possibilities available, and the reasons for requiring an effective representation.

The second is a comprehensive overview of previous major analysis systems (at the level of single sentences). These are compared with respect to both good and bad features, and the motivation behind my particular choice (predictions) is explained. The prediction system is then described in detail, and an example of its operation is presented.

The intent here is to review the requisites of discourse analysis, and provide the computational tools necessary for the job.

Throughout this chapter, I will make reference to the psychological criteria just mentioned, indicating how the proposed system satisfies (or fails to satisfy) them.

IV.1 The Representation

As mentioned, one of the major questions in discourse analysis (and in any natural language work) is that of representation -- the form in which the input is finally stored.

To decide this, a number of questions must be answered:

i) What is the purpose of the representation -- is it for execution (Winograd), query (Woods, Petrick, Simmons), or just storage and examination (Rumelhart and Norman, Schank)?

ii) How broad a domain is to be dealt with -- blocks world (Winograd), closed data base (Woods), motivated humans (Schank), or something more general?

iii) How deep is the representation to go -- is it to be surface-oriented (Kintsch, Frederiksen, Simmons), conceptually deep (Schank), or something in between (Rumelhart and Norman, Miller, Wilks)?

With respect to analysis of narrative discourse, the questions can be answered:

i) mainly storage -- my intent here is just to show that a meaning can be represented, without specifying any particular use;

ii) relatively general -- except that the children's stories we will deal with usually feature predominantly concrete forms (i.e. one rarely encounters abstract topics);

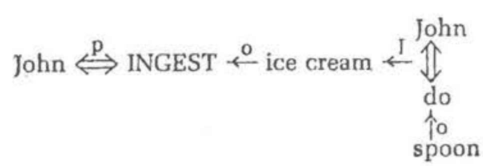
iii) probably deep - more will said about this later, but there are very good reasons for favouring a 'canonical form', if one can be found.

Given these criteria, the system I settled on was Schank's (1973; 1975) Conceptual Dependency. This effectively meets the goals we have established, and has a number of significant benefits.

Conceptual Dependency (CD) is an extension of the idea of dependency grammars (Hays, 1964), but is based upon the notion that there is a small number of 'primitive' acts (in this case, about 12), in terms of which everything can be expressed. This reduction to primitives greatly facilitates certain aspects of language analysis, since any two sentences which are paraphrases of each other are guaranteed to have the same underlying representation (one of Schank's fundamental assertions).

In addition to this taxonomy of verbs, CD also features a rigid set of cases. There are four (object, instrument, direction, and recipient), of which each verb must be associated with two or three (these are 'deep' or conceptual cases, not to be confused with Fillmore's (1968) more surface-oriented system). Another restriction is that the 'instrument' case, rather than being a simple nominal as in most systems, must be another conceptualization (i.e. another act). In addition to the cases, various modifiers, describing tense, location, manner, PART-OF, etc., are used. One complete group (i.e. actor, action, and associated cases) is referred to as a coneceptualization, and normally corresponds to on event.

Perhaps an example would help here. 'John ate the ice cream' would be represented as the conceptualization:



(Schank and Colby, 1973, p. 200)
Figure 2 - A Sample Conceptual Dependency Diagram

This has been a cursory overview of CD. Those unfamiliar with Schank's work, or wishing more information, are invited to read any of his many writings on the subject.

CD, as a representation system, has both good and bad points. Davidson (1976b) presents a comprehensive evaluation of the system. I shall indicate a few main points here.

First the advantages. Probably the most important (to Schank, at least) is that CD is (purportedly) <u>language-free</u>; that is, it represents information at a sufficiently deep level that it can be used with any language.

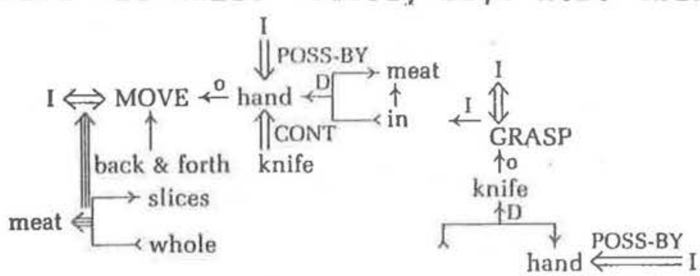
Another is the fact that the conceptually deep nature of CD has a certain amount of psychological verification. It satisfies point (iv) mentioned at the end of the previous section -- that the result of understanding language is some highly-abstracted semantic structure.

Third, and probably most important for our purposes, is the fact that the system is <u>computationally</u> valid. That is, despite the disputes regarding the effectiveness of CD from a linguistic point of view, computational linguistics (including the work described here) remains at such an unsophisticated stage that these subtle flaws are not crucial.

An interesting feature of CD is its ability to characterize the relative salience of different parts of a story. Schank (1974b, pp. 26-27) formulated quite specific rules for recognizing and representing the important points of a story. This fulfills psychological criterion (vi) -- that the importance of an event to the overall story affects how well it is remembered.

A final benefit of CD, related to the question of canonical form, is that it serves to capture regularities of language. That is, equivalence and similarities of meaning become much more obvious when everything is represented in a uniform formalism.

There are also some disadvantages of CD, which must be mentioned. First, there is the question of <u>accuracy</u>; much of the meaning of language (especially the more subtle nuances) seems to disappear in the transition into primitives. 'I sliced the meat with the knife' surely says more than:



(Schank and Colby, 1973, p. 228)
Figure 3 - Another Conceptual Dependency Diagram

Schank himself admits to this problem (1975a, p. 32):

> 'one must be careful not to lose information in a conceptual analysis (that is 'kiss' is more than just 'MOVE lips towards')'

However, as mentioned, we require only computational adequacy (i.e., the ability to function effectively within the restricted sphere of current computational linguistics work), and this has been achieved (as evidenced by the fact that the MARGIE system functions effectively).

Another problem lies with Schank's insistence on a relentless and immediate expansion into primitives. That is, every sentence must be expressed in terms of the primitive acts, with no opportunity to delay processing and wait for disambiguation. This is a fairly serious flaw in CD, and it results in some non-trivial difficulties. In our restricted domain, however, we will be able to avoid the problems occasioned by this approach.

Note that this latter difficulty contrasts with the 'wait-and-see' approach to analysis, mentioned earlier. Intuitively, humans do not rush headlong into interpretation, but make decisions only when they have sufficient information.

This concludes the commentary on CD in its standard form. For purposes of discourse analysis, I have made a number of minor modifications, mentioned next.

IV.1.1 Extensions to Conceptual Dependency

To handle the extended requirements of discourse, several minor changes have been made to CD. These are all ad hoc, and no theoretical motivation is claimed; the changes are made specifically for the goal of discourse analysis.

a) theme

One of the basic flaws in CD is its failure to retain any indication of the theme of a sentence; as with transformational grammar, the difference between passives and actives is ignored. This lack is barely noticeable at the level of single sentences, but becomes crucial when dealing with extended discourse. Since the theme can be detected in a straightforward manner (see 'staging', chapter II), it should be recognized and retained.

I have dealt with this in a simplistic manner; an extra case, marked 'theme', has been added to the standard conceptualization. This can point to any of the other elements of the conceptualization -- i.e. actor, action, object, instrument, recipient, or direction.

For example,

> John gave Mary the book.
> The book was given to Mary by John.
> Mary was given the book by John.

would all be represented by:

```
                   o           |-> Mary
John <=> ATRANS <-- book <-|
                               |-<
```

except that the theme would be the ACTOR, OBJECT, and RECIPIENT, respectively.

Interestingly, Halliday (1970) differentiated between psychological subject, grammatical subject, and logical subject -- corresponding roughly to theme, ACTOR, and surface subject.

Obviously, this ad hoc solution does not completely solve the problem, but it provides enough of a basis for us to continue.

b) 'then' links

Schank was one of the first researchers to examine connected discourse, and he did produce a coherent taxonomy of linkage between conceptualizations (Schank, 1974a). This set was based on causality, with nine different types of causation (including reason, enablement, and initiation) identified. Unfortunately, he stated later (Schank, 1974b, p. 16) that these types were the <u>only</u> connections between conceptualizaiton in a story.

This seems excessively restrictive; what is lacking is the simple notion of a 'then' link -- i.e. of one event just following another. Even Riesbeck (1974) recognized this, and Phillips (1975) arrived at the same conclusion independently.

This problem is another manifestation of Schank's insistence upon expansion into primitives. That is, even though it may be possible at some very deep conceptual level to identify causal links between all events, there are frequently times when we do not wish to force this degree of specificity. Schank recognized this, and attempted to amend it by providing for an 'unspecified causal connection', which is 'unexpanded and undirectional' (Schank, 1974b, p. 38). Intuitively, this is a 'then' connection.

The solution I propose here, as in the previous case, is simplistic; I have arbitrarily posited a 'then' link which can connect two conceptualizations, with the obvious meaning. This will be a sort of default link, in that it will be arbitrarily inserted whenever no other connection can be found. For the kinds of simple narratives we shall deal with, this assumption is probably valid.

As before, this is by no means a complete solution, but it is adequate for our purposes. Note that it reflects psychological criterion (v) -- that certain local linkages tend to unify a discourse.

c) other links

An effort of represent a discourse runs into problems when it is approached from a microstructural direction, as we have done so far. What is needed is some more global characterization of a discourse, which captures the overall picture.

The significant point for our work lies in the need to be able to represent a 'text grammar' in CD terms. For example, we must decide how to link the setting into the rest of the meaning structure. Is it a causal link? A 'then' link? Intuitively, neither of these will do, and additional representational structures will be needed. I will say no more about this here, preferring to develop things on an empirical basis, as needed.

d) unification of representation

Schank (1974b, p. 16) remarks that '[after a paragraph has been input] the conceptual dependency representation of each sentence is included [in the result]'. However, as Bransford and Franks (1972) showed, subjects often unify the memory into a more holistic mass, within which it is impossible to distinguish the original sentence. I have taken this finding as a guideline, and tried to integrate the meaning representation as much as possible.

To conclude the section on representation, this is an example of the meaning structure built from the short paragraph given in I.2 (pp. 10-11):

```
                         Mary <=> DO
                                A
                               ||| c
                        /      |||
                       /     dinner
          setting     /
                     /
                   |_
                     |>John  o                     o          |>oven
  Mary <=>MTRANS <-|       <-- John <=>PTRANS <--casserole <-|
                     |<                                        |<

                     o        |> Peter
  Mary <=> ATRANS <--salad <-|
                              |<
          A
          | then
          |
                     o         |> out-of (fridge)
  Mary <=> PTRANS <--dessert <-|
                               |< in (fridge)
          A
          | then
          |
               |->
     dinner <==|  ready
               |-<
             /
            /  denouement
           /
         |_
                           o              |>down  loc
 everybody <=> PROPEL <-- everybody <-|         <---table
```

|<

Figure 4 - Representation for a Short Paragraph

This representation is like standard CD except that the 'then' links are new, as are the ones labelled 'setting' and 'denouement'. For the sake of simplicity the theme has not been shown here; in this example, it is unimportant.

Thus we have a method of representing text. Note that the 'setting' link points to the entire centre part of the diagram. This reflects the fact that the statement 'Mary was making dinner.' holds true throughout the story, and serves, in a sense, as a 'grounding' for the events mentioned.

To more effectively verify this style of representation, we would have to show that it can be effectively *used* for different purposes: paraphrase, summary, etc. As mentioned, this is not one of my goals.

IV.2 *An Analysis system*

Now that a representation has been outlined, we can return to the question of a method of analysis. Essentially, the criteria for such a method are:

- i) the system must be *modular*, so that the various information sources can contribute when they are relevant, and lie dormant otherwise;
- ii) it must reflect the guidelines set in the previous chapter (on psychology);
- iii) it must be *flexible*, so that extensions can be made easily (this is related to the first criterion).

There are two options: either to take an existing sentence-level system, and extend it, or to write one from scratch. I have chosen the former, for a couple of reasons -- first, time constraints prohibit a complete design effort, and second, I believe that previous work in computational linguistics has produced some nice results, from which we can build.

Thus, I will first present a brief overview of existing analysis systems, explaining why I favour one such scheme (predictions), then (in the next chapter) outline the manner in which it could be extended to handle discourse.

IV.2.1 Comparison of Systems

Three systems will be dealt with here, as being the most advanced work in computational linguistics to date: Woods's (1968; 1970) Augmented Transition Nets (ATNs), Winograd's (1971; 1972) PROGRAMMAR, and Riesbeck's (1974; 1975) prediction-based system.

Woods's ATN system marked the beginning of what has been called the 'first generation' of computational linguistics work. ATNs have a number of strong advantages as a method of language analysis. First they are formal; the grammar and interpreter can be expressed in a rigid mathematical formalism, and the symbolic manipulations performed are well-specified. Second, they are simple; the relatively small set of operations makes design or interpretation of a grammar quite easy. Lastly, they are perspicuous; the passive nature of a grammar (it can be viewed as a state transition graph) facilitates rapid understanding of the actions and interactions of a given grammar.

There are, however, a number of drawbacks, many of which are prices paid for the advantages given. First, the ATNs are strictly syntax-driven; the semantics, if any, are added on as a set of Katz/Fodor-style set of restrictions. This is acceptable to a certain point, but current work in computational linguistics indicates that serious flaws exist. An interesting example is Jervis' (1974) implementation of an ATN; examination of her (large) grammar reveals that semantic checking is clearly divided from, and subservient to, the syntax, and also that such an approach leads to a somewhat unnatural structure.

Another drawback of ATNs is a certain degree of counterintuitiveness; these mathematical symbols and formal operations do not seem to reflect what we know about language comprehension. However, Kaplan (1972; 1975) has provided some interesting evidence to support the claim that ATNs are a valid model of human cognition and comprehension. The case is still open.

A third drawback to ATNs is their tremendous inflexibility. Arcs must be laid out beforehand, in a static manner, and cannot be changed dynamically during processing. (Scarl (1976) has developed a system with arc-moving capabilities, but this is at best a patch.) That is, there may be an arc which is taken in only one out of every hundred cases; this arc still must be represented in the grammar, although it is invalid 99 times out of 100 (the ordering of arcs at a node permits a certain amount of 'tuning', but this does not change the fundamental static nature of the system). The problems caused by this static organization will become apparent later, when we return to the question of discourse analysis.

The last flaw in the system, one which is currently out of favour in natural language circles, is its use of automatic backup. The system, upon failure, begins wildly undoing and revising its decisions, in an effort to find a valid parse. Scarl, in his system, provides a number of basic mechanisms to control the backup, but again the approach is wrong; backup should not be automatic. Rather, it should be designed by the programmer, so that:

i) it does not occur in all cases;

ii) when it does occur, the system can make an intelligent guess about where and why it went wrong, and act accordingly.

In summary, ATNs would seem to be a good mechanism for getting up a small grammar very quickly, and also a good pedagogical tool for an introduction to computational linguistics. They would not seem to be suitable for the kinds of large systems needed to handle discourse.

Winograd's PROGRAMMAR, designed at the same time as Woods's work, has a number of significant advantages.

First, and probably foremost, is the fact that the syntax/semantics/pragmatics distinction has been blurred (although not completely erased). Semantic checking is done at each step of the analysis, and pragmatic resolution (i.e. data base checking) is performed each time a complete component is built. This approach, revolutionary for its time, has since gained popularity in computational linguistics. (In fact, this method could be incorporated into an ATN system, but Woods's original work did not include this.)

Another interesting point, which was true as well of Woods's work, is the fact that SHRDLU's grammar is based on a well-specified linguistic system -- in this case Halliday's (1967; 1970) systemic grammar.

A minor advantage of PROGRAMMAR, which is still open to debate, is the fact that it is procedural, rather than declarative. This fact in itself is not significant, but one can describe benefits of both approaches (see Winograd, 1975, for a thorough discussion of this issue).

Last, and probably most significant in terms of advances over Woods's approach, is the power. A PROGRAMMAR program can be made arbitrarily complex, to do whatever actions are desired. True, an arc of an ATN can also have these actions added, but the procedural nature of PROGRAMMAR provides more expressive power.

So much for the advantages. The primary disadvantage of the PROGRAMMAR formalism is its unreadability. SHRDLU's grammar is so opaque and incomprehensible that Rubin (1973) found it necessary to try to flowchart it; even this was not easy. (Compare this to the simple transition diagrams in any of Woods's papers.) This problem stems directly from the procedural nature of the system -- nothing will 'hold still' long enough to pin it down.

Another drawback is the fact that the system is still static; the order and structure of the grammar are specified by the programmer during the design phase, and cannot be modified during execution (the system builds Microplanner programs 'on the fly'; unfortunately, it does not do the same with PROGRAMMAR programs).

A flaw shared with Woods's system is the heavy dependency on order. This is not merely the constraint that only grammatical sentences can be accepted, but also the fact that the interpretation of a sentence depends completely on the order of words in that sentence. Thus, for any given 'meaning' (i.e. deep sentence), all possible surface manifestations of that meaning must be accounted for.

Perhaps an example would help. If we wish to parse the sentence:

John ran down the street.

the appropriate ATN grammar would look like:

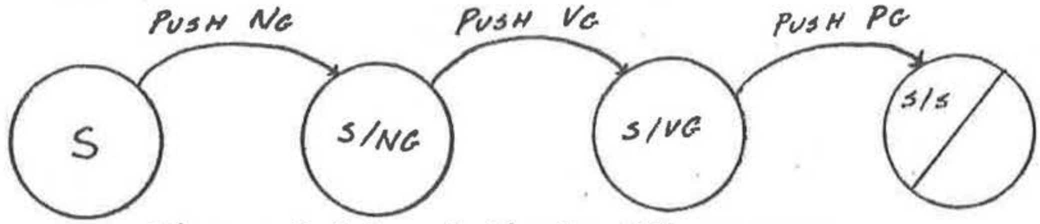


Figure 5 (a) - A Simple ATN grammar

(Note: the remarks about ATNs apply as well to PROGRAMMAR grammars.) Now, suppose the adverb 'quickly' were added to the sentence. It could appear in any of <u>four</u> places, and still mean essentially the same thing. But, to handle this in an ATN, we would need a grammar of the form:

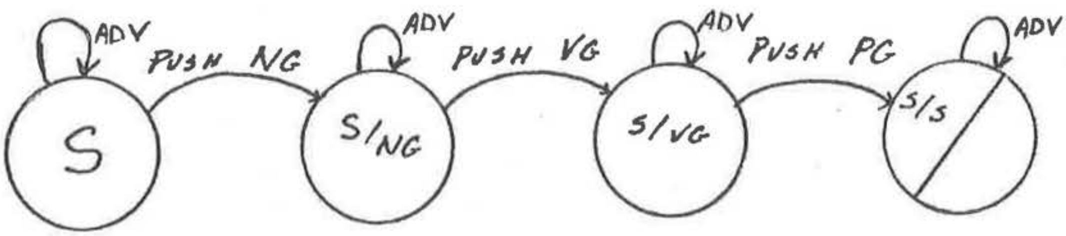


Figure 5 (b) - An Extended Grammar

i.e. with four separate checks for the adverb. My contention is that, since the adverb has the same meaning, and the same surface manifestation, it should be recognized by the same piece of the system. This will become clearer when we deal with predictions, below.

To conclude the discussion of PROGRAMMAR, it has one clear advantage over Woods's system -- expressive power (used here in the programming language sense of <u>convenience</u> of expression; the two systems are obviously formally equivalent) -- and one associated drawback -- lack of perspicuity.

Riesbeck's system of <u>predictions</u> is a new, and somewhat badly described, method of analysis. Basically, a sentence is analyzed via a set of predictions, spawned by the various words in the sentence. These predictions (or REQUESTS, as Riesbeck calls them), consist of two parts: a <u>test</u> i.e. a linguistic construct to be scanned for in the input) and an <u>action</u> (i.e. a set of functions to be performed if it is found. In this respect, the control structure is similar to that of <u>demons</u> (Charniak, 1972) or <u>production systems</u> (Newell, 1973). Again, readers wishing further details are directed to Riesbeck's thesis.

Predictions, like the other systems described, have their good and bad points.

The primary advantage of the prediction system is that it is not as dependent on order as the two previous systems. For instance, to handle the 'John ran down the street' example, we need only one simple prediction, of the form

((ADV) (RUMINATE MANNER))

(roughly translated, this means 'if an adverb is found, treat it as the MANNER of the sentence'.) This prediction would remain active throughout the entire sentence; thus (to return to our example), wherever in the sentence the adverb appeared, it would be picked up by this one prediction. This feature will be described in more detail in the next section.

Another point, which I have been stressing, is the fact that predictions are dynamic. The grammar is not specified beforehand (indeed, it is never really specified at all); rather, predictions are added and deleted from the central prediction list in a continuous process, so that the flow of control is never rigidly established. The advantages of this flexibility will become clear later, when we discuss the applications of this system to discourse.

A third advantage of the prediction system is that, to a certain extent, it is not as domain-dependent as the others. That is, both ATNs and PROGRAMMAR were originally aimed at particular domains, but the very nature of Conceptual Dependency required that the predictions be as general as possible. Whether or not this objective was achieved is questionable; at least the spirit is there.

Like Winograd's system, Riesbeck reduces the barrier between syntax and semantics. In this case, it is almost eliminated; predictions are semantically based, but can use syntactic information whenever it is helpful. Thus, the two subdomains have been more or less merged.

Riesbeck's system has a certain amount of psychological motivation, in that language comprehension is treated as a memory process (this feature was also present, implicitly, in Winograd's work). It satisfies psychological critera (ii) -- that analysis be predictive -- and (iii) -- that people use whatever type of information is available to understand language. Compare this to the normal 'competence' approach in linguistics, which is concerned with the speaker/hearer's ideal knowledge of his language, separate from any considerations of memory, and from any attempt to use the language (Chomsky, 1965, pp. 2-3).

A final advantage of Riesbeck's system, which was also present in the other two, is that it builds a form to be returned from the parse. In early computational linguistics work, it was assumed that the result of the parse would be a history of the steps taken in the parsing process. Woods changed this; in his system, only the components that are "BUILDQed" into the form are returned. In general, though, an

ATN parse is usually similar to the surface structure. Riesbeck carried this one step further; in the CD paradigm, the parse returned rarely, if ever, bears any resemblance to the original sentence. (Note that this is in many ways a result of the nature of Conceptual Dependency, rather than an inherent feature of the prediction process.) Again, this reflects psychological criterion (iv), concerning abstraction of meaning.

This concludes the list of the good points (in my opinion) of a predictive parsing mechanism. There are also some bad ones.

First, as a result of the magnificent and flexible nature of the prediction system, the grammar is totally incomprehensible. That is, since the system is so fluid, and since the set of predictions active at any one time is not fixed in any way, the grammar simply cannot be represented in any convenient way; even the designer of the system operates close to Winograd's 'complexity barrier' (Winograd, 1975a) when trying to understand the possible interaction.

Another flaw in prediction systems, which is at the same time perhaps a virtue, is their heavy bias towards semantics. The problem here is that it is actually difficult to express syntactic information. There are ways around this (these will be developed in the next section, when we look at predictions in more detail), but in general it's a non-trivial problem.

A third drawback is the problem of <u>control</u>. As Charniak (1972) found out with his work on demons, the method of spawning predictions can often be explosive, resulting in a large number of active predictions.

Thus, there are three major analysis systems available for extension to the discourse domain. I have chosen the last of these (predictions), for a number of reasons:

i) it is more powerful; as just described (and will be shown in the following section), information can in general be expressed more easily in a prediction style system than in any other;

ii) it corresponds most closely to what we discovered in the last section about psychology; interestingly, Riesbeck himself verified his theories only on the basis of introspection, but we have more concrete proof;

iii) it is more flexible; as explained, the dynamic nature of the grammar permits a great degree of modularity, and powerful interactions between the predictions.

Of the three, the last one is the major concern in designing a model for discourse analysis. The dynamic flexibility of the prediction system provides the control needed to handle a number of weak, but possibly relevant, I.S.'s.

Perhaps an analogy is useful here. Novice programmers, when learning a new programming language, tend to start with a basic subset of that language which, while not powerful computationally, is simple and easily understood. As they become more sophisticated, the programmers tend to make use of more esoteric constructs which, while providing a great deal of power, increase the complexity of the program, making it harder to understand.

Thus with natural language analysis. An ATN provides a clear, elegant formalism which is effective within a certain restricted approach to language. At the other extreme, a prediction-based system provides a great deal of power, at the cost of comprehensibility. More will be said about the power of predictions in later sections.

IV.2.2 The Prediction System

Having identified the prediction system, perhaps I should describe it in detail.

I see the basic mechanism of predictions as being language-independent, although the predictions themselves are obviously language-dependent. That is, the control structure described here could be applied intact to any language, but the surface level predictions would have to change.

Essentially, a <u>prediction</u> is a series of fields:

(TEST ACTION STRENGTH SOURCE CANCEL OBL?)

where:

test -is a specific surface-level construct this prediction is looking for.
 example: (NP ANIMATE)
  says to look for an animate noun phrase.

action -is the action to be performed upon finding the specified component; there are three main types of instructions:
 i) what to do with the input just found
 example: (RUMINATE OBJ)
  says to treat the component just found as the object;
 ii) what to do with the previously-built meaning base
 example: (REPLACE :SENTMEAN: (OBJ (EXTRACT ACTOR)))
  says to take the concept which was previously thought to be the actor, and put it into the object slot (this would occur, for example, in the analysis of a passive sentence);
 iii) what to look for next
 example: (PREDICT (PG TO) (RUMINATE RECIP))
  if it appears in the 'action' slot of a prediction, tells the system to spawn a new prediction, looking for a preposition group beginning with 'to', and if it is found, to place it in the 'recipient' slot (this would occur, for example, after the verb 'give' was discovered);
 Thus, the actions can encode the two types of information

available from I.S.'s -- what to look for next, and what to do with what we've already got.

strength -is an estimate of the 'importance' of the prediction (on a scale 0-10), assigned when it is created. The key here is that the predictions are kept in an ordered list, so that less important ones can be easily ignored (to a small degree, this corresponds to the ordering of arcs leaving a node in an ATN). It should be mentioned here that Riesbeck, in his system, kept the predictions unordered, to simulate parallelism; my goal in providing the ordering was to achieve an extra degree of control, not otherwise available.

source -is the name of the module which spawned this particular prediction; this information can often be useful in working with the prediction.

cancel -is the 'cancellation factor' for this prediction, decided either when the prediction was spawned, or when the prediction manager (q.v.) examined it. The cancellation factor indicates when the prediction should be deleted (i.e. deactivated), and can have three forms:

i) NIL - this prediction will never be cancelled;

ii) a GENSYMed atom -- if this prediction is successful, all predictions having the same value of CANCEL (including obviously the prediction itself) are deleted.
example: CAN157

iii) an arbitrary LISP predicate -- if this prediction is successful, all predictions satisfying this predicate (including possibly this prediction itself) are deleted.
example: (EQ (CAR (TEST PREDICTION)) 'PG)
will result in deleting all predictions for any sort of prepositional group.

obl? -is the 'obligatory' flag, set to T or NIL to indicate whether or not this prediction must be fulfilled to complete the current sentence. Essentially, this corresponds to the idea of obligatory and optional cases.

The important points are, obviously, test and action; the others are less significant, and will not be dealt with in detail.

The control structure of a prediction system is simple, but powerful. The work is done through the ordered list of predictions. At each step of the analysis, the system passes through this list in order. As soon as a prediction is found which succeeds (i.e. has its test satisfied), that prediction is allowed to execute its attached action. Then, the cycle repeats, checking for other satisfied predictions (note that the actions of the first prediction may have changed some global information, and caused later ones to succeed). When no further predictions can be satisfied, the next word of the input is accepted, and the process starts over.

Perhaps more should be said about the nature of predictions themselves. At the sentence level, I see predictions being spawned by

i) the words in a sentence (i.e. the current input);
ii) the conceptual structure built so far.

The predictions can be about three different <u>levels</u> (for the moment, at least):

i) <u>words</u> to be looked for in the input
ii) <u>concepts</u> (i.e. actions, noun groups) that might be referred to
iii) <u>conceptualizations</u> (i.e. full sentences) that might be seen yet.

(note that at the concept or conceptualization level, a prediction is essentially operating on the output of other predictions, in a manner similar to production rules.)

The question of levels is by no means solved; it will be further discussed later.

It is important to realize that judicious use of this sort of mechanism permits extremely powerful manipulation of the analysis process. For example, there is a function, EXPECT, which takes a list of components and builds the appropriate predictions for them. EXPECT takes three basic sorts of arguments:

i) AND - sequence
-causes a set of predictions to be set up to handle a linear sequence of the form indicated.
example: (EXPECT (AND (NG) (VG) (PG)))
would build a series of predictions in turn to handle, successively, a noun group, a verb group, and a preposition group (i.e., a simple declarative sentence). The equivalent form of an ATN would be:

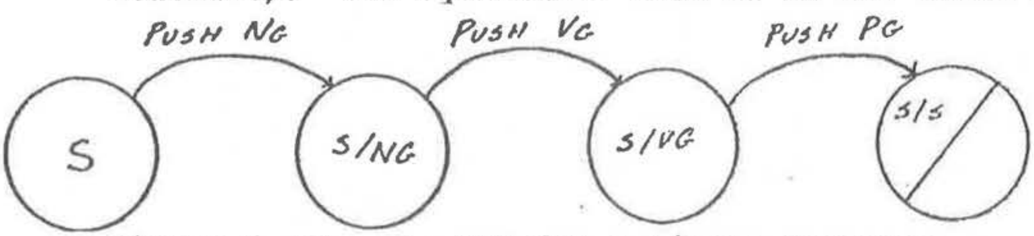


Figure 6 (a) - An ATN for a Linear Sequence

ii) XOR - exclusive or
-causes a set of predictions to be set up, so that if one succeeds, the others are removed
example: (EXPECT (XOR (NG ANIMATE) (PG TO ANIMATE)))
would set up two expectations, one looking for an animate noun group, the other looking for a preposition group beginning with 'to' and having an animate object of the preposition (this is obviously looking for the indirect object of a verb like 'give'). In ATN

formalism, this would be:

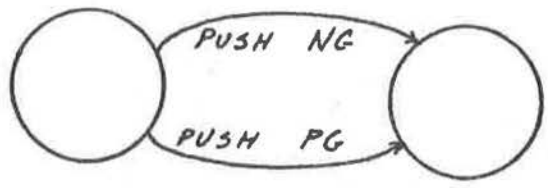


Figure 6 (b) - An ATN for an Exclusive Or

(with the appropriate restrictions on the arcs). Note, however, that in an ATN, these arcs would actually be on different nodes, and a certain amount of regularity would be hidden.

iii) OR - inclusive or
-causes a set of predictions to be spawned, looking for any member of the set, but making no effort to kill off other predictions in the set.
example: (EXPECT (PG LOCATIVE) (PG TEMPORAL))
would establish predictions looking for either a temporal or locative preposition group, <u>or</u> <u>both</u> (essentially a setting). This could <u>not</u> be represented easily in an ATN; the closest one could come is:

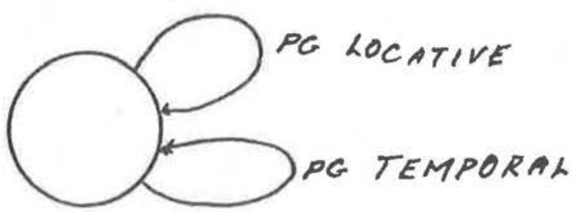


Figure 6 (c) - An ATN for an Inclusive Or

but there would have to be complex use of flags, if the number of choices was large.

Thus, a model using the prediction mechanism for its basic control structure is <u>at least</u> as powerful as the other system, and has some additional expressive capabilities.

One very instructive exercise is to try to model the various flows of control allowed by a prediction system, in a pattern-matching language like SNOBOL. This provides interesting proof of the power.

At this point I would like to explain the general orientation of the system. Essentially, I'm proceeding under the assumption that, in comprehending natural language, humans rarely back up; rather, at each stage they are <u>expecting</u> the next input, and know unambiguously what to do with it. This point was implicit in Riesbeck's work; it has been reified by Marcus (1975, p. 11):

'the structure of natural language provides enough and the right information to determine exactly what to do next at each point of a parse'

This will be referred to as the 'sufficient-information assumption'. My contention is that this is a valid model of natural language comprehension. To implement the sufficient-information assumption, we must recognize three points:

i) in some cases, a limited amount of lookahead will be required;

ii) since decisions are postponed as long as possible, there may be a large number of choices active in the system at any given time;

iii) there is obviously a great deal of information available from the structure of natural language, and from the surrounding context.

The first of these is obviously a reflection of Miller's work on chunking. If we restrict the amount of lookahead sufficiently well, our system will within his guidelines, satisfying psychological criterion (i), concerning chunking (pp. 46-47).

The second assertion, about the number of choices, will be recognized as an instance of Earley's (1970) parsing algorithm (with respect to which Marcus and Riesbeck have been somewhat lax in giving credit), as implemented by Fowler (1976). The essential point here is that decisions are postponed as long as possible; hence backup is rarely required.

Upon consideration of this point, it seems intuitively wrong (I say 'intuitively' because I have no psychological support in this regard). In parsing, humans do not seem to carry a large number of potential parses forward, eliminating the incorrect ones only as they are proved wrong. Nor, however, do they back up frequently. Rather, they seem to fasten very quickly upon a particular choice, which is almost invariably the right one.

The answer to this problem, it seems, lies in the third statement, concerning the 'sufficient information' which is provided by the content of natural language. It appears that there _is_ enough information available to enable us to predict with reasonable accuracy (as in Shannon's 'minimum entropy' discovery); the problem lies in recognizing it. My goal, therefore, has been to try to characterize the kinds of information available from natural language and use them in analysis. The identification of information sources mentioned in Chapter II, was a first step in this direction.

A related feature, which arose in the discussion on psychology, concerns the _direction_ of the analysis. Various psychological experiments, and the fact that we can comprehend spoken input, indicate that we do not wait for the end of a sentence before beginning processing; rather, we are processing the input as it comes in. Following this guideline, I intend that analysis should be 'always-forward' -- i.e. that the analyzer should have to look back at previous input as seldom as possible. Instead, it should check the information it has abstracted from the previous input, and which is currently providing guidance for the analysis process.

Perhaps an example would serve here. Most case-based systems extract the verb, as the central component of the sentence, then proceed to categorize the rest. More specifically, if a verb like 'grow' were found, the system would form a set of constraints on the subject (e.g. (MUST-BE ANIMATE)), then check back to see if these can be satisified. In my system, the subject (say, 'John') would be found first, and this would put constraints on the verb (e.g. (MUST-BE SUITABLE-FOR HUMAN)), which would then restrict the class of verbs to be found (i.e., a verb like 'exploded' would not be accepted).

This is a trivial example, used to emphasize what I consider to be the fundamental point of natural language: all the information available at any point in the analysis process must be used to restrict the choices at later steps. That is, at any point in a parse, we must have extracted the maximum information possible from the previous context, and this must restrict our present choices.

The idea of extracting the maximum information is not new; what is crucial here is my contention that there is enough information available to greatly reduce our possible choices. This goes back to Marcus' work and the sufficient-information assumption. In this respect my work differs from that of Riesbeck, who made a slightly weaker assertion concerning the 'always-forward' direction.

An example here will serve to indicate the potential benefits of this approach: I will compare the performance of my model to another, similar, system: Taylor's (Taylor and Rosenberg, 1975) case-driven parser.

Taylor's system was semantically oriented, and based on an extended taxonomy of cases. His parser operated by finding the main verb of a sentence (and sub-verbs, if any), obtaining the list of case slots for that verb, then moving around the sentence selecting components to fill these slots.

On correct sentences, our systems behave in a somewhat similar manner; the basic differences are not noticeable. The variations in approaches can more easily be seen by observing their behavior on incorrect sentences.

Take an example where a case is missing:

John put the ball.

Here, the missing component is obviously a location specification. Taylor's system and mine would discover the anomaly at the same time: when the end of the sentence was reached. In his system, the error would be manifest as an unfilled case slot(which was marked as being obligatory); in mine, it would show up as an unsatisfied prediction (also marked as obligatory). Thus, on this sentence, the results would be similar.

Now, however, look at an example with an extra component:

John came the book in the morning.

In this sentence, 'the book' is anomalous, and should be flagged as such by the analyzer.

In my system, the anomaly would be discovered as soon as 'the book' was encountered. That is, there would be a number of active predictions (many spawned by 'came'), all of which would reject 'the book' as a component. This would cause an immediate error (and possibly, although not likely, backup).

In Taylor's analyzer, the difficulty would be noticed only when all the rest of the sentence had been processed. The system would have filled all the slots for 'came' (in this case, only a subject and a temporal phrase), when it found that one component was still not accounted for. It would try to fit this component into a number of default slots, and only then signal an error.

If we introspect for a moment, it seems obvious that people do not use a complete top-down, slot-filling approach. Rather, they absorb the sentence one piece at a time, forcing each piece to fit into some place in the progressively-developing prospective meaning structure. Thus, in a sentence like the previous example, they would discover an error as soon as the anomalous component was heard, in the same manner as my system did.

This minor example was but a variation on a theme -- the theme of 'always-forward' parsing.

By following the 'sufficient-information assumption', and extracting the maximum amount of information from the previous context, we should be able to keep the number of predictions active at any one time down to a minimum. Obviously, much testing is needed here, to provide an empirical estimate of the number.

To conclude this description of the basic analysis mechanism, I will present a sample parse for a simple sentence. The sentence is 'John was coming home from school.', and it would be analyzed in the following manner (similar to Riesbeck, 1975, pp. 89-90):

word found	predictions active	predictions successful	action taken
-	1 - (NP)	-	-
John	1 - (NP)	1	(RUMINATE ACTOR)
was	2 - (VG)	-	-
coming	2 - (VG)	2	(SETQ :CONCEPT: '((ACTOR actor) (ACTION PTRANS) (OBJECT actor) (DIR (TO ?)(FROM ?))) (PREDICT #3) (PREDICT #4) (PREDICT #5)
home	3 - (ADV LOC) 4 - (PG TO) 5 - (PG FROM)	3	(RUMINATE (DIR TO)) (KILL #4)
from	5 - (PG FROM)	-	-
school	5 - (PG FROM)	5	(RUMINATE (DIR FROM))

Figure 7 (a) - Flow of Control in Parsing a Sentence

with the result, in Conceptual Dependency terms:

```
                                  |-> home
John <=> PTRANS <-- John <--|
                                  |-< school
```

Figure 7 (b) - Resulting Parse

Note that I have not bothered to show the tense in the diagram. Tense is easily recognized (both Woods and Winograd had sophisticated systems for discovering tense) and represented (Schank (1973, p. 206) presented a complete taxonomy of tense, for Conceptual Dependency), but much harder to <u>use</u>. In the simple stories with which we shall be concerned, tense will rarely be a factor; thus the tense markers will be omitted most of the time.

V. A Model for Discourse Analysis

Next, I shall return to the central goal of this thesis -- discourse analysis. In this chapter, I hope to unify what has been said previously, and build upon the work of the last chapter, to present a coherent model for the analysis of discourse.

I should state at the outset that many of the features here are by no means fully specified, even in my own mind; many of the ideas are at best inchoate notions about possible future directions. The intent here is to give a good indication of the problems involved in discourse analysis. To that end, I have presented the various stages of development, and the rationale behind each decision taken.

The chapter consists of four sections.

In the first, the prediction system is re-examined, this time with the intent of extending it to handle connected discourse. Various problems are discussed, including difficulties with top-down/bottom-up interaction, control of interaction of predictions, the question of levels of predictions, some basid bookkeeping work, and other more subtle features.

Next, I return to the set of Information Sources presented in chapter II. This time, I try to show how they fit into a discourse analysis system, and more importantly, what happens without them.

Third, a extended example is presented, detailing the steps taken at each stage of the analysis process, and showing how the whole system fits together.

Finally, the implementation itself is discussed. As mentioned, this is at an incomplete stage, and much remains to be done. I try to indicate what have been the most enlightening aspects of the implementation process, and where I feel the most difficult work remains.

V.1 Extending the Prediction System

There was a preliminary effort by Riesbeck (1974, Pt. II) to extend his system for the purpose of discourse analysis. This involved a somewhat vague notion of context, which included what I have called 'context' (foregrounding and frames) and 'real-world-knowledge' (inferencing), and a brief mention of style. However, the treatment was at best superficial and ad hoc, and no implementation was provided.

I would like to outline a possible extension to the predicti keeping in mind the various information sources that we have identified.

V.1.1 The Basic Control Structure

The I.S.'s available were found to have three characteristics:

i) weakness;
ii) potential lack of information;
iii) gratuitousness.

To handle this effectively, a modular system is needed. One possible structure is:

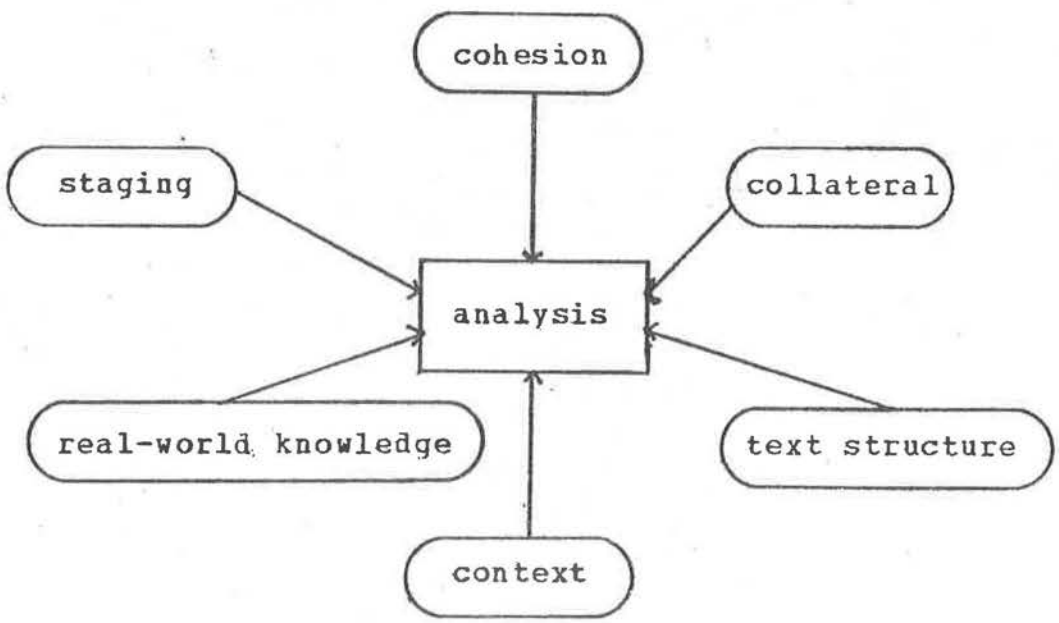


Figure 8 - Control Structure of the System

i.e., a number of independent modules, interacting through a central control system.

In light of the characteristics of prediction-based systems, perhaps this can be formalized. It seems that the central unit (here labelled 'analysis') should be an autonomous system, able to perform analysis by itself. This corresponds to the prediction model outlined in the previous section -- i.e. the system actually implemented by Riesbeck. This module will operate, as in the previous example, through the use of a prediction list. The only difference is that in this case the prediction list is global -- i.e. accessible to other modules.

The six peripheral pieces would each be separate modules, able to give information to the system only through the global prediction list (compare this to Hearsay's 'blackboard' (Reddy and Newell, 1974; Reddy, Erman, and Neely, 1973)). Thus, these modules would remain silent until a specific form was recognized in the input. At this time, they would contribute their knowledge by adding predictions to the list (hence helping to control the structure-building, as well). Under this scheme, some modules would be more active than others; for example, a text grammar might be interacting continually, whereas a 'collateral' module might run only once in a while.

V.1.2 Interaction of Predictions

Examination of this system, however, exposes a conceptual flaw. Predictions cannot be allowed to be completely oblivious of the other predictions in the list (as is now the case), although the notion is tempting. If this were so, we would eventually reach the stage of having several predictions <u>competing</u> for the same input, each prediction with a different action it wishes to perform.

For example, if we were following the story:

> John and Mary were playing baseball. John was angry at Mary. John ...

At this point, three of the modules would be appropriate:

- staging (theme): would say that we are discussing John and Mary, hence that 'John' is all right as the subject of the sentence;

- real-world (inferences): would have seen that John was angry at Mary, and inferred (inter alia) that John is liable to do something to Mary;

- context (frames): would still be in the baseball frame, and would therefore be look for some action from John relevant to baseball.

Thus, 'John' can be interpreted in two ways, depending on whether the inference module or the frame module is given more authority.

Of course, the decision could be postponed until the next component is found (and probably would be, in this case), but the underlying problem of competing predictions still exists.

The example is a trivial manifestation of a much deeper problem. The flaw is even more glaring in the case of two coinciding predictions. Suppose, for instance, that the inference module and the frame module are both expecting the same input, and intend to perform the same actions with it. The two predictions should interact, and, in some sense, reinforce each other, rather than trying to handle the situation independently.

This is a common A.I. problem: controlling the interaction of various sources (see, for example, Reddy and Newell (1974), Lenat (1975), Paxton and Robinson (1975), or Erman and Lesser (1975)). We want the I.S.'s to remain independent and local, and not require them to be aware of each other (or indeed of the rest of the system). At the same time, we must control their interaction in some way, for the reasons just mentioned.

The solution, it seems, lies in the uniform nature of the predictions. All work in the system is done through the global prediction list, and all elements of this list have the same form. Therefore, I propose to create a new, autonomous, module called the prediction manager. The prediction manager will scan the prediction list at each iteration, looking for cooperating and competing predictions, and other potential sources of trouble. It will have the power to make changes to the prediction list, by adding new predictions and deleting or modifying current ones.

The prediction manager, as described here, will obviously be a large module, with a lot of information and a corresponding amount of power. It will need to know about all types of predictions, about their associated tests and actions, and about the types of interactions which can occur between predictions.

Much of this information can only be gathered empirically, by running the system and observing the kinds of decisions demanded of the prediction manager. However, a couple of tasks can be identified, based only on theoretical requirements.

Given two cooperating predictions: the prediction manager should check to see that they really are looking for the same component, and for the same general reason. Then, it should merge the two predictions to produce a new one with the same test, a higher strength, and a combined list of actions. example: if the theme module had posted a prediction (note that irrelevant fields are omitted):

((NG) (SETQ :THEME: :COMPONENT:) 7)

(i.e. find a noun group, and assert that it is the theme), and the inferencer had posted:

((NG) (RUMINATE ACTOR) 7)

(i.e. find a noun group and assume that it is the subject of the conceptualization), the prediction manager should merge these, and produce:

```
((NG) (PROG (RUMINATE ACTOR)
            (SETQ :THEME: :COMPONENT:)) 9)
```

(i.e. a stronger prediction, but looking for the same thing, and performing the actions that would have been done by both the previous predictions). This reflects the intuitive notion that if two (or more) separate sources are expecting the same thing, we should favour that expectation.

Given two competing predictions: the prediction manager should attempt to defer a decision (i.e. keep both predictions active) until later information arises. If that is not possible, it will probably have to chose the one with the higher strength (remember, these strengths were assigned subjectively by the various creating modules), and kill the other, making a note of this fact, in case backup sould be required.
example: if, say, the inferencer (a marvelously erratic and inconsistent creature) had created two predictions:

```
((NG) (RUMINATE ACTOR) 7)
```

(i.e. if a noun group is found, make it the actor),

```
((NG) (RUMINATE OBJ) 5)
```

(i.e. if a noun group is found, make it the object), the prediction manager should first check to see if a noun group has been found; if it hasn't, no decision need be made. If it has, the prediction manager should let the first prediction succeed, (i.e. let the noun group be recorded as the actor), while deleting the second one from the prediction list, and making a note to that effect.

Obviously, more subtle kinds of interactions will be taking place, and the prediction manager will have to be much more sophisticated. The scheme presented here is just an outline.

Note that this sort of problem never arose in Riesbeck's system, because of its simplicity. He stated (Riesbeck, 1975, p. 103) that, by fiat, there would be no coinciding or competing predictions. This was quite reasonable, since his system only had one source of predictions -- the sentence-level syntax/semantics ones spawned by the various words in the sentence. Thus, there was never any danger of unforeseen interactions.

In the system described here, things aren't so smooth. The central analysis system (i.e. the part that Riesbeck included) is but one or <u>several</u> interacting sources. True, the central module is the strongest, but all work must be done through the prediction list, and any module can write into that if it feels so inclined. One module (the inferencer) can even create contradictory predictions by itself; the interaction between several modules is therefore complex.

This is also the strong point of the system. Since the method of communication is so uniform (and is the same one used by the central module for its own internal work), the peripheral modules can interact with the central section in a natural way.

The control structure of the system, with the prediction manager added, would probably now look like:

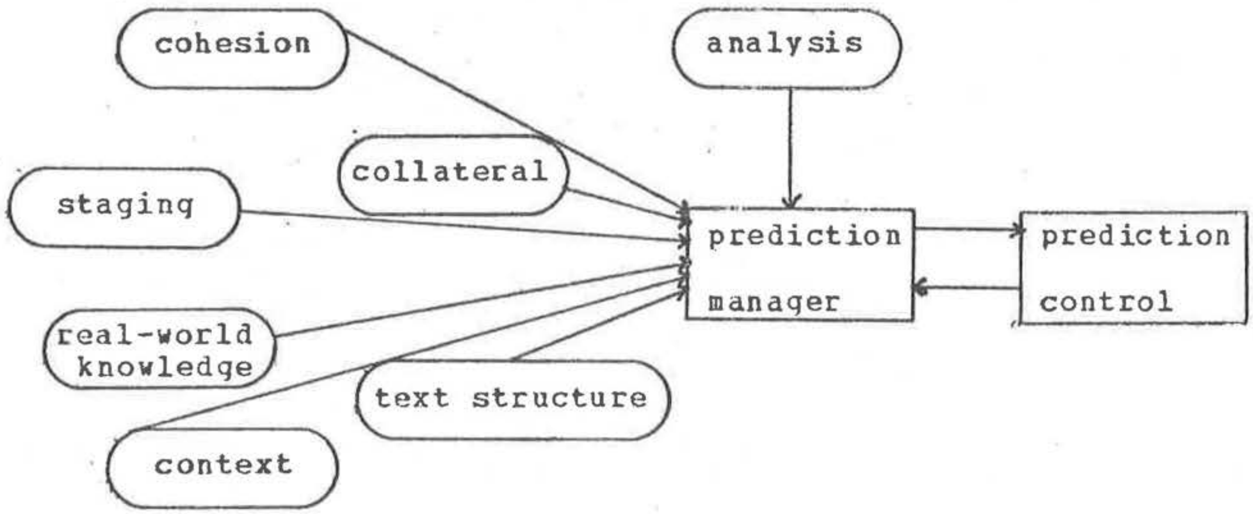


Figure 9 - Modified Control Structure

That is, the actual executive part of the system, which runs the predictions and performs the appropriate actions, has been removed from the 'analysis' module, and given its own autonomous location. All the other modules communicate with it via the prediction manager -- i.e., through the global prediction list. Note that the 'analysis' module (i.e. the autonomous sentence-level analyzer) is now the same as all the others; the only difference lies in the fact that it is stronger and more active.

This concludes the study of interactions. Suffice it to say that the prediction manager is an ill-specified piece of the system, which probably won't stabilize until after several iterations of implementation.

V.1.3 Levels of Predictions

Once the interaction of predictions is taken care of, some more basic questions must be asked about the predictions themselves. The problem in this case concerns the levels of a prediction -- i.e. how well-specified should the test part of a prediction be? Several levels can be identified, as mentioned previously:

i) words -- e.g. 'give','John';
ii) concepts -- e.g. 'noun group','prep-group to';
iii) conceptualizations -- e.g. 'a hit event','John doing

something to Mary'.

Ideally, we would like to specify things at the word level -- i.e., to predict sufficiently well from the context that we can guess the next word.

Unfortunately, such is seldom the case. As mentioned in the section on text grammars (although it is true of all I.S.'s), expectations generally come at a much higher level -- usually at the level of a conceptualization. One example, from the section on text grammars, is 'expect a setting', where it was shown that 'setting' could be manifest in several ways at a more detailed level. Essentially, what we were doing at that point was to convert requests from the conceptual level to the language level.

It is tempting to think that this can be done in general; that, for any high-level component, we merely specify all of its surface manifestations. This fails, however, because the situation is explosive. In reality, the predictions form a <u>fan</u>, in moving from higher levels to more specific forms.

```
conceptualization
          concept
             word
```

Thus, the prospect of 'pushing predictions forward' to the word level is just not viable.

An alternative approach would be to leave the predictions at a high level (say, the level of conceptualizations) and let the analysis proceed bottom-up until it reaches that level.

For example, in a text grammar, we might have an expectation 'look for an event' (where an event is a specific type of action). We could then let the analysis proceed bottom-up (i.e. using only local, sentence-level information), until a conceptualization was built. Then, this conceptualization is checked to see whether it is, in fact, an event.

Unfortunately, this approach is also not feasible. The problem is obvious -- after carefully extracting the maximum amount of information from the surrounding discourse, we ignore this information, failing to use it in parsing a sentence, and make use of it only once the sentence has been completely parsed. This seems counterintuitive, and also requires the ability to back up -- something we wanted to avoid as much as possible

For once our intuition is right. If the analyzer is turned loose 'bottom-up' on a sentence, it is subject to all of the contradictions and ambiguities which plaque natural language at the local level. Again, explosion occurs; this time, in the opposite direction.

```
conceptualization
          concept
             word
```

Perhaps the degree of bushiness is exaggerated. The point still remains: failure to make use of discourse level constraints when processing single sentences invalidates our method. Humans do not perform bottom-up analysis of sentences; why should our system?

The solution, it seems, is to pick a middle ground -- a level at which neither tree is too large. As the diagrams have indicated, the level I consider to be most effective, and which I have used in implementing the system, is the concept. This is essentially one element of a conceptualization (e.g. actor, action, object, etc.), and can be manifest in several ways (e.g. noun group, verb group, preposition group, adverb, etc.). As indicated, processing proceeds bottom-up (i.e. without constraints from higher levels) up to the concept level.

To implement this bottom-up work in a clean way, the basic transition net (BTN) was used.

The BTN is the same as Woods's ATN, except for two differences:

i) it is not augmented; no use is made of registers and arc tests;

ii) it is not recursive; no PUSH or POP work is allowed.

These restrictions were made to ensure that the bottom-up work was limited in power, and that no hidden processing was being done.

This approach is not new. Taylor, in his system, used a restricted ATN to enable him to identify the components of the sentence. Riesbeck, who claimed that no syntactic work was being done, actually had a primitive noun phrase recognizer built into his interpreter.

The work done by the BTN is not completely bottom-up; constraints (i.e. Katz/Fodor type selectional restrictions) can be passed down from the higher level whenever the BTN is invoked. For example, to look for an animate noun group, we might use the form:

```
(PARSE NG ANIMATE)
```

(where PARSE is the top-level call to the BTN). This would cause the feature 'animate' to be passed down, and it would be checked against the noun-group component built, just before the POP was executed.

This decision to arbitrarily posit the concept level as the point of intersection between top-down and bottom-up is simplistic, of course. In reality, this sort of interaction is going on at all times, on all levels (see Bobrow and Norman (1975), Palmer (1975), or Havens (1976), for interesting discussions on this point).

Given our decision to 'push' all predictions forward to at least the concept level, there is still a small amount of combinatorial explosion to be dealt with. This is the price paid for the transition from the higher levels down to the concept level. It is also a fundamental characteristic of language -- the number of surface manifestations of a particular construct is often large.

All of the work is done by a large, complex, function called SURFACE, which takes an expectation at a higher level, and converts it into a series of forms at the concept level, which can then be passed to the EXPECT function, mentioned earlier. For example, a call

```
(SURFACE SENTENCE DECLARATIVE)
```

would produce the linear sequence:

```
(AND (NG) (VG) (XOR (PG) (ADJ) (NG)))
```

(i.e. a somewhat simplistic syntactic structure for a sentence). Given

```
(SURFACE RECIP)
```

it would produce the set of appropriate formss for the 'recipient':

```
(XOR (PG TO) (NG ANIMATE)
```

Thus, the operation is fairly straightforward. Problems arise, however, when the arguments to SURFACE come from higher levels. For instance

```
(SURFACE EVENT)
```

means that a set of predictions characterizing an 'event' should be spawned. But in doing this, we run into combinatorial explosion; the distance between 'event' and the concept level is too great.

What I am saying here is that I have no definite answer to the problem, just an approach which works in simple cases.

An interesting idea here would be to use <u>predictions</u> to convert from higher levels to lower ones. That is, given the abstract form 'event', there might be a prediction which says 'an event can be manifest as either an action or a response'. In turn, another prediction might say 'an action can be manifest by a benefactive act or a destructive act', and so on. Essentially, these trees of predictions are performing the same work that SURFACE did, except that they are now under the control and cognition of the system; thus the explosive nature can be constrained. Again, this aspect of the system is remarkably similar to the production system method. This approach, while not fully developed, would seem to be the most promising.

Riesbeck, in his thesis, outlined a model of this nature, but it was restricted to linear chains rather than trees, and was heavily dependent on some of the 'slot-and-filler' aspects of conceptual dependency. At least it's a start, though.

V.1.4 Comparison with Production Systems

It is interesting, at this point, to re-examine the fundamental computational nature of prediction-based systems. Essentially, predictions consist of a left-hand-side (TEST), and a right-hand-side (ACTION). The control mechanism operates by cycling through the list of predictions, searching for one whose test succeeds, and executing the attached action.

This corresponds closely to the methodology of production systems. These were first proposed by Post (1943), as a general computational mechanism. In its simplest form, a production system (PS) consists of three components: a set of rules, a data base, and an interpreter. The rule set is ordered, and the data base is simply a collection of symbols. The interpreter operates by searching the rule set until one is found whose left-hand-side (LHS) can be matched against the data base. When one is found, the right-hand-side (RHS) of the appropriate rule is inserted into the data base in place of the pattern which was matched (i.e., the LHS), and the cycle continues.

In more recent work, the structure of production systems has been extended. The LHS can be an arbitrary form, which is evaluated, and the RHS is an action, which is executed (possibly causing side effects). The data base is no longer a simple collection of symbols, but can now include such aspects as perception, although it is still completely global. This style of PS will be referred to as pure production systems (Davis and King, 1975).

At this point, production systems and prediction-based systems would seem to be similar (note that 'rules' in one system are 'predictions' in the other); a comparison is in order.

To provide this we must first identify the characteristics of 'pure' production systems, then verify whether they apply as well to our prediction system.

Pure PSs have the following features (taken from Davis and King, 1975):

i) restrictions on interactions between the rules
   In pure PSs, the only interaction allowed between the various system modules is via the global data base. This tends to preserve independence of knowledge sources, at the expense of explicitness in the control structure.

ii) constrained format of rules
   The LHS can be a Boolean combination of simple predicates; the RHS is limited to 'conceptually simple' operations.

iii) modularity
   This is a byproduct of feature (i); the control flow is completely decoupled from the actual rules.

iv) opacity
   This is an unfortunate effect of the decoupling just mentioned -- the system is difficult to understand.

v) second-order understanding

> Essentially, a pure PS, because of the rigid formalism into which its rules have been forced, should be capable of 'introspecting', and examining its own rules.

This is a relatively general description of a pure PS; it is obvious that the prediction system we have just outlined does not fit completely into this class. More important, however, is the fact that most of the production systems actually implemented also differ in various ways from these guidelines. In discussing our prediction-based system, I shall try to point out the parallels with current PSs.

One of the major differences between our system and pure PSs is the interaction of rules. I found it necessary to create a prediction manager, with the ability to examine and manipulate predictions, to handle possible interactions (conflicting and cooperating). Interestingly the DENDRAL system (Feigenbaum, 1971) also faced the problem of conflict of rules; it was dealt with, in a manner similar to my prediction manager, through a system of rule precedence (i.e., strength). In LISP70 (Tesler et al, 1973), conflicts were resolved by chosing the most specific rule.

Another difference concerns the constrained form of rules. In our system, the LHS is quite simple, as required (this is not always the case; ACT (Anderson, 1976) has a complex node-matching scheme, and DENDRAL permits not only complex matches, but also side effects in the match). The RHS, however, is more complex; the structure-manipulating and prediction-building operations specified could hardly be called 'conceptually simple'. Again, there is a precedent; in the PASII system (Waterman, 1974), the RHS can specify operations to construct new productions.

The question of second-order understanding (i.e. meta-predictions) has not been treated in detail in our system, although the prediction manager obviously requires a certain amount of this.

The characteristics of modularity and opacity are observable in the prediction system, as in pure PSs.

Thus, it seems that a prediction-based system, while not corresponding completely to pure PSs, is similar in many ways to the PSs actually implemented (perhaps these should be called 'extended PSs').

One difference, however, remains; this concerns the globality of rules. Under the pure PS paradigm (and in most of the extended systems), the rules are explicitly global. That is, they are active at all times, and only the ordering (together, of course, with the data base match) determines which rules may execute. This often leads to use of complex 'tags', simply to block the execution of a particular rule.

In the prediction system, the opposite approach is taken. Much effort is expended to ensure that the set of predictions active at any one time is 'relevant' -- that is, of interest to the current situation. The system of cancellation markers ensures that predictions are deleted as soon as they are no longer valid.

This approach has also been used in certain production systems. In particular, Moran (1973) and Rowat (1976) outlined methods of 'grouping' rules, so that larger amounts of knowledge can be manipulated. MYCIN (Davis et al, 1975) and DENDRAL have similar mechanisms. In general, however, this is not emphasized in production systems.

In conclusion, I feel that, while our system is similar to PSs, there is a minor difference in approaches. In the prediction system, the data base is relatively simple, and the situational knowledge resides in the set of predictions; in a PS, the converse is true. Thus, although the prediction system is a form of PS, and the goal is the same (i.e., dynamic control), I shall continue to deal with it as a different mechanism.

V.2 Use of Information Sources

In this section, we will deal again with the information sources developed in Chapter II, this time with an eye to incorporating them into a prediction-based system.

V.2.1 Preliminary Requirements

In chapter II, we described two things about each I.S.:

i) its surface manifestation;
ii) the information it tells us.

The problem here will be to encode these into predictions, so that they will fit into the system outlined previously.

Number (i) is relatively straightforward, since recognition of surface-structure phenomena has already been fully specified, and is a standard aspect of computational linguistics. This corresponds to the 'test' part of a prediction.

Number (ii) might be harder. Intuitively, the information from an I.S. will have to be encoded as the 'action' part of predictions. Upon examination, I.S.'s are seen to carry two types of information:

i) what to look for next;
ii) what to do with what's already been found (i.e., how to organize the representation).

But, this list corresponds almost exactly to the possible actions available from predictions (IV.2.2, p. ?)! The problem is almost solved.

What is missing, and what was skipped in the earlier discussion of predictions, is some indication about the <u>details</u> of these actions -- i.e. how would they be represented in LISP code.

Number (i) above -- what to look for next -- is obviously well suited to a prediction based system. One LISP function, PREDICT, is required. PREDICT takes two arguments, a test and an action (plus a number of optional arguments), and posts the appropriate prediction.

Number (ii) -- organizing what's already been found -- is more complex. Intuitively, what we want here is a set of structure-manipulating operations to permit us to organize the information in the desired way. These operations will depend on the level of the representation, the structure of the representation, and various other things. Currently, a simple set exists:

(RUMINATE slot)
- takes the component just found, and tries to put it into the 'slot' in the current sentence. It does this in an 'intelligent' way, in that it checks to see
  - i) if the slot is already filled;
  - ii) if filling this slot completes the conceptualization;
  - iii) if any side effects should take place (as, for example, when the ACTION slot is filled).

(REPLACE conceptualization (slot filler))
- takes the filler, and tries to put it into the appropriate slot in the appropriate conceptualization. Essentially, this function is used to rearrange previously-built structure. It is different from RUMINATE, in that
  - i) the filler can be arbitrary, and not just the component most recently found;
  - ii) the conceptualization into which it is placed is not restricted to the current one, but can be any previous conceptualization in the meaning structure;
  - iii) no checking is done; REPLACE assumes that whoever called knows what is going on.

(EXTRACT slot <conceptualization>)
- returns the contents of the given slot in the given conceptualization (which defaults to the current conceptualization, if omitted).

Thus, the example given earlier

(REPLACE :SENTMEAN: (OBJ (EXTRACT ACTOR)))

says to take the concept which is in the ACTOR slot of the current conceptualization, and put it into the object slot.

Besides these three functions, there are a number of global variables (:SENTMEAN:, :CONCEPT:, :CLAUSE:, :COMPONENT:, etc.) where the meaning is apparent. There are also additional functions whose effects should be equally apparent.

With these functions, we are able to perform the manipulations required to use the I.S.'s effectively.

V.2.2 Information Sources

In dealing with the I.S.'s described here, it must be understood that they are all, in some sense, <u>optional</u>; without them, analysis could still proceed, although not as effectively. Thus, in developing this chapter, I shall emphasize the <u>added</u> benefits bought by these sources, by giving an indication of what the effects would be without them. In the examples given here, I will assume that we are dealing with a system in which all the modules (I.S.'s) are functioning and generating predictions, although I shall emphasize only one module at a time.

<u>staging</u>:

It will be remembered that the major point here is theme. An example of the use of this in analysis would be:

Mr. Smith's window was broken by the flying ball.

The presence of the theme should be flagged, in this case, as a side effect of the prediction which recognizes passive sentences:

```
((VERB PASTPART)
       (PROG (REPLACE :SENTMEAN: (OBJ (EXTRACT ACTOR))
             (SETTHEME OBJ)))
```

That is, in the process of rearranging the subject and object components, the theme is also flagged.

A prediction which makes use of this information might look like:

```
((THEME)  (EXPECT (MORE-ABOUT THEME)))
```

which, in the case of the example sentence given, would produce expectations like:

-expect to find out more about Mr. Smith's window
-expect Mr. Smith to be angry

Without some notion of theme, this sentence would be treated in the same manner as its active equivalent; if this were the case, the set of expectations following the sentence might be:

-expect more about the ball
-expect the baseball game to continue

Thus, use of thematic information provides the ability to 'focus' on a part of the discourse. Other uses might include the ability to detect a change of theme; this usually indicates a <u>transition</u> in the focus, which might be significant.

<u>cohesion</u>:

The important aspects here are information blocks and referential specificity.

Information blocks, as mentioned, are usually signalled by punctuation or grammatical arrangement. Compare:

(a) Peter was running home, and he fell down.
(b) Peter was running home. He fell down.
(c) When Peter was running home, he fell down.

In this example, the act of falling is emphasized, to a successively greater degree, in each of the three cases. Intuitively, we would like the representation to be different in each of these three cases.

(a)

```
                o         |>home                         o         |>down
Peter<=>PTRANS <--Peter <-|             Peter<=>PROPEL <--Peter <-|
                          |<                                       |<
```

(b)

```
                      o          |> home
       Peter <=> PTRANS <-- Peter <-|
                 A               |<
                 | then
                 |
                    o         |> down
       Peter<=>PROPEL <--Peter <-|
                              |<
```

(c)

```
               o        |>down   time                    o        |>home
Peter<=>PROPEL<--Peter<-|       <----- Peter<=>PTRANS<--Peter<-|
                        |<                                      |<
```

That is, the representation should somehow capture the relative salience of the two sections, in particular the one concerning falling.

Predictions to encode this knowledge might look like:

```
    ((WORD AND)
         (PREDICT ((SENTENCE)
                   (SETQ :CONCEPT: (:OLDSENT:  :NEWSENT:)))))
```

i.e., if the word 'and' is found, look for a second sentence to go with the first, and produce as the meaning the ANDed conjunction of the individual sentences.

```
((SENTENCE)
                                          then
      (SETQ :CONCEPT: (:CONCEPT: <-- :SENTMEAN:)))
```

i.e., if two sentences are found without any other linkage between them assume a default 'then' link, and join the new sentence into the previously built structure. This corresponds to case (b) above, where the use of separate sentences for the two sections serves to emphasize both of them.

```
((CLAUSE TEMPORAL) (REPLACE :SENTMEAN: (TIME :CLAUSE:)))
```

i.e., if a temporal clause is found, put it into a 'time' slot (thereby reducing its importance).

The rules here are obviously somewhat simplistic; they are not intended to be complete, but rather to illustrate the possible use of information blocks.

Without the sort of information specified here, a system would miss the essential differences in importance; probably, all the cases would produce the same representation:

(a)

```
                  o         |> home                    o         |> down
Peter<=>PTRANS<--Peter<-|              Peter<=>PROPEL<--Peter<-|
                            |<                                   |<
```

Many systems do make use of this sort of information. Unfortunately, their work is implicit, in that rules concerning information injection are not explicitly stated. I feel that to function effectively, these rules must be recognized by the system designer, and be encoded as such.

Referential specificity is somewhat harder to deal with. A sample of the problem can be seen in the sentence pairs:

(a) John hit the ball hard. It flew in a long, high arc.
(b) John hit the ball hard. The ball flew in a long, high arc.

What we want here is the ability to detect the fact that the repetition of the definite noun phrase in (b) is mildly unusual, and that it possibly refers to a different ball (or that the story is aimed at a younger audience).

This sort of information is not easily represented by predictions. Rather, I would envision a separate reference resolution program, which is called upon to handle any anaphora. This program would have two sections to turn to for guidance:

i) a list of the concepts currently in working memory, from which the current referent should be chosen;
ii) a list of pronominalization rules for discourse, used to tell whether the current reference is over- or under-specified (a sample of these rules is contained in Charniak (1972), as borrowed from Lees and Klima

(1963)).

Obviously more work needs to be done here. In particular, it's not clear what an overspecified referent tells us, or more importantly, how to use that information.

<u>collateral</u>:

Some forms of this are easier to use than others. One example is 'alternative':

> John could have been safe, but he didn't run quickly enough.

where the negating conjunction serves to emphasize the (unstated) fact that John was out. One simplistic way to handle this is to have it flagged by the prediction which recognizes 'but':

```
((WORD BUT)  (ASSERT (NEG :SENTMEAN:)))
```

i.e., if the word 'but' is found, assert the negation of the first clause (in this case, that John was not safe). Again, this is obviously a restricted approach; among other things, it fails to differentiate between different meanings of 'but'. However, these are essentially bookkeeping details, and could be worked out without too much difficulty.

Note, incidentally, that the content of the second clause is totally irrelevant here. That is, once we see:

> John could have been safe, but ....

we know that he was out, and can assert such without waiting for the rest of the sentence.

Without this capability to use alternatives, the system might never realize the implicit meaning of the sentence. In the example here, if the fact that John was out is not explicitly mentioned elsewhere, the system would never know this -- an obvious omission.

Another example of collateral is foreshadowing:

> The boys were playing ball dangerously close to Mr. Smith's house.

where the word 'dangerously' hints that possible trouble is ahead (this might also be called 'evaluative'). A prediction to handle this might be:

```
((ADV EVALUATIVE)  (EXPECT VALUE))
```

which in this case, would produce an expectation like:

-expect something bad

which, once the inferencer got through with it, might be

-expect Mr. Smith's window to be broken

Without this particular feature (i.e. recognition of foreshadowing), the system would still perform the analysis, but in a different manner. The expectation generated would be:

-expect the baseball game to continue

Thus, the breaking of the window, if indeed it happened, would come as a surprise, and some extra processing would be required to handle it.

Obviously, humans would detect a loaded word such as 'dangerously'; the system should do the same.

text structure:

This has been discussed thoroughly in a number of places; what remains is to show how it can be incorporated into a prediction system.

Given the fairly simple grammar mentioned previously:

```
story --> setting + episode + denouement
setting --> temporal location |
            spatial location |
            (character)* |
            continuing event
episode --> event |
            action + reaction
```

The analysis of a story might begin with

-expect a setting

which would be converted into the call

```
(EXPECT (OR (LOCATION TEMPORAL)
            (LOCATION SPATIAL)
            (CHARACTER)
            (EVENT CONTINUOUS)))
```

Thus, the sentence

John and Peter were coming home from school.

besides satisfying all the local (i.e. sentence-level) predictions, would also satisfy the expectation for a setting, and hence would produce

-expect an episode

as the text grammar continues. This in turn would generate the call

```
(EXPECT (XOR (EVENT) (AND (ACTION) (REACTION))))
```

and the cycle repeats.

The most significant benefit of this is that if a story is finally recognized (i.e. the expectation 'expect a story' is satisfied), the meaning of the story can be organized and manipulated in whatever way is desired. Essentially, we have captured the ability to deal with the story at the macrostructural level, and thus to impose our own organization on it. This corresponds to the 'semantics' of the text grammar.

Without the use of text grammars, two flaws occur in the system. First, the expectations will be less directed, since the system has limited knowledge of how events should follow each other. This is a comparatively minor difficulty; the system can get by without the added expectations the text grammar provides.

Second, and more importantly, the system will have no idea of what constitutes a coherent story. It will be able to deal with the linkages at a local level (i.e. between pairs of sentences: causal, temporal, etc.), but will not be able to treat a text as a unified whole. The current work with scripts, etc., is an obvious attempt to remedy this.

<u>context</u>:

Again, there are two relevant aspects: foregrounding and frames.

Foregrounding is essentially a superset of the problem of reference (mentioned under 'cohesion'). As I see it, foregrounding should not be a part of the prediction system itself, but would have its own module, at the 'meta-prediction' level. Its task would be to foreground the desired concepts; as mentioned a set of rules can be delineated specifying when and how a concept is to be foregrounded and backgrounded.

Foregrounding of itself serves no purpose, except to move concepts in and out of the (bounded) working memory. This working memory is used by other modules such as reference resolution.

An example would be:

There's a ball in the yard. It's green.

where the foregrounding routine will recognize the instantiation of a new concept (the ball), and 'activate' that concept. Thus, when reference resolution is required (as in the second sentence), the resolution routines will have a easier job deciding what 'it' refers to (note also that the 'theme' module would probably also help here, by indicating that 'the ball' is the theme of the first sentence).

Without this foregrounding capability, the system loses a lot of the power which humans use so effectively in processing language. That is, without an effective representation of what is 'on stage' at the present moment, we are faced with potentially explosive searches in various phases of the processing.

Frames, in the situational sense, provide similar power. Like foregrounding, the frame instantiation mechanism would operate at the meta-prediction level, in that frame recognition and instantiation would be performed by a separate module, outside the normal prediction system. Unlike foregrounding, however, the information that the frame carries would be injected into the normal prediction list. For example, when the frame for 'bicycle' is instantiated, a series of expectations might be spawned, corresponding to our knowledge about the structure and possible uses of a bike:

-expect to hear more about the bike itself
-expect to hear about the handlebars
-expect to hear about the fenders
....
-expect to hear about someone riding the bike
-expect to hear about someone locking the bike
....

Thus, a combination such as:

> Jane was riding her new bicycle. The fenders were a pretty shade of red.

could be handled, since there would exist an expectation for fenders, which would resolve them as part of the bicycle.

This use of uncontrolled forward inferencing is obviously explosive; perhaps better power could be attained through an inference-on-demand (i.e. deep binding) approach. For the moment, I'll let things stand.

The need for a frame system is obvious. Without it, the sample sentences given above could hardly be handled. True, we could trigger a search through the semantic net, which would eventually discover that fenders are PART-OF a bicycle; however, this approach would be even more explosive than the frame method, and, besides, would miss the essential fact that humans have their knowledge 'clustered' into useful chunks (see Scragg, 1976, for some interesting comments on this).

Note, incidentally, that the general notion of frames enables us to satisfy the last of the psychological criteria -- that memory be an active, self-organizing process. Minsky's frames are based very heavily on Bartlett's schemata, so the correspondence is no accident.

real-world knowledge:

This module has played a large role in the examples to this point; it's time to explain how things work.

Essentially, I see inferences as being verb-driven; that is, the set of possible inferences would be keyed by the surface-structure verbs found in each sentence. This set of inferences would be loaded into working memory, hence would be subject to the size constraints mentioned previously. Thus, we have an upper bound on the number of possible inferences active at any one time (this should reduce the problem of combinatorial explosion, although it will not eliminate it).

Inferences themselves will be of the form:

(TEMPLATE ACTION)

where

template - is the pattern to be matched in the current memory (and the match can be quite complex)

action -is the inference to be made if the template can be matched.

For example: attached to the verb 'go', we might find the inference

```
                                      |-> Y
      (   ( X <=> PTRANS <--X <-|          )      ( X<=> loc(Y) )  )
                                      |-<
```

i.e., if we find that person X is going to location Y, infer that later on X might be <u>at</u> location Y.

Thus, if we receive the input

John was going to school.

we might make the inference that John is at school.

Note that inferences, as described, are similar to predictions. Actually, they work at a somewhat higher lever, and have different characteristics.

Essentially, inferences provide two types of information. First, there is factual information that might be required for later deductions. This is the type shown above, where the fact that John is at school might never be explicitly stated, but might be needed for later processing.

The other type of information concerns expectations about what might be seen next. This was illustrated earlier, where the sentence 'John hated Mary' produced the inference that John might do something to harm Mary. This is probably the more important of the two types, since it provides direction to guide the analysis of later input. That is, it tells us not only what to expect next, but also what to do when we find it.

Again, this reflects what humans seem to do. Intuitively, they perform a certain amount of inferencing, in preparation for whatever follows.

Without this sort of capability, the system is obviously restricted, since it has no way of fully comprehending the meaning of the sentence, or of discovering logical implications. One example: causal links are very seldom stated explicitly, and must usually be inferred; without some inference capability, the system will be unable to connect events together.

The approach here obviously does not solve the inference problem; in particular, the danger of explosion still remains. Some controlling mechanisms to prevent this have been suggested; perhaps others would be needed. For the moment, we'll let things stand.

This concludes the section on Information Sources. Of the six covered, it will be noted that the first three are 'stylistic', and the last three 'semantic'. Essentially, they are manifest differently in discourse, although their function and meaning are essentially the same. This chapter has shown that a single, uniform method -- predictions -- can be used to encode diverse types of information.

V.3 A Detailed Example

To conclude the chapter, we will follow the analysis of one complete paragraph-length story. The story we shall be concerned with is:

> John and Peter were coming home from school. As they were walking through the playground, they saw an animal in the grass. It was a rabbit. The boys chased after it, but it hopped away. They went home.

We will present the analysis here in a manner similar to that used for the single-sentence example shown earlier, except that fewer details will be given.

Assume that all of the I.S.'s mentioned earlier are available, and that the text grammar we are using is the one specified before.

At the start, we have the standard default prediction

(a) ((NG) (RUMINATE SUBJ))

plus the four predictions spawned by the expectation for setting.

(b) ((LOCATION TEMPORAL) (RECORD SETTING))
(c) ((LOCATION SPATIAL) (RECORD SETTING))
(d) ((CHARACTER) (RECORD SETTING))
(e) ((ACTION CONTINUING) (RECORD SETTING))

The prediction manager looks these over, and discovers that two of them (the default one for NG, and the one for characters) are esentially looking for the same thing. It merges these, producing a new prediction, with a higher strength (not shown):

(f) ((NG) (PROG (RECORD SETTING) (RUMINATE SUBJ)))

The predictions active are now (b), (c), (d), (e), and (f). Note that (b), because of its particular style of cancellation, has not been killed off by the merger.

The phrase John and Peter is now found, satisfying (f), which removes itself, but spawns a new prediction:

(g) ((VG) (RUMINATE ACTION))

Again the prediction manager recognizes this, and merges it with (c), to form

(h) ((VG) (PROG (RUMINATE ACTION) (RECORD SETTING)))

Thus, analysis proceeds through the sentence, in a more-or-less straightforward manner. At the end of the sentence, the only predictions active are (b), (c), (d), and (e). However, the fact that the expectation for 'setting' has been satisfied causes all of these to be removed.

At this point, some other modules begin to contribute. Since the setting has been satisfied, the text grammar recognizes this, and an expectation for an 'episode' is generated. This translates at the lower level to a prediction for an event, or an action/reaction pair:

(h) ((EVENT) (RECORD EPISODE))
(i) ((ACTION) (PREDICT (REACTION) (RECORD EPISODE)))

The foregrounding module has also been at work, dutifully noting that in the first sentence, Peter, John, home, and school were all mentioned. These four concepts are thus foregrounded, i.e., loaded into working memory.

The frame module has also been active, recognizing that the 'school' frame should probably be instantiated. This creates expectations for school-house, playground, parking lot, and various other facets.

The real-world component is dormant at this point. It has noted that John and Peter are moving from school to home, but has also noted the progressive tense, which indicates that the action has not yet been completed.

The representation to this point is:

```
Peter  p                Peter      |-> home
 &    <=> PTRANS <---    &     <--|
John                    John       |-< school
```

(note: the 'p' above the '<=>' denotes the progressive tense; in general tense markers will be omitted from the diagrams, for the sake of simplicity). and the set of predictions active is something like

(h) ((EVENT) (RECORD EPISODE))
(i) ((ACTION) (PREDICT (REACTION) (RECORD EPISODE)))
(j) ((PLAYGROUND) (FRAMEREF SCHOOL))
....

plus predictions for other aspects of the 'school' frame, not shown here.

Since we have no strong predictions as to the form or content of the second sentence, we start off with the 'default' set (in order):

(k) ((NG) (RUMINATE ACTOR))
(l) ((PG) (RUMINATE (TIME SPACE)))
(m) ((CLAUSE SUBORDINATE) (RUMINATE (INST MANNER)))

(note that this is remarkably similar to the standard 'start' node of an ATN:

with one significant difference: prediction (1) will pick up locative or temporal phrases anywhere in the sentence; with an ATN, the arc would have to be duplicated, at every node at which it could conceivably occur.)

Thus, we begin parsing the second sentence. The subordinate clause is picked up first. As it is being processed, the collateral module has two brief moments of activity. First, the pronoun 'they' is interpreted; this causes no difficulty, since the theme of the previous sentence was 'Peter and John', and the concept 'Peter and John' is also foregrounded. In addition, the use of a pronoun at this point in the discourse is consistent with the somewhat simplistic rules for specificity.

The other significant point occurs when 'the playground' is encountered. The use of the definite NG is explained by the expectations spawned by the 'school' frame, so everything is all right. However, the frame module jumps in again at this point, to instantiate the 'playground' frame, with appropriate expectations for grass, swings, etc.

To return to the top level: the system finishes picking up the subordinate clause. At this point, the collateral module again becomes active, realizing that the use of information blocks here is significant. In accordance with its principles, it overrides the default prediction, which is trying to treat the clause as either instrument of manner, and instead places the clause in the 'time' slot of the newly built structure.

Parsing continues, with the rest of the sentence being handled in the obvious way, to produce:

```
John             o           time  John           o John  |->
 &  <=>MTRANS <--animal  <------  & <=>PTRANS <--  & <-| plygd
Peter               A              Peter            Peter |-<
                    | loc
                  grass
```

Again nothing unusual has happened; the theme has remained consistent, but a new concept, an animal, has been introduced. Note that the definite reference to grass is valid, since grass is mentioned in the playground frame. Interestingly, direct mention of grass in the school frame would be invalid; the playground frame must be invoked first. Scragg (1975, pp. 9-11) discusses this problem of 'continuity of contexts'.

Since the conceptualization is recognized as an action, a prediction for 'reaction' is spawned, together with the ones still active from the school and playground frames.

As we move into the next sentence, the word 'it' is picked up first. This causes a couple of interesting effects: first of all, the pronoun is resolved; this causes no problem, as there is only one concept currently foregrounded which it could match, and this binding is performed. Second, the theme module, on looking this over, discovers that the theme has changed. This tells it that we are not talking exclusively about Peter and John, but rather about an event in which they happen to be involved. This causes the system to 'back off' a little, and reorganize the global structure it is building.

Processing of the sentence continues, without any unusual happenings. However, when the sentence is interpreted, the system realizes that it is but a further specification of a previously mentioned concept. Thus (in accordance with and the principle of 'integrated memory') it does not build a separate conceptual structure for this, but instead adds the new information to the old slot. The last conceptualization now looks like:

```
John                o
 &  <=> MTRANS <-- rabbit    . . . .
Peter               A
                    | loc
                   grass
```

(note that I have been using proper names and generic names to label concepts; of course, these are only 'token' instances of 'type' nodes in a semantic net -- the labelling used here is for convenience).

Moving on, we begin the next sentence, and pick up the first clause: 'the boys ran after it'. One point is worth mentioning here -- the use of 'the boys' rather than 'they' to refer to Peter and John is consistent, because this concept was not mentioned in the previous sentence, hence faded 'off-stage'. Thus, the use of 'they' here would have caused problems in resolution.

Once the clause is picked up, a structure is built for it:

```
John                o   John       |-> rabbit
 &   <=>  PTRANS <----   &    <---|
Peter                  Peter       |-<
```

At this point, the inferencing module, at last, is allowed to say something. Triggered by the verb 'chased', it loads a series of inferences:

(a)

```
                      o     |-> Y             |->
 ( (X <=> PTRANS <--X <-|        )   (Y <==|    frightened )    )
                            |-<               |-<
```

i.e., if something chases an animal Y, that animal might be frightened.

(b)

```
        |->                                         o      |-> away
 (  (Y <==|     frightened )   ( Y <=> PTRANS <--Y <--|          ) )
        |-<                                                |-<
```

i.e., if an animal Y is frightened, that animal will move away from whatever is frightening it.

(c)

```
                      o     |-> Y         X    |->
 (  (X <=> PTRANS <--X <-|        )   (   <==|    physcont  )  )
                            |-<           Y    |-<
```

i.e., if X is chasing Y, X might catch Y.

Figure 10 - Set of Inferences for the Verb 'Chased'

The first and third of these can go immediately, and the second is fired by the hypothetical situation which is instantiated as a result of the first (this chaining effect is very similar to Rieger's work). Thus we have three expectations:

- -that the rabbit will be frightened
- -that it will run away
- -that the boys will catch it

Armed with these, the system analyzes the second part of the sentence. Conveniently, the structure found (that of the rabbit hopping away) corresponds to one of our expectations (the second). Thus, the chain of causal inferences joining the two events is instantiated, with the final form represented as:

```
John                       John        |-> rabbit
 &   <=>  PTRANS <----      &    <---|
Peter                     Peter        |-<
            A
           ||| c
           |||

               |->
    rabbit <==|      frightened
               |-<
            A
           ||| c
           |||
                                  |-> away
rabbit <=> PTRANS <-- rabbit |
                                  |-<
```

Figure 11 - A Causal Chain

One more point remains, however; the use of the word 'but' satisfies the collateral check for alternatives, hence the only remaining result of the first clause is <u>negated</u>:

```
John & Peter    ¬  |->
              <====|    physcont
    rabbit         |-<
```

At this point, we have found an action/reaction sequence, which satisfies the criteria for an <u>episode</u>. Thus, the text grammar proceeds, and predicts for the only remaining feature, denouement.

The last sentence satisfies this expectation, and the story is finished, with the result being:

```
        John  p          o  John    |> home
          &  <=> PTRANS <--  &    <-|
        Peter               Peter   |< school
                   /
                  /
                 / setting
                /
              |_

John          o          time   John           o John  |->
 &  <=>MTRANS <--rabbit  <------  & <=>PTRANS <--  & <-| plygd
Peter            A             Peter            Peter |-<
                 | loc
        A       grass
        |
        | then
        |
John           o  John   |->rabbit  John & Peter  ¬  |->
 &   <=>PTRANS <---  &  <-|                     <====| physcont
Peter             Peter  |-<           rabbit        |-<
             A
             ||| c
             |||
              |->
    rabbit <==|     frightened
              |-<
             A
             ||| c
             |||
                                   |-> away
 rabbit <=> PTRANS <-- rabbit |
                                   |-<
             /
            /
           / denouement
          /
        |_
 John          o  John    |-> home
  &  <=> PTRANS <--  &   <-|
 Peter            Peter   |-<
```

Figure 12 - Final Representation of 'Rabbit' Paragraph

V.4 The Implementation

The system described here has been partially implemented, to the point that the remaining work should be (hopefully) straightforward.

The system is written in LISP/MTS (Wilcox and Hafner, 1974), and runs in 290K bytes (including the LISP interpreter) on an IBM 370/168.

Of the sections diagrammed in the thesis (p. 92), the analysis (i.e., single sentence), prediction control (i.e., execution), and prediction manager have been fully implemented. The modules corresponding to the various I.S's remain in different stages of completion. Theme, real-world knowledge, and foregrounding are relatively complete; text structure and collateral, somewhat less so; frames and cohesion are still very fluid.

I feel that the system could be completed in a reasonable amount of time; the remaining work consists mainly of empirical observations, to provide breadth for the system. However, there still remains the possibility of some unseen conceptual block destroying an unsubstantiated premise.

Of the work remaining, probably the most interesting would lie in modification to the prediction manager. That is, the prediction manager as currently implemented is somewhat simplistic, because of an absence of thorough testing. The next step should be to run the system on a large corpus of examples, observing the decisions demanded of the prediction manager, and making extensions accordingly.

With respect to the sections implemented thus far, probably the most instructive has been the prediction control system itself. The basic nature of predictions has changed several times, as increasingly complex actions and interactions have been demanded. The current form of predictions -- with TEST, ACTION, STRENGTH, SOURCE, CANCEL, and OBL? -- was developed as successive needs arose; there is no guarantee that it will not be changed again. This work has also provided considerable insight into the questions of control of large systems; in particular a deeper understanding of psychologs was gained.

Probably the largest single benefit gained in the implementation effort was an appreciation of the complexity of natural language, and the structures required to handle it.

In conclusion, the implementation has been educational, if not completely successful.

VI. Conclusions

VI.1 Review

This paper has been an attempt to delineate the problems involved in discourse analysis, together with some possible solutions.

Discourse itself was described, and six features -- staging, cohesion, collateral, text structure, context, and real-world knowledge -- were identified as being potentially useful in the analysis process.

After appropriate psychological and computational preparation, a model for discourse analysis was presented. This model reflected two design features:

i) use of a modified prediction system, to handle the complexities of discourse;

ii) a 'sufficient-information' assumption, which specifies that natural language analysis must proceed by extracting the maximum amount of information available at each point in the processing.

The model was specified in detail, and examples were presented.

I feel that the system described here is reasonably sound. The list of information sources provides an extensive, although not complete, characterization of the useful features of discourse. The development of the prediction-based system of analysis illustrates one possible method; there are undoubtedly others. Most importantly, this thesis has discussed one approach to the problem of discourse analysis, and shown that it is viable.

VI.2 Future Work

The next step in this framework would be to finish the implementation. As discussed in section V.4, this is semi-complete, but much remains to be done. The effort of specifying the system at the level of code would probably reveal some hidden flaws, as well as some interesting effects. Also, a complete implementation would make the system available for testing of various sorts, concerning efficiency, completeness, etc.

After this, more empirical work could be performed. That is, the system could be run on a large corpus of examples, to determine:

i) the characteristics of each information source. The system developed in Chapter II was somewhat superficial; more detail is needed.

ii) additional information sources. There are obviously features of discourse which have not been mentioned here (e.g., presupposition, diction). These and others must be empirically developed.

iii) general control requirements of the system. The current control structure (diagrammed on p. 92) represents the present system. With more thorough testing, this would evolve, as increasingly complex demands were put on the system.

This work would augment the breadth of the system -- something that is entirely too restricted at present.

The final step would be to iterate, and return to the design phase, this time with a more specific set of constraints.

Probably a number of things wuld change here. The use of Conceptual Dependency as a representation, while adequate for the demands made of it so far, seems to be approaching the limits of its effectiveness. A more effective representation would be needed; this would involve a number of questions about the use of primitives, propositional vs. analogical forms, etc.

The prediction system would also undergo modification. Based on the (somewhat belated) discovery that it is a form of psycholog, I would try to use the results of work on PSs (in particular, knowledge-source ones like DENDRAL and MYCIN) to provide a computationally (and theoretically) more elegant system.

Other, minor, modifications would also occur. The prediction manager, if it still existed, would be redesigned, on a more theoretical basis. The 'analysis' module, and some of its methods, would probably also need revision.

Thus, the next step is -- keep going.

VI.3 The Future of Discourse Analysis

I feel that discourse analysis has a definite place in computational linguistics. Discourse is one of the richer areas of linguistics; at the same time, it is one of the most structured. This inherent structure facilitates the process of analysis, and provides a lot of gratuitous information for our use. Why not use it?

In conclusion, the thesis has indicated that we need a flexible, dynamic system to handle discourse, and has discussed one such system -- predictions.

VII. Bibliography

Ammon, P.R., The Perception of grammatical relations in sentences, *J. Verbal Learning and Verbal Behavior*, 7, 1968, 869-875

Anderson, J., *Language, Memory, and Thought*, Potomac, Erlbaum, 1976

Anderson, J.R. and G.H. Bower, *Human Associative Memory*, Toronto, Winston, 1973

Bartlett, F., *Remembering: a Study in Experimental and Social Psychology*, Cambridge, Cambridge University Press, 1932

Becker, A.L., Symposium of the paragraph, *College Composition and Communication*, 17, 1966, 67-72

Bever, T., The Cognitive Basis for Language Structures, in *Cognition and the Development of Language*, J.R. Hayes (Ed.), New York, Wiley, 1970

Bobrow, D. and D. Norman, Some Principles of Memory Schemata, in *Representation and Understanding*, D. Bobrow and A. Collins (Ed.), New York, Academic Press, 1975

Bock, J.K. and W.F. Brewer, Recall of sentences with alternate surface structures, *J. Exp. Psych.*, 103, 1974, 837-843

Borgida, A., *Topics in the Understanding of English Sentences by Computer*, TR 78, University of Toronto, 1975

Bransford, J. and J. Franks, The abstraction of linguistic ideas, *Cognitive Psychology*, 1, 1971, 331-350

Bransford, J., J. Barclay, and J. Franks, Sentence memory: a constructive vs. interpretive approach, *Cognitive Psychology*, 3, 1972, 193-209

Carney, T., *Content Analysis: A Technique for Systematic Inference from Communications*, Winnipeg, University of Manitoba Press, 1972

Cercone, N., *Representing English in Extended Semantic Nets*, TR 75-19, University of Alberta, 1975

Chafe, W., *Meaning and the Structure of Language*, Chicago, University of Chicago Press, 1970

------, Language and memory, <u>Language</u>, 49, 1973, 261-284

------, Language and consciousness, <u>Language</u>, 50, 1974, 113-134

Charniak, E., <u>Toward a Model of Children's Story Comprehension</u>, AI-TR-266, M.I.T., 1972

Chomsky, N., <u>Aspects of the Theory of Syntax</u>, Cambridge, M.I.T. Press, 1965

------, Deep Structure, Surface Structure, and Semantic Interpretation, in <u>Semantics</u>, D.D. Steinberg and L. Jakobovits (Ed.), Cambridge, Cambridge University Press, 1971

Christensen, F., Symposium on the paragraph, <u>College Composition and Communication</u>, 17, 1966, 60-66

Clark, H.H. and E.V. Clark, Semantic distinctions and memory for complex sentences, <u>Quart. J. Exp. Psych.</u>, 20, 1968, 129-138

Cofer, C.N., The Structure of Human Memory, in <u>San Francisco</u>, Freeman (Ed.), 1976,

Cohen, P.F., <u>A Prototype Natural Language Understanding System</u>, TR-64, University of Toronto, 1974

Crothers, E., Memory Structure and the Recall of Discourse, in <u>Language Comprehension of Knowledge</u>, J. Carroll and R. Freedle (Ed.), Washington, Winston, 1972

Davidson, J.E., Sources of Information in Discourse Analysis, <u>Proc. 1st CSCSI/SCEIO Nat'l Conference</u>, Vancouver, 1976a

------, An Evaluation of Conceptual Dependency as a Representation for Language and Knowledge, unpublished term paper, U.B.C., 1976b

Davis, R., B. Buchanan, and E. Shortliffe, <u>Production Rules as a Representation for a Knowledge-based Consultation Program</u>, A.I. Memo 266, Stanford University, 1975

Davis, R. and J. King, <u>An Overview of Production Systems</u>, A.I. Memo 271, Stanford University, 1975

van Dijk, T.A., <u>Some Aspects of Text Grammars</u>, The Hague, Mouton, 1972

Earl, L., Use of word government in resolving syntactic andsemantic ambiguity, <u>Information Storage and Retrieval</u>, 9, 1973, 639-664

Earley, J., An efficient context-free parsing algorithm, Comm. A.C.M., 13, 1970, 94-102

Erman, L. and V. Lesser, A Multi-Level Organization for Problem-solving Using Many, Diverse, Cooperating Sources of Knowledge, Proc. 4th Int'l Joint Conference Artificial Intelligence, Tbilisi, 1975, 126-133

Fabens, W., PEDAGLOT and understanding natural language processing, Amer. J. Computational Linguistics, 1975, mf 32

Fiegenbaum, E. et al., On Generality and Problem Solving -- A Case Study Involving the DENDRAL Program, in Machine Intelligence 6, B. Meltzer and D. Michie (Ed.), Edinburgh, University Press, 1971

Fillenbaum, S., On coping with ordered and unordered conjunctive sentences, J. Exp. Psychology, 87, 1971, 93-98

Fillmore, C.J., The Case for Case, in Universals in Linguistic Theory, E. Bach and R. Harms (Ed.), New York Winston, 1968

Fodor, J. and M.F. Garrett, Some Reflections on Competence and Performance, in Psycholinguistic Papers, J. Lyons and R. Wales (Ed.), Edinburgh, University Press, 1975

Fowler, J., LOTUS - A Procedural Environment, M.Sc. Thesis, Computer Science Department, U.B.C., 1976

Frase, L., Structural analysis of the knowledge that results from thinking about text, J. Educational Psych., 60, 1969, Monograph

Frederiksen, C., Effects of context-induced processing operations on semantic information acquired from discourse, Cognitive Psychology, 7, 1975, 139-166

------, Semantic Processing Units in Understanding Text, in Discourse Production and Comprehension, R. Freedle (Ed.), Hillsdale, Erlbaum, 1976

Goffman, E., Frame Analysis, Cambridge, Harvard U. Press, 1974

Griggs, R.A., The Recall of Linguistic Ideas, J. Exp. Psychology, 103, 1974, 807-809

Grimes, J.E., Kinds of information in discourse, Kivung, 4, 1971, 64-74

------, Outlines and overlays, Language, 48, 1972, 513-524

------, Thread of Discourse, The Hague, Mouton, 1975

Halliday, M.A.K., Notes on Transitivity and Theme in English II, Journal of Linguistics, 3, 1969, 199-244

------, Language Structure and Language Function, in New Horizons in Linguistics, J. Lyons (Ed.), London, Pelican, 1970

Halliday, M.A.K. and R. Hasan, Cohesion in English, London, Longman's, 1976

Hausenblas, K., On the characterization and classification of discourses, Travaux Linguistiques de Prague, 1, 1964, 67-83

Havens, W.S., Can Frames Solve the Chicken and Egg Problem?, Proc. First CSCSI/SCEIO Nat'l Conference, Vancouver, 1976

Hays, D., Dependency theory: A formalism and some observations, Language, 40, 1964

Holsti, O., Content Analysis of Social Sciences and Humanities, Reading, Addison-Wesley, 1969

Hunt, E. and S. Poltrock, The Mechanics of Thought, in Human Information Processing: Tutorials in Performance and Cognition, B. Kantowitz (Ed.), Hillsdale, Erlbaum, 1974

James, C., Theme and imagery in the recall of active and passive sentences, J. Verbal Learning and Verbal Behavior, 1, 1972, 205-211

Jervis, J., An Augmented Transition Network Grammar for English, Master's Thesis, Computer Science Department, U.B.C., 1974

Johnson-Laird, P.N., The Perception and Memory of Sentences, in New Horizons in Linguistics, J. Lyons (Ed.), London, Pelican, 1970

Kaplan, R., Augmented transition networks as psychological models, Artificial Intelligence, 3, 1972, 77-100

------, On Process Models for Sentence Analysis, in Explorations in Cognition, D. Norman and D. Rumelhart (Ed.), San Francisco, Freeman, 1975

Kinneavy, J.L., A Theory of Discourse, Englewood Cliffs, Prentice-Hall, 1971

Kintsch, W., Representation of Meaning in Memory, Hillsdale, Erlbaum, 1974

------, Memory Representation of Text, in Information Processing and Cognition, R. Solso (Ed.), Hillsdale, Erlbaum, 1975

------, Memory for Prose, in The Structure of Human Memory, C.N. Cofer (Ed.), San Francisco, Freeman, 1976

Kintsch, W. et al, Composition and recall of text as a function of contextual variables, J. Verbal Learning and Verbal Behavior, 14, 1975, 196-214

Kummer, W., Outlines of a model for a grammar of discourse, Poetics, 3, 1972

Lakoff, G., Structural complexity of fairytales, The Story of Man, 1, 1972, 128-150

Lees, R.B. and E.S. Klima, Rules for English pronominalization, Language, 39, 1963, 17-29

Lenat, D., BEINGS: Knowledge as Interacting Experts, Proc. 4th Int'l Joint Conference Artificial Intelligence, Tbilisi, 1975, 483-490

Lesgold, A.M., Pronominalization: a device for unifying sentences in memory, J. Verbal Learning and Verbal Behavior, 11, 1972, 316-326

Lindsay, P. and D. Norman, Human Information Processing: An Introduction to Psychology, New York, Academic Press, 1972

Longacre, R.E., Sentence structure as a statement calculus, Language, 46, 1970, 783-815

McCalla, G. and M. Kuttner, An extensible feature-based procedural question-answering system to handle the written section of the B.C. Driver's Exam, CSCSI/SCEIO Newsletter, 2, 1975

McCalla, G., An Approach to the Organization of Knowledge for a Model of Conversation, Thesis Progress Report, U.B.C., 1976

McDermott, D., Artificial intelligence meets natural stupidity, SIGART Newsletter, 57, 1976, 4-10

Marcus, M., Wait and See Strategies for Parsing Natural Language, working paper 75, M.I.T. A.I. Lab, 1974

------, Diagnosis as a Notion of Grammar, in Theoretical Issues in Natural Language Processing, R. Schank and B. Nash-Webber (Ed.), Proceedings of the conference, Cambridge, 1975

Marks, L.E. and Miller, G., The role of semantic and syntactic constraints in the memorization of English sentences, J. Verbal Learning and Verbal Behavior, 3, 1964, 1-5

Meyer, B.J.F., The Organization of Prose and its Effects on Memory, Amsterdam, North-Holland, 1975

Miller, G., The magical number 7, plus or minus 2, Psychological Review, 63, 1956a, 81-97

------, Human memory and the storage of information, I.R.E. Trans. on Information Theory, IT-2, 1956b, 129-137

------, English Verbs of Motion: A Case Study in Semantics and Lexical Memory, in Coding Processes in Human Memory, A. Melton and E. Martin (Ed.), Washington, Winston, 1972

Miller, G. and L.A. McKean, A chronometric study of some relations between sentences, Quart. J. Exp. Psych., 16, 1964, 297-308

Minsky, M., A Framework for Representing Knowledge, AIM-206, M.I.T. A.I. Lab, 1974

Moran, T.P., The Symbolic Imagery Hypothesis, Computer Science Department, Carnegie-Mellon Univeristy, 1973

Morris, C.W., Signs, Language, and Behavior, Englewood Cliffs, Prentice-Hall, 1946

Newell, A., Production Systems: Models of Control Structure, in Visual Information Processing, W. Chase (Ed.), Washington, Winston, 1973

Norman, D., Memory and Attention, New York, Wiley, 1976

Norman, D. and D. Rumelhart (Ed.), Explorations in Cognition, San Francisco, Freeman, 1975

Olson, D., Language and thought -- aspects of a cognitive theory of memory, Psych. Review, 77, 1970, 257-273

Palmer, S., Visual Perception and World Knowledge, in Explorations in Cognition, D. Norman and D. Rumelhart (Ed.), San Francisco, Freeman, 1975

Paxton, W. and A. Robinson, system integration and control in a speech understanding system, Amer. J. Computational Linguistics, 1975, mf 34

Petrick, S., On Natural Language Based Query Systems, Res. Rpt. #5577, I.B.M. T.J. Watson, 1975

Phillips, B., Judging the coherency of discourse, American J. Computational Linguistics, 1975, mf 35

Post, E., Formal reductions of the general combinatorial problem, Amer. J. Math., 65, 1943, 197-268

Propp, V., Morphology of the Folktale, Austin, U. of Texas Press, 1968

Quillian, M.R., The teachable language comprehender: a simulation program and a theory of language, Comm. A.C.M., 12, 1969, 459-476

Reddy, R., L. Erman, and Neely, A model and system for machine recognition of speech, I.E.E.E. Trans. Audio and Electroacoustics, QU-21, 1973, 229-238

Reddy, R. and A. Newell, Knowledge and Its Representation in a Speech Understanding System, in Knowledge and Cognition, Gregg (Ed.), Potomac, Erlbaum, 1974

Rieger, C.J., Conceptual Memory and Inference, in Conceptual Information Processing, R. Schank (Ed.), Amsterdam, North-Holland, 1975

Riesbeck, C., Computational Understanding: Analysis of Sentences in Context, Ph.D. thesis, Computer Science Department, Stanford University, 1974

------, Conceptual Analysis, in Conceptual Information Processing, R. Schank (Ed.), Amsterdam, North-Holland, 1975

Robinson, J., A tuneable performance grammar, Amer. J. Computation Linguistics, 1975, mf 34

Rodgers, P., A discourse-dentred thetoric of the paragraph, College Composition and Communication, 17, 1965, 2-11

Rosenschein, S., How does a system know when to stop inferencing?, Amer. J. Computation Linguistics, 1975, mf 36

Rowat, P., Of Crayfish and Production Nets, Proc. First CSCSI/SCEIO Nat'l Conference, Vancouver, 1976

Rubin, A., Grammar for the People: Flowcharts of SHRDLU's Grammar, AIM-282, M.I.T. A.I. Lab, 1973

Rumelhart, D., Notes on a Schema for Stories, in Representation and Understanding, D. Bobrow and A. Collins (Ed.), New York, Academic Press, 1975

Rumelhart, D., P. Lindsay, and D. Norman, A Process Model for Long-term Memory, in The Organization of Memory, E. Tulving and W. Donaldson (Ed.), New York, Academic Press, 1972

Sachs, J., Recognition memory for syntactic and semantic aspects of connected discourse, Percpetion and Psychophysics, 2, 1967, 437-442

Sanders, Gerald A., On the natural domain of grammar, Linguistics, 63, 1970, 51-123

Scarl, E., A Flexible ATN, a Flexible Morpher, informal talk, First CSCSI/SCEIO Nat'l Conference, Vancouver, 1976

Schacter, P., Focus and relativization, Language, 49, 1973, 19-45

Schank, R., Identification of the Conceptualizations Underlying Natural Language, in Computer Models of Thought and Language, R. Schank and K.M. Colby (Ed.), San Francisco, Freeman, 1973

------, Causality and Reasoning, TR-1, I.S.S.C., 1974a

------, Understanding Paragraphs, TR-6, I.S.S.C., 1974b

------, Representation and Understanding of Text, NATO ASI, Santa Cruz, 1975a

------ (Ed.), Conceptual Information Processing, Amsterdam, North-Holland, 1975b

Scragg, G., Frames, Planes and Nets: A Synthesis, TR 19, I.S.S.C., 1976

Shannon, C., Prediction and entropy of printed English, Bell Sys. Tech. J., 30, 1951, 50-64

Simmons, R., Semantic Nets: Their Computation and Use for Understanding English Sentences, in Computer Models of Thought and Language, R. Schank and K.M. Colby (Ed.), San Francisco, Freeman, 1973

Simon, H., How big is a chunk?, Science, 183, 1974, 482-488

Sulin, R. and D. Dooling, Intrusion of a thematic idea in retention of prose, J. Exp. Psych., 103, 1974, 255-263

Taylor, B.H. and R.S. Rosenberg, A case-driven parser for natural language, Amer. J. Computational Linguistics, 1975, mf 31

Tesler, L., H. Enea, and D. Smith, The LISP70 Pattern Matcher, Proc. Third Int'l Joint Conference Artificial Intelligence, Tbilisi, 1975

Tulving, E. and W. Donaldson (Ed.), The Organization of Memory, New York, Academic Press, 1972

de Villiers, P.A., Imagery and theme in recall of connected discourse, J. Exp. Psych., 103, 1974, 263-268

Waterman, D., Adaptive Production Systems, Complex Information Processing Working Paper No. 285, Carnegie-Mellon University, 1974

Wilcox, B. and C. Hafer, LISP/MTS User's Guide, Mental Health Research Institute, Ann Arbour, 1974

Wilks, Y., Preference Semantics, AIM-206, Stanford University, 1974

Winograd, T., Procedures as a Representation for Data in a Computer Program for Understanding Natural Language, MAC-TR-84, M.I.T., 1970

------, Breaking the complexity barrier: again, SIGPLAN Notices, 10(1), Jan. 1975, 13-22

------, Frame Representations and the Declarative/Procedural Controversy, in Representation and Understanding, D. Bobrow and A. Collins (Ed.), New York, Academic Press, 1975

Woods, W.A., Procedural Semantics for a Question-Answering Machine, Proc. AFIPS 1968 F.J.C.C., 1968, 457-471

Woods, W.A., Transition Network Grammars for Natural Language Analysis, Comm. A.C.M., 13, 1970, 591-606