Interpreting Narratives with Evaluative Statements

Annanya Bhattacharjee

AI TR87-47 January 1987

Support for this research was provided by Grant No. ARO DAAG29-084-K-0060 from the US Army Research Office.
INTERPRETING NARRATIVES
WITH EVALUATIVE STATEMENTS

BY

ANANNYA BHATTACHARJEE, B.A.

THESIS
Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE
THE UNIVERSITY OF TEXAS AT AUSTIN

December, 1986
ACKNOWLEDGEMENTS

I am very grateful to my advisor, Dr. Robert F. Simmons, for his guidance and encouragement. I appreciate his patience through the process of writing this thesis. I would also like to thank my second reader, Dr. Gordon Novak for his time.

I would like to thank my parents for instilling in me the desire to learn. I cannot thank enough my brother, Amitava, and my sister-in-law, Jael, for changing the course of my life by presenting me the opportunity to grow and study in the way I wanted to.

I must express my appreciation for some of my friends, who have had the most occasion to help me out through discouraging times with their support: Dilip, and Ravi. I would also like to thank my friend and colleague, Ezat Karimi, for her encouragement. I hope our discussions continue to be stimulating for both of us.

Anannya Bhattacharjee

The University of Texas at Austin
December, 1986
ABSTRACT

Discourse structure is one of the foremost areas of study in natural language. Discourse is concluded to be governed by structures that go beyond the sentence or the relation between sentences. These structures capture a global purpose, which is the motivation for discourse. A form of such a global structure is an evaluative statement in discourse. This thesis is a case study of narratives and their associated evaluative statements. It is a computational attempt at reflecting the process by which an evaluative statement is used in interpreting narratives.
# TABLE OF CONTENTS

Acknowledgements .................................................. ii

Abstract ............................................................. iii

Table of Contents .................................................. v

Chapter 1. INTRODUCTION ........................................... 1
   1.1. NATURE OF DISCOURSE ..................................... 1
   1.2. THE MICROLEVEL IN DISCOURSE ............................. 3
   1.3. THE MACROLEVEL IN DISCOURSE ............................. 4
   1.4. MACROSTRUCTURES GIVE MEANING .......................... 5
   1.5. MACROSTRUCTURES ARE EVALUATIVE ....................... 5
   1.6. NARRATIVES AND MACROSTRUCTURES ....................... 6

Chapter 2. RESEARCH AND BACKGROUND ............................. 8
   2.1. NATURE OF READING ....................................... 8
   2.2. NARRATIVES AS PROBLEM-SOLVING ........................ 9
   2.3. NARRATIVES AND THEIR COMPLEXITIES .................... 9
   2.4. GENERATION OF NARRATIVES ............................... 10
   2.5. NARRATIVES CONVEY BELIEF SYSTEMS ..................... 11
   2.6. SEMANTIC ANALYSIS AS RULE-DRIVEN ..................... 12
   2.7. NARRATIVES AND THEIR MACROSTRUCTURES ................. 13
Chapter 1

INTRODUCTION

1.1 NATURE OF DISCOURSE

One of the central areas of research in natural language is discourse analysis. Discourse analysis developed into a discipline in the 1970's, as research on various aspects of language led to the realization that it was not enough to have grammars which provide "structural characterizations of isolated sentences" [Vandijk 85]. Attempts to understand the nature of discourse led to the realization that sentential structures function as local units to create a global environment, which is discourse. The need to explicate the meaning contained in this global environment made it increasingly necessary to study sentential structures in the context of cognitive processes which use them to arrive at a global meaning.

One is often tempted to think of text as a whole composed of clauses which are the parts. However, it is too simplistic to think of text as the larger structure composed of clauses as the smaller structures. A text is more than a structure: "A text is a semantic entity. It is true that texts are, on the whole, larger than clauses; what is more significant, however, is that they are one level of abstraction beyond the clause. The relationship is not so much one of size as one of overt to covert; the text is realized in clauses" [Halliday 82]. Therefore, to
study text merely as a conglomerate of clausal structures without attention to
the processes by which textual meaning is realized in the clausal structure, is to
miss the point.

Artificial Intelligence and Linguistics have both contributed to the
research in discourse analysis. The computational view of discourse differs from
the linguistic view, thus reflecting the difference between the nature of their
tasks. In computation, any meaningful form in discourse has to be explicitly
defined using symbols and procedures which manipulate these symbols. The
symbols, therefore, adopt a dynamic role in arriving at an interpretation of any
aspect of meaning in discourse. The linguist, on the other hand, uses forms with
pre-established definitions to study how they interrelate to establish meaning in
discourse. As Nils Erik Enkvist says,

"One difference between grammarians and students of artificial
intelligence has been in the nature of their tasks ... (The student of
artificial intelligence) will have to simulate entire systems of
cognition, systems of potential meanings, out of which he must find
those which correspond to a certain sentence of text. This compels
him to set up explicit descriptions of searches for meaning. Many
grammarians on the contrary have often assumed that they already
know what a certain sentence means. In these terms, members in
artificial intelligence will be compelled to set up descriptions for
processes that many grammarians have taken for granted. And this,
of course, will further increase the emphasis of processes".

--- [Enkvist 82]
In artificial intelligence, today, the worlds used to demonstrate such complex processes, are limited in their complexity. Case studies of such processes range from technical expository texts of relatively limited vocabulary to simple narratives. The enormity of the task involved in simulating processes to explicate meaning is apparent even in these limited domains.

1.2 THE MICROLEVEL IN DISCOURSE

The processing of a text begins with processing of the sentences in the text. Each clause is syntactically processed to reveal the main predicates and their cases. The local coherence between clauses is established by a process of reference resolution and by demonstrating links between the main predicates of a new clause and a previous one. The clausal level in discourse is referred to as its microlevel and processing at this level results in locally coherent microstructures [Vandijka 80].

Local coherence in text, through reference resolution and predicate links establishes an expected flow of events in the text. Links between the main predicates (that is, main verbs) provide a justification for the local occurrences of the predicates. Reference resolution reveals links between an object’s presence at different locations in the text. These two types of coherence are dependent on each other for their coherence. Thus, main predicates are linked, in part, due to links between the objects they contain. Similarly, objects in a text are linked, in part, due to links between the predicates which contain them.

However, local coherence by itself is not enough evidence of a global textual meaning because it is possible to have a locally coherent but aimless piece of text:
John was ill, so he called the doctor. But, the doctor could not come, because his wife wanted to go to the theatre with him. They were playing Othello, which she thought they could not miss because Shakespeare is one of the few drammatical authors, who .......

In order to arrive at a global purpose in discourse, one must study how local coherence plays a dynamic role in asserting global coherence.

1.3 THE MACROLEVEL IN DISCOURSE

The purpose of the text is not captured at the clausal level, but at the global level. The global textual meaning resides at the global or macrolevel which is a higher level abstracted from the microlevel. The global meaning is encoded in a macrostructure and the processes by which a macrostructure is processed to reveal the microstructures subsumed by it are called macroprocesses.

"Characteristic of (macrostructures) is that they dominate (ordered) sets of sentences or propositions..." [Vandijkb 85], and they also serve as "a sort of macro-interpretation of the content of sets of propositions". Such a macrostructure is considered to be stored in memory and used as an indexing tool to retrieve microstructures. The assignment of a macro unit to the units at the microlevel provides "a means to assess that all the units at the micro level belong together" [Vandijka 80]. If one imagines the macrostructure to be a higher-order action, then it is an action achieved by performing a sequence of actions at the micro level. Without the macro level interpretation one cannot distinguish one action sequence from another. Thus at each moment in the discourse, whenever a local act is performed, a global act is also being satisfied.
Van dijk advances his hypothesis about macrostructures further to claim that "all discourse is produced on such a global semantic plan" [Vandijk 79]. Thus a macrostructure can be a reader's cognitive structure, as it is constructed during reading, or it can be a writer's global plan based on which he writes the text.

1.4 MACROSTRUCTURES GIVE MEANING

The task of understanding a text is a complex one and can be reflected in the process of mapping a macrostructure onto the microstructures of a text. A series of mapping rules map the macrostructure to the microstructures of the text. A computational effort at such a mapping should reflect the processes by which a reader maps macrostructures onto a text.

1.5 MACROSTRUCTURES ARE EVALUATIVE

Text is a form of communication and hence subject to the interests, attitudes, and beliefs of the communication. Therefore, a macrostructure for discourse not only reflects the information content in the discourse but also the communicator's evaluation of the content. In other words, a macrostructure can be alternatively an "evaluative statement" [Grimes 75].

An evaluative statement is often a part of the discourse either at the beginning (in the form of an introduction) or at the end (in the form of a moral to a story). An evaluative statement at the beginning or at the end has global scope over the discourse, and reveals the aim of the discourse. Due to its subjective nature, "it indicates an appropriate reading for the text, by expressing the macrostructure of the text as intended by the author" [Vandijk 80].
Evaluative statements "express what the author found to be the more important information of the text."

1.6 NARRATIVES AND MACROSTRUCTURES

Discourse in natural language is, thus, concluded to be governed by structures that go beyond the sentence or the relation between sentences. These structures capture a global purpose, which is the motivation for discourse.

At the microlevel, discourse is an ordered sequence of n-tuples. At the macrolevel, the macrostructure is also an ordered sequence of n-tuples, each of which is defined by a particular sequence of n-tuples from the microlevel. The relations between the macrolevel n-tuples demonstrate the relations between sequences of microlevel n-tuples.

A narrative "is a specific type of action discourse" [Vandijkb 85] and consists of action descriptions and state descriptions. Action descriptions consist of sequences of propositions denoting actions and state descriptions consist of initial and final state descriptions of the actors and objects involved.

Each action is a complex consisting of "ordered sequences" [Vandijka 80] of actions with a local intention. At a global level, each such sequence serves a global purpose. This level determines how complex actions interact with one another to serve the global purpose.

A macrostructure for a narrative serves to condense the action sequences in the narrative and to express their global purpose.
In this thesis, simple narratives and a type of macrostructure called the moral are studied. Each narrative studied has an associated moral which captures its global meaning. The computational effort attempts to associate a moral with a narrative by mapping the moral onto the events in the narrative.

This thesis thus examines computationally:

1. the local level of a narrative and its associated microstructures
2. the global level of a narrative and its associated macrostructure
3. the process by which the macrostructure is mapped onto the microstructures.
Chapter 2

RESEARCH AND BACKGROUND

2.1 NATURE OF READING

Research on narratives is distinguished by attempts to demonstrate the interaction between the reader and the writer as the reader reads the text. The writer communicates a set of actions and beliefs to the reader through the narrative and the reader is expected to infer this set as he/she reads the text. The inferences can be based either on explicit facts stated in the text or on implicit facts derived from the stated facts and the store of world and linguistic knowledge assumed in common between author and reader.

The reader gets a chance to evaluate the accuracy of his inferences at some point in the narrative, where the writer exposes the theme or the macrostructure dominating the narrative. Such an exposition may be present either in the beginning or at the end of the text, and is distinguished from the rest of the text by its higher level of abstraction.

A computational study of the process of thematic evaluation of a narrative must necessarily involve a study of the nature and structure of the macrostructure and of a process mapping the macrostructure onto the facts in the narrative.
2.2 NARRATIVES AS PROBLEM-SOLVING

James Meehan introduced the study of narratives based on the goals and plans of a character [Meehan]. Thus, the narrative consisted of actions that the character performed to realize the goal. The macrostructure for such a narrative represents a problem to be solved, a goal to be realized. For example, "X is hungry" creates a goal "X wants food". The goal is then the basis for events such as X's search for food leading to X's acquiring some form of food.

Meehan's system being clean and powerful as it is, produced intricate plans in narratives. But, its domains remained fairly simple and well-defined. A more complex narrative will necessarily demand a more complex set of motives, plans and actions, and thus a more complex processing of facts.

2.3 NARRATIVES AND THEIR COMPLEXITIES

Michael Dyer's system, BORIS, is a computational study of more complex narratives [Dyera 82]. The complex nature of the structures and interactions that he uses reflects the enormity of the task. He uses seventeen different knowledge structures, which interconnect, to compute different aspects of a narrative. To mention a few of the knowledge structures that he uses: Thematic Abstract Units (TAUs), GOALs, PLANs, SCRIPTs, AFFECTs, InterPersonal Themes (IPTs), Relationships/Roles (RTs), among others. Each of these structures is created to handle a particular type of knowledge.

Most relevant in this discussion is his use of TAUs. TAUs are abstract structures encoding such information as is applicable to more than one specific activity or narrative. They are based on adages such as "A friend in need is a friend indeed" to encode knowledge in the form:
TAU-DIRE-straits:

x has a crisis goal G

x has experienced a planning failure
   (i.e., x can’t resolve the crisis by himself)

x seeks a friend y to be his agent
   (since y knows what plans to execute).

He uses TAU’s in two different ways:

1) as planning heuristics: For example, TAU-T00-LATE based on "A stitch in time saves nine" recognizes the fact that a planner needlessly delayed using an available plan, or in other words, if plans are executed too late, they may not fail, but it may make their the cost much higher.

2) as indices for episodes: For example, a TAU based on "Once you have outlived your usefulness, you are discarded" may be an index into an episode involving "job troubles". Thus, TAU’s are treated as macrostructures, since they are abstract structures which dominate sets of actions. He uses appropriate TAU’s to index episodes because TAU’s capture at an abstract level the theme or point of the episode.

2.4 GENERATION OF NARRATIVES

When Scott Turner and Michael Dyer built their story invention system, MINSTREL, they discovered that a general theory of themes and morals is necessary to produce narratives of interest and complexity [Turner,Dyer 85, Dyer]. Authors create new stories on the basis of recalled episodes and their
related themes. The episodes that an author recalls depends on his state of mind at the time, viz., his goals, the most recent facts he has heard, and so on. MINSTREL is an attempt to model this behaviour. On being given some inputs (corresponding to an author's initial state), it uses TAU's, prompted by the input, to recall episodes. The TAU, being a macrostructure, provides a global framework which has to be instantiated with events to create a new story.

Macrostructures are thus perceived to be a dominating feature of texts, playing an instrumental role in the process of recall or reproduction of text.

2.5 NARRATIVES CONVEY BELIEF SYSTEMS

Marcy Dorfman's model for understanding the point of stories is based on the assumption that in order to understand the theme of a narrative, one must also take into account the beliefs of the reader and the author [Dorfman 85]. Thus, in order to understand in depth the point the author tries to make, the reader must be able to understand the nature of the events and the outcome. Sometimes an author conveys his/her message in a direct fashion; but, there are times when an author conveys the message indirectly, say, by means of irony. Thus, for the reader to interpret the message correctly in the latter case, he/she must realize the irony of the situation.

For example, in a narrative where an over-confident and lazy character is rewarded in a situation over a diligent character, the author is probably trying to reveal instances of injustice that exist in the world. It would be a mistake to conclude that the author believes that over-confidence and laziness lead to success.
Marcy Dorfman's model understands such indirect narrative style with the use of +/- values attached to actions and outcomes. A reader's world is represented as the just world and the story's world is mapped onto the just world. If the mapping shows them to be identical, then the author's message is interpreted directly. If, however, the mappings show conflicting values, then the message is arrived at indirectly.

### 2.6 SEMANTIC ANALYSIS AS RULE-DRIVEN

Martha Stone Palmer, in her paper "Inference-Driven Semantic Analysis", discusses her approach to semantic analysis in natural language [Palmer 83]. Her paper is relevant here because the computational method used in this thesis to map macrostructure onto microstructures has analogies with her approach to semantic analysis in natural language.

Some of the tasks in semantic analysis are:

- filling semantic roles of semantic predicates, that is, the verbs,

- using inference rules to expand the representation of the verb into a more detailed representation,

- constraining the application of these inference rules,

- connecting the final representation of clauses with prior clauses.

Her approach involves features such as:

- lexical entries of verbs are filled with the help of domain-specific inference rules. The format is like

  - Verb(Entry1,Entry2) <- rules to fill in 'Entry1', 'Entry2'.

- inference rules define the semantic roles of the semantic predicates.
• inference rules infer information about semantic roles that are not directly expressed by the semantic predicate which contain them,

• application of inference rules is constrained by the fact that only those rules which are relevant for deducing the value of a particular semantic role are selected.

These features will be discussed in conjunction with the computation method of this thesis in chapter 4.

2.7 NARRATIVES AND THEIR MACROSTRUCTURES

This thesis is a case study involving simple narratives, called fables and their associated macrostructures, called morals. It attempts to follow computationally the process the reader undergoes when he/she reads a fable, followed by a moral. The process involves recapitulation and evaluation. The moral prompts the reader to recapitulate the events of the narrative in the light of the moral while at the same time evaluating the legitimacy of associating the moral with the narrative.
Chapter 3

COMPUTATIONAL STUDY

3.1 NATURE OF THE STUDY

This thesis is a computational case study to examine the process of mapping a macrostructure onto a text. The form of text used is simple narrative and the form of macrostructure studied is its moral. The narratives are chosen from Aesop’s Fables, which is a collection of enjoyable short stories from the animal world. The characters belong to the world of imagination but the events lead up to morals that are realistic reflections on the nature of the reader’s world.

Input to the computational system consists of the fables with their morals. The output is the sequence of events in the narrative which led to the moral, thus revealing the events which prompted the moral.

As a reader reads a fable, she/he retains the events in it. The moral occurs at the end of the story, giving the author’s evaluation of the preceding events. As soon as the reader encounters the first word in the moral, she begins to evaluate the story’s events in light of the moral. Thus, reading of the moral and evaluation of the story are interwoven together.
The computational process tries to reflect the reader's processes. At first the story is parsed and its events retained in a representation scheme. As soon as the moral is encountered the two processes, viz., parsing of the moral and tracing of events in the story leading to the moral, are carried on in an interwoven fashion.

An alternative way to compute the process would have been to first parse the moral completely and, then, to evaluate the story against the moral. However, this proved to be a cumbersome and unnatural process, not reflecting the reader's sequence of operations.

3.2 OVERVIEW OF THE DIFFERENT PARTS

The computational process consists of four parts: grammatical analysis, pronoun resolution, evaluation of story with respect to moral, and generation of the path traced in the story leading to the moral. The implementations of the first three processes are closely interrelated. The last process, generation, is the formation of a readable output from results obtained from the preceding three processes.

The grammar parses the story and converts each sentence into n-tuples. An n-tuple consists of values for the constituents of the sentences, viz., the action, its agent, its affected entity, its location. Each n-tuple is stored as part of a representation for the story. Therefore, the representation for the story consists of a list of n-tuples. The references are resolved during parsing. An alternative way to resolve references would have been to produce all the n-tuples and then resolve references in them. However, it is an unnatural method considering that
the reader reads and resolves references simultaneously. An elegant computational process should reflect this simultaneity.

The same grammar also parses the sentences expressing the morals. As mentioned earlier, a reader begins to evaluate the story in light of the moral as soon as she/he encounters the first word in the moral. To reflect the simultaneity of reading and evaluation, the computational process begins to evaluate the story against the moral as soon as it begins to parse the moral. The moral is also parsed into n-tuples. These n-tuples however are not stored. As soon as an n-tuple is formed, it is evaluated against the n-tuples from the story to see if the events in the story successfully lead to the moral. When all the n-tuples in the moral are satisfied and their constraints met, the process of evaluation ends successfully. The sequences of events traced in the story are stored for the last process, generation.

References are resolved with the help of Alterman's seven coherence relations: subclass, sequel, sequence/subsequence, coordinate, antecedent, precedent, consequent [Alterman 82]. As each n-tuple is formed, its references are resolved with respect to previous n-tuples. Features such as number, inanimate/animate, are used to facilitate reference resolution.

The input to the generation part consists of lists of n-tuples. Each list contains an n-tuple derived from the moral, followed by the path of events traced in the story to satisfy the particular n-tuple. The path consists of n-tuples from the story representation, connected by Alterman's relations. The n-tuples are converted into simple English sentences and the relations between them are preserved by connecting them with appropriate connectives.
3.3 GRAMMAR

The grammar is written in the Prolog version of definite clause grammar, and it parses 25 sentences. At the clausal level as each clause is successfully parsed, an n-tuple is created to store the constituents of the clause. An n-tuple is a list, containing names of the constituents and their values. For example:

\[
[\text{act}, \text{Act}], [\text{agt}, \text{Agt}], [\text{ae}, \text{Ae}]
\]

\[
/ / / / / / / / \text{name value name value}
\]

The grammar parses a variety of sentences. Some examples are:

(1) The donkey found a lion's skin.

This is a simple sentence with one action, one agent and one affected entity. Thus the n-tuple formed is

\[
[\text{act,find}],[\text{agt,donkey}],[\text{ae,skin}]
\]

There is, however a possessive case which gives rise to another n-tuple

\[
[\text{act,poss}],[\text{agt,lion}],[\text{ae,skin}]
\]

(2) The donkey swaggered about the forest, roaring and growling.

In this sentence donkey performs three actions in the same location. The n-tuples formed are:
(3) A dog, carrying a bone in his mouth, was crossing a bridge.

This is an ellipsis in which the agent of carry is also the agent of cross.

The n-tuples are:

[[act,carry],[agt,dog],[ae,bone],[loc,mouth]]

[[pred,poss],[agt,dog],[ae,mouth]]

[[act,cross],[agt,dog],[ae,bridge]].

The resolution of references and the evaluation of the story's events against the moral are encoded in Prolog procedures. At the clausal level, that is, the level at which an n-tuple is formed, the three activities, viz., parsing, reference resolution and evaluation are performed together. Thus the clausal level rule is a complex rule consisting of three modules:
For example a clausal rule may look like:

\[ s \rightarrow np(Agt), \ vp(Act,Ae) \]

the n-tuple formed here is

\[
[ [act,Act],[agt,Agt],[ae,Ae] ]
\]

\{
pronoun\_resolve(\ resolved\_tuple,
[ [act,Act],[agt,Agt],[ae,Ae] ] )
\}

the references in the n-tuple that is formed are resolved in

\textit{pronoun\_resolve}. \textit{Resolved\_tuple} is the n-tuple returned after
resolution. If the n-tuple belongs to the body of the story then it is
stored in the story representation.

\{
\textit{softasserta}(\text{fact}(\text{resolved\_tuple}))
\}

However, if the n-tuple is derived from a sentence expressing the
moral, it is evaluated against the n-tuples of the story.

\{
\textit{trace\_path}(\text{resolved\_tuple})
\}

Function \textit{trace\_path} maps the n-tuple from the moral onto events
in the story.

The database for the story consists of a list of Facts which are n-tuples
from the sentences in the story. The Facts are stored such that the most recent
Fact is the first element in the list.
The noun and pronoun lexicon in the grammar have the features, number and animate/inanimate, defined. These features are essential in helping to avoid establishing the wrong reference.

* e.g. A flock of pigeons lived in dread of a hawk who preyed upon them. They devised various measures for self-protection.

* They should not refer to hawk, but to pigeons by agreement in number.

### 3.4 Resolution of References

The reference resolution is done with the help of Alterman's coherence relations. An n-tuple, when formed, is sent as input to this portion of the program to have its references resolved.

The database for the story consists of a list of n-tuples, the most recent one being the most accessible member of the list. A last-in-first-out list is useful for resolving references because more likely than not pronouns in a n-tuple refer to the most recent n-tuples.

Therefore, in order to resolve references in a given n-tuple its relations with the previous n-tuples are checked. If a coherence relation exists, then, the constraints between the n-tuples are used to make the reference. If a relation does not exist, then the references are resolved by taking the first n-tuple whose properties match the given n-tuple's references.
Alterman's seven relations are:

```
coherence relations
    /\                  /
   taxonomic  partonomic  temporal
   |  \                |
class/ subclass sequence/ subsequence coord- before after
       \                |
       \                |
antecedent prece- dent
        \            rquent
        \           sequel
```

Some examples of these relations are:

**Rule1:**

```
relation(antecedent,[[act,roar],[R1],
                   [[act,open],[R2],[[agt,A]]]) :-
same(R1,R2,[[agt,A]]),
present(R2,[ae,mouth]),
check-feat(A,animate).
```

These says that antecedent of *roar* is *open* one's *mouth*. The *agt* of roar and open mouth should match—this constraint is specified by the fourth argument of the *relation*. *Same* checks two n-tuples to see if they have a certain common member. Here, R1 and R2, the two sublists from *roar* and *open mouth* n-tuples are checked to see if they have a common agt.

*Present* checks if a n-tuple contains a certain member. Here, R2, the sublist from *open* is checked to see if the affected entity is *mouth*. *Check-feat* checks if *agt* of *roar* and *open* is animate.
Rule2:

\[
\text{relation ( subclass,[[act,communicate]]R1,} \\
\text{[[act,bark]]R2,[[agt,A]]} \) :- \\
\text{same(R1,R2,[[agt,A]]).}
\]

*Bark* is a subclass of *communicate* and both have the same *agt.*

**Rule3:**

\[
\text{relation(coordinate,[[act,carry]]R1,} \\
\text{[[act,have]]R2,[[agt,A1],[ae,A2]]} \) :- \\
\text{same(R1,R2,[[agt,A1]]),} \\
\text{same(R1,R2,[[ae,A2]]).}
\]

A coordinate relation exists between carrying an object and having an object because one must have an object while carrying it.

When a relation between two tuples is being examined, the relation may not be directly found. The two n-tuples may be connected via other n-tuples. Thus, there are two ways of establishing relation between two n-tuples: they may be directly connected by a link or they may be indirectly connected via other n-tuples.

*Exist-conn* is called with two n-tuples to establish the link:

\[
\text{exist-conn} ( \text{Snt1, Snt2, L, Conn} ) :- \\
\text{exist-conn1} ( \text{Snt1, Snt2, L, Conn} ).
\]

*Exist-conn1* checks if there is a direct link. Snt1, and Snt2 are the two n-tuples; L is the list of constraints, such as same agt. "Conn" is the name of the relation between the n-tuples.

\[
\text{exist-conn} ( \text{Snt1, Snt2, L, Conn} ) :- \\
\text{exist-conn2} ( \text{Snt1, Snt2, L, Conn} ).
\]
Exist-conn2 checks if an indirect link exists between the n-tuples via other n-tuples. It is defined as:

exist-conn2 ( Snt1, Snt2, L, _ ) :-
exist-conn1(Snt1, Snt3, L, _),
exist-conn(Snt3, Snt2, L2, _).

It finds a direct relation between Snt1 and another n-tuple, Snt3. Then, it checks to see if there exists a direct or indirect link between Snt3 and Snt2.

An example of a direct relation is the subclass link between communicate and bark.

Examples of indirect relation are:

e.g., carry is a coordinate of have. Precedent of have is find. Thus, carry is related to find, indirectly.
e.g., open mouth is precedent of talk and bark because open mouth is precedent of communication; and talk and bark belong to a subclass of communication.

By inheritance, talk and bark share the same precedent.

For reflexive pronouns, Alterman’s relations are not used. They are resolved syntactically from the references in the same sentence.

3.5 EVALUATION

At the clausal level, as the moral is parsed and each n-tuple is formed, it is sent to the part of the program that evaluates n-tuples. The list of n-tuples from the story is evaluated based on this n-tuple from the moral to see if there exists a sequence of events satisfying this part of that moral.

To take an example,
Pretenders betray themselves by talking too much

The first n-tuple formed while parsing is

$$([\text{act}, \text{betray}],[\text{agt}, \text{pretender}],[\text{ae}, \text{pretender}]).$$

Actions and descriptive common nouns are evaluated separately. Actions were found to be easier to describe than descriptive nouns. Thus, it is easier to deduce an action of betrayal than it is to deduce who is a pretender.

Evaluation proceeds in the following manner: First, only the act of betrayal is deduced, that is, the n-tuple

$$([\text{act}, \text{betray}],[\text{agt}, \text{X}],[\text{ae}, \text{X}]) \quad (1)$$

is evaluated and the value of X is instantiated. Second, whether X is a pretender is verified. Thus, we evaluate

$$([\text{pred}, \text{be}],[\text{patient}, \text{X}],[\text{condition}, \text{pretender}]) \quad (2)$$

The value of X, being known, constrains the search. It is easier to deduce if a certain X is a pretender rather than who is a pretender?

The next tuple from the moral is

$$([\text{act}, \text{talk}],[\text{agt}, \text{pretender}]) \quad (3)$$

However, this n-tuple is dependent on n-tuple (1) in the sense that they
both have the same agent. By now, however, the sequence of events leading to n-tuple (1) is known. Since agent of betray and talk must be the same, the search for events satisfying talk is constrained by our knowledge of the value of X. Thus, to deduce talk we check if ([act,talk],[agt,X]) is true in the story.

Thus, we see there are two ways to evaluate a n-tuple from the moral: directly as in evaluation of (1), or by using the constraints imposed on it by previously satisfied n-tuples, as in the case of (3).

Trace_path is called to evaluate the n-tuples from the moral. It has two different argument lists depending on the evaluation method used. In one instance, trace_path is called with one n-tuple to evaluate it independently. In the second instance, trace_path is called with two n-tuples, the first being the evaluated n-tuple and the second being the n-tuple to be evaluated using the constraints imposed on it by the first one.

In the case of independent evaluation:

\[\text{trace\_path ( Moral ) : -}\]

\[\text{deduce the sequence of events in the story which satisfy the moral} \]

\[\text{deduce ( Moral , Sequence ) , softasserta ( path ( [Moral | Sequence] ) ).} \]

each n-tuple from the moral is stored with its corresponding sequence of events as a predicate called Path
In the case of interdependent evaluation, as in the case of n tuples (1) and (2) previously described:

\[[\text{act,betray}],[\text{agt,X}],[\text{ae,X}]\]  \hspace{1cm} (1)

\[[\text{act,talk}],[\text{agt,X}]\]  \hspace{1cm} (2)

At this point (1) has already been evaluated and its sequences stored. (2) needs to be evaluated and its evaluation is constrained by the fact that the agent of (1) must be the agent of (2). So, evaluation begins with:

\textit{trace\_path} ( Tuple1, Tuple2, Conn ) :-

Tuple1 is the n tuple (1), Tuple2 is the n tuple (2) and Conn is the list of constraints, in this case \[\text{agt}\]. The \textit{path} of Tuple1 is recalled

\textit{path} ( [Tuple1|Sequence] ),

Now a Prolog procedure is called to look into the \textit{path} and extract the value of \textit{agt}. Then, the \textit{agt} slot of Tuple2 is instantiated by this value. Now, the sequence of events in the story which satisfy Tuple2 is deduced and stored

\textit{deduce} (Tuple2 , Sequence2 ),

\textit{softasserta} ( path ( [Tuple2|Sequence2] )).
3.6 CLOSER LOOK AT THE EVALUATION PROCESS

One of the stories is as follows:

One day, a donkey found a lion's skin in the forest. He flung it over his shoulders and swaggered about the forest, roaring and growling. There were a few foolish animals who were terrified by the noise that he made. But, the sly fox knew the donkey by his bray.

-- Pretenders betray themselves by talking too much.

The n-tuples from the story are:

[[act,find],[agt,donkey1],[ae,skin1],[loc,forest1]]

[[pred,poss],[agt,lion1],[ae,skin1]]

[[act,fling],[agt,donkey1],[ae,skin1], [loc,shoulders1]]

[[pred,poss],[agt,donkey1],[ae,shoulders1]]

[[act,swagger],[agt,donkey1],[loc,forest1]]

[[act,roar],[agt,donkey1],[loc,forest1]]

[[act,growl],[agt,donkey1],[loc,forest1]]

[[pred,be],[agt,animals1],[condition,terrified], [by,noise1]]

[[pred,be],[agt,animals1],[condition,foolish]],

[[act,make],[agt,donkey1],[ae,noise1]]

[[pred,be],[patient,fox1],[condition,sly]]
[[act,know],[agt,fox1],[ae,donkey1],[by,bray]]

[[pred,poss],[agt,donkey],[ae,bray]]

The n-tuples from the moral are:

[[act,betray],[agt,pretender],[ae,pretender]]

[[pred,be],[patient,X],[condition,pretender]]

[[act,talk],[agt,pretender]]

Now, the next step is the tracing of events in the story to satisfy each n-tuple of the moral.

As mentioned earlier, actions are easier to evaluate than descriptive nouns. Thus, it is easier to trace the act of betrayal than it is to deduce the description of a pretender.

The evaluation begins as soon as the first n-tuple is formed

[[act,betray],[agt,pretender],[ae,pretender]] --- (1)

The *agt* and *ae* values are descriptive nouns. Thus, they need to be evaluated separately. This is postponed till the action of betrayal is evaluated. Since the values of *agt* and *ae* are identical, the act of betrayal is one of self-betrayal. The n-tuple to be evaluated is therefore

[[act,betray],[agt,X],[ae,X]] --- (1)
where \( X \) is a variable to be instantiated from events in the story.

An act of betrayal by \( X \) necessarily requires another character \( Y \) to recognize \( X \) in an act or by some physical feature. Therefore,

- If \( Y \) perceives \( X \) by a physical feature, then, the story must indicate that \( X \) possesses that feature.

- If \( Y \) perceives \( X \) by an act (Act1), then in the story, \( X \) must have performed that particular act (Act1) or another act (Act2) such that Act1 and Act2 are in some way related.

Thus, when evaluation begins

\[\text{deduce} ( [[\text{act, betray}], [\text{agt, } X], [\text{ae, } X], \text{Sequence}]:- )\]

First, the story is checked to see if there is a character \( Y \) who recognized another character \( X \) in an act or by some physical feature. We see that the fox recognizes the donkey when the fox perceives the noise the donkey makes and knows the donkey by his bray. Second, we see by Alterman's relations that bray is a form of communication and the consequence of bray is to make noise. We check the story again to see if the donkey brays. It doesn't bray, however, it roars and growls, which are also forms of communication and have the desired effect of making noise.

Thus we deduce that the donkey betrays himself to the fox by his bray, when he makes noise by roaring and growling. We store

\[ [[\text{act, betray}], [\text{agt, donkey}], [\text{ae, donkey}]] \]

followed by the trace just described in a predicate called Path.
The second step in evaluation is to trace the events to see if the donkey satisfies the role of the descriptive noun, 'pretender'. A character can be a pretender either by an action he performs or by an appearance he adopts. Thus, we trace

\[ \text{deduce} \ (\text{Sequence,} \nonumber \\
\quad \text{[[pred,be],[agt,donkey],} 
onumber \\
\quad \text{[ae,pretender]]):} \nonumber \]

The two possibilities explored are:

(1) \emph{deduce\_appearance}:

The story is checked to see if the donkey attempts to look like another character, Z. We see that the donkey flings a skin on his shoulders. \emph{Skin} is a form of clothing and \emph{shoulders} is a part of a body used to wear clothing. We also see that the skin belonged to a lion. Thus, when the donkey places clothing belonging to a lion around his body, he is possibly pretending to be a lion, by wearing the clothing of a lion.

(2) \emph{deduce\_action}:

The story is checked to see if the donkey performs an action which is known to be performed by a lion. The donkey roars and 'roar' is a form of 'communication' performed by an animate agent, whose default role is lion. Thus, the donkey pretends to be a lion by roaring.
Now, the path for donkey being a pretender is stored.

The third step in evaluation occurs as the rest of the moral is parsed to give us the n-tuple

\[
\text{[[act,talk],[agt,pretender]]} -- (3)
\]

The agent of \textit{talk} must be same as that for \textit{betray}. So, now our next step is to evaluate

\[
\text{deduce} \left( \text{[[act,talk],[agt,donkey]], Sequence} \right)
\]

Talk is a form of \textit{communication} and in the story the donkey \textit{roars} and \textit{growls} which are also forms of \textit{communication}. So, we store this third Path.

An important point to note is the fact that the process of evaluation is made sturdier by computing it in an interdependent manner, using constraints between n-tuples from the moral. Thus the constraint that all the three n-tuples must have the same agent must be satisfied before the moral is concluded to be satisfied.

The paths stored during the process of evaluation are as follows:

\[
\text{path ( \{ [[act,betray],[agt,donkey],[ae,donkey]] , }

\{\text{conseq , [[act,growl],[agt,donkey]},

\{\text{loc,forest]} ,

\{[[act,make],[agt,donkey]},

\{ae,noise] \})
\]
[subseq, [[act, recognize], [agt, fox],
[ae, donkey]],
[[act, perceive], [agt, fox],
[ae, noise]],
[[act, know], [agt, fox],
[ae, donkey], [by, bray]]]

path ( [ [[pred, be], [patient, donkey],
[condition, pretend]],
[operator, pretend], [pred, be],
[agt, donkey], [ae, lion]],
[ [[act, wear], [agt, donkey], [ae, skin]],
[[pred, poss], [agt, lion], [ae, skin]],
[[act, roar], [agt, donkey], [loc, forest]]])

(Note: The term *operator* applies to those verbs which take infinitives as complements:

* e.g. He pretended to be a lion.
* He intends to be a doctor.)

path ( [ [[act, talk], [agt, donkey]],

[conseq, [[act, growl], [agt, donkey],
[loc, forest]],
[[act, talk], [agt, donkey]]])
3.7 GENERATION OF OUTPUT

Each list of n-tuples denoted by Path is converted into English sentences.

In each list stored in Path, there are either sublists of single n-tuples or of more than one n-tuples connected by a relation. Each member of a sublist is converted into a simple or compound English sentence depending on whether it contains single n-tuples or related n-tuples.

The features of the generation grammar are:

1) If a member of a sublist is a list of n-tuples connected by a relation then each n-tuple is converted into a sentence. Then a compound sentence is made by connecting the simple sentences with appropriate connectives.

2) If the member is a single n-tuple then the following member in the sublist is examined to see if the two members can form a compound sentence. If the second member is also a single n-tuple then
   i) if the two n-tuples contain the same value for 'agt' then they could be made into an elliptical sentence.

   ii) If the 'agt' value of the first n-tuple is the same as the 'ae' value for the second n-tuple then the second n-tuple is made into a relative clause for the first.

For example,
can be:

The donkey wore a skin.
The skin belonged to a lion.

(Note: 'belong to' is the translation for the possessive case.)

or,

The donkey wore a skin that belonged to a lion.

3) Else, the member is converted into a simple sentence.

The output for the given story is:

The donkey betrayed himself. He made noise because he growled in the forest. The fox recognized the donkey when he perceived the noise and knew him by the bray. The donkey is a pretender. He pretended to be a lion when he wore the skin that belonged to a lion, and roared in the forest. The donkey talked because he growled in the forest. The donkey is the pretender who betrayed himself by talking.
Chapter 4

TASKS AND FEATURES

4.1 AN EXAMPLE

After a look at another example, I shall attempt to summarize the tasks and the features of the computational process.

Fable:

A dog carrying a bone in his mouth was crossing a bridge. He caught sight of his reflection in the smooth water. But, he thought it was another dog and greedy for its bone began to bark. As soon as he opened his mouth, his bone dropped into the water and was lost forever.

-- One should take care of ones possessions rather than be greedy for those of others.

The n-tuples from the story are:

[[act,carry],[agt,dog1],[ae,bone1],[loc,mouth1]]

[[pred,poss],[agt,dog1],[ae,bone1]]

[[act,cross],[agt,dog1],[ae,bridge1],[loc,water]]
The n-tuples from the moral are:

[[act, take-care], [agt, one], [ae, possessions1]]

[[pred, be], [patient, one], [condition, greedy], [for, possessions2]]

[[pred, poss], [agt, others1], [ae, possessions2]]

In the moral, the author suggests an alternative world as opposed to the
world described in the narrative. In the alternative world one should take care of one's possessions as opposed to being greedy for those of others. The alternative world is thus not represented in the narrative, but rather, its inverse is. That is, there is a character in the narrative who does not take care of his possessions. The notion of a hypothetical alternative world is conveyed by the author's use of a modal operator in conjunction with the connective 'rather than'. In sentences of this nature, the proposition preceding the connective and containing the modal operator belongs to the hypothetical world and the proposition following the connective belongs to the real world. Thus, the proposition of one being greedy is represented in the world of the narrative. Since the evaluation process is interleaved with the grammatical analysis, it is possible to guide the process of evaluation with syntactic cues of the form mentioned above.

Evaluation of the first n-tuple of this moral proceeds in a slightly different manner. Instead of mapping the act of "taking care of one's possessions", the system tries to map the inverse of the act, that is the act of "not taking care of ones possessions". The n-tuples of the narrative are checked to see if there is a character X, who possessed an object Y, but later due to his own actions loses his possession Y. Indeed there is a dog who possesses a bone. But, he barks, causing his mouth to open, and thus causing the bone to drop. As a result of dropping the bone from his mouth, he loses the bone.

The second and third n-tuples are evaluated in a direct manner, as they are represented in the narrative.
4.2 SUMMARY OF TASKS AND FEATURES

As discussed in earlier chapters, the mapping process is closely knit with the process of grammatical analysis, in this computational study. The close integration between the two processes allows syntactic cues and constraints to be used in facilitating and constraining the mapping process. This is an important computational feature because it simplifies the task and at the same time reflects the natural process of reading.

In section 2.6, I had introduced Martha Palmer's work on "Inference-Driven Semantic Analysis" for the reason that I discovered analogies between the computational method used to map a moral onto a fable in this thesis, and her methods of semantic analysis in natural language.

As an n-tuple from a moral is mapped onto a fable, some of the tasks that need to be performed are similar to hers, described in Section 2.6:

1) fill in the arguments of the semantic predicate of the n-tuple from the moral, with the help of inference rules.
   
   e.g. [[act,betray],[agt,X],[ae,X]] ->
       
       { Rules to infer the value of X and
         instantiate the slots of the predicate}

2) expand each representation of a semantic predicate by encoding inference rules in a procedural form. The rules mentioned in (1) serve to expand the meaning of the semantic predicate, thus providing more detailed information on it. As mentioned above, they also facilitate the instantiation of the arguments of the predicate.
e.g. \([\text{act,betray}],[\text{agt},X],[\text{ae},X]\) ->

{ Rules which expand the meaning of
betray to reveal more detail, at the
same time facilitating the instantiation
of \(X\). }

3) constrain the application of inference rules.
e.g. Deduce \([\text{act,betray}],[\text{agt},X],[\text{ae},X]\) ->

If i) Character \(Y\) recognizes \(X\) by an act
or some physical feature

and ii) Character \(X\) must have performed that
act or possessed that feature in the
narrative.

Thus, the application of (ii) is constrained by the result achieved by (i)
because the character \(X\) in (ii) must be the same \(X\) that application of (i) returns.

4) integrate final representation of clauses into prior clauses.
e.g. After having deduced

\([\text{act,betray}],[\text{agt, donkey}],[\text{ae, donkey}]\),

when the n-tuple \([\text{act,talk}],[\text{agt},X]\) is evaluated, the knowledge that the
two n-tuples have the same \textit{agt} integrates the evaluation of the second n-tuple to
that of the first. Thus, \(X\) in \([\text{act,talk}],[\text{agt},X]\) is instantiated with \textit{donkey} and
evaluation proceeds.

Thus each semantic predicate of the moral with its argument slots is
expanded by a procedure of domain-specific inference rules which in addition to
giving greater detail to the meaning of the predicate, also infer the values of the
arguments.
Chapter 5

CONCLUSION

5.1 A PERSPECTIVE

The concept of macrostructure in a text can be looked at from two perspectives. One perspective that has been the concern of this thesis is that of interpreting the events in the text in terms of the macrostructure assigned to it by the author. Another perspective would be the reverse process and concerns the reader's gradual construction of the macrostructure as he/she reads and interprets the text.

This thesis attempted to show that when a reader maps a macrostructure onto a text he/she begins the mapping process as soon as he/she encounters the first proposition of the macrostructure. The reader does not wait till the whole macrostructure has been read to begin mapping.

The process of building a macrostructures can be seen to be analogous to the mapping process because the reader does not wait till the end of the text to infer its theme. As soon as he/she reads the first line of the text, the process of inference begins. Thus, the reader explores potential macrostructures as he/she reads each line of the text.
5.2 OBSERVATIONS DURING COMPUTATION

Macrostructures by nature must express macroconcepts. Each macroconcept provides us with a conceptual overview of those microconcepts which it subsumes. It is instructive to examine examples of macrostructures in the form of morals, adages or summaries because such a study reveals a language that facilitates the expression of macroconcepts. That is, macrostructures must employ those predicates and objects which are conducive for the expression of macroconcepts.

It appears that there exist in natural language semantic predicates and objects which are macros by nature. However, their function with respect to the text can be varied. To give a few examples:

1) a macropredicate or object can be interpretive in function. Their function is to lend a macro-interpretation to a series of micropredicates or objects.

   e.g. "betray", "imitate" are macropredicates which interpret the essence of series of micropredicates.

   "pretender" and "tyrant" are macroobjects which interpret the nature of an object.

   The meaning of an interpretive macropredicate is bound and defined by the micropredicates it subsumes. It is possible to find alternative definitions for interpretive macropredicates and thus, it is important to preserve the particular series of micropredicates in order to retain the definition.
2) a macropredicate or object can be culminative in function. They represent the culmination of a series of micropredicates.

  e.g. "soak and scour dishes" culminates in "clean dishes".
  "lighting" and "puffing" a cigarette culminates in "smoking" cigarette.

  The meaning of a culminative macropredicate is not really bound to its micropredicates. The macropredicate possesses only one definition and is not dependent on its micropredicates for its definition. Thus, it is possible to delete a series of micropredicates and substitute for it its culminative macropredicate.

3) a macropredicate can be taxonomomic in function. The function is to classify micropredicates.

  e.g. "communicate" is a class to which "talk", "bark" belong.

  e.g. "skin" (of animals) is a form of clothing.

  A taxonomic macropredicate can be substituted for its associated micropredicate.

  During the mapping process, the nature of the search through micropredicates in order to satisfy a macropredicate varies with the nature of the macropredicate:

  \[ M = \text{macropredicate}. \]

  \[ m[i] = \text{micropredicate, } i = 1,2,3,\ldots,n \]
1. An interpretive macropredicate, M, as mentioned earlier, can be defined in more than one way by different series of micropredicates. Thus M can subsume series

\[ s_1 = m[i]...m[j] \]
\[ s_2 = m[k]...m[l] \]

where \( \text{Intersection}(s_1, s_2) \) may not be empty
and \( s_1 \) is not identical to \( s_2 \).

This can also be represented as:

\[
\begin{array}{c}
M \\
* \\
| \\
* \\
| \\
* \\
\end{array}
\]

where * = a micropredicate.

During mapping of M it was necessary to explore all the paths possible since the meaning of M is closely bound to the series it represents. Successful traversal of one or more paths can conclude the existence of M. It is necessary that all paths be explored, not satisfied.

2. A culminative predicate, M, is a culmination of a series of micropredicates. Thus M subsumes a series

\[ s = m[i]...m[j] \]
In order for $M$ to be true $m[i]...m[j]$ must take place in their totality. Thus it is necessary to satisfy all the sons of $M$.

3. A taxonomic macropredicate, $M$, represents a class of micropredicates. The presence of one such micropredicate is enough to conclude the existence of $M$.

As soon as one micropredicate is satisfied, the search can conclude $M$.

The study of macrostructures provides us with a vocabulary of the type of macroconcepts possible and the nature of their structure.

### 5.3 ALONG THIS LINE OF STUDY

The computational study here has been a top-down process, travelling from macrostructures downwards into the microstructures of the text. As discussed in the previous section, the nature of the paths traversed depends on the nature of the macroconcepts. A study of more varied texts should yield a broader vocabulary of the types of macroconcepts possible.

Just as the structural nature of a macroconcept affects the top-down
search during evaluation, so it must the reverse bottom-up process of
construction. A broader vocabulary of the structural types of macroconcepts will
help reveal some heuristics to guide the bottom-up process. The purpose of
computing this process would be to reflect the reader’s gradual construction of
a macrostructure for a text as he/she reads it.
References

A system of Seven Coherence Relations for Hierarchically
Organizing Event Concepts in Text.
PhD thesis, Department of Computer Science, The University of
Texas at Austin, 1982.

[Alvarado,Dyer,Flower 85]
Sergio Alvarado, Michael Dyer, Margot Flower.
Memory Representation and Retrieval for Editorial
Comprehension.

[August,Dyer 85] Stephanie August, Michael Dyer.
Analogy Recognition and Comprehension in Editorials.

Some effects of titles on Building and Recalling.

A Model for Understanding the Points of Stories.

In-Depth Understanding, a Computational Model for Narrative
Comprehension.

[Enkvist 82] Nils Erik Enkvist.
Report on Topic : Aspects of Text Analysis & Generation.
In Sture Allen (editor), Text Processing : Text Analysis &
Generation, Text Typology & Attribution. Almqvist &

Integrating Verb Meanings into Context.

The Thread of Discourse.
How is a text like a clause?
In Sture Allen (editor), Text Processing : Text Analysis &
Generation, Text Typology & Attribution. Almqvist &

Abstracting Main Ideas from Simple Technical Prose.

A Model of Reader Strategy for abstracting main ideas from
Simple Technical Prose.

Local and Global Structures in Discourse Understanding.
ACL Proceedings, 1983.

[Langleben 81] Maria Langleben.
Latent Coherence, Contextual Meanings, Interpretation of a
Text.
Text 1-3, 1981.

[McCutchen,Perfelt 82]
Deborah McCutchen, Charles A. Perfelt.
Coherence and Connectedness in Discourse.

[Miller,Kintsch 81]
James Miller, Walter Kintsch.
Knowledge-based aspects of Prose Comprehension.
Text 1-3, 1981.

[Norvig 83] Peter Norvig.
Six Problems for Story Understanding.

Inference-Driven Semantic Analysis.

[Reiser,Black 82] Brian J. Reiser, John B. Black.
Processing and Structural Models of Comprehension.
[Schank 82] Roger Schank. 
Representing meaning: an AI Perspective. 

[Simmons 83] Robert F. Simmons. 
Computations from the English. 
Prentice-Hall, 1983.

Thematic Knowledge, Episodic Memory & Analogy in MINSTREL, A Story Invention System. 

[Vandijka 80] T.A. Van Dijk. 
Macrostructures. 

[Vandijkb 85] T.A. Van Dijk. 
Strategic Discourse Comprehension. 

Recalling and Summarizing Complex Discourse. 

[Vandijkd 85] T.A. Van Dijk. 
Handbook of Discourse Analysis. 