COMPUTERIZED THEMATIC CONTENT ANALYSIS

Catherine L. Hausmann
John Seely Brown

TECHNICAL REPORT #45- JUNE 1973

We wish to thank Dr. Louis A. Gottschalk, Chairman of the Department of Psychiatry and Human Behavior, for providing financial assistance, moral support and intellectual guidance for this project.

TABLE OF CONTENTS

Introduction

This report describes a feasibility study of using state of the art techniques in natural language processing and Artificial Intelligence to perform thematic content analysis. The first part of this report contains a description of the thematic scale chosen for this investigation followed by descriptions of the parser and scoring algorithms. Considerable attention is given to some of the more severe problems we encountered in order to indicate some of the potential pitfalls that face any comprehensive computer system attempting to do thematic content analysis.

The Gottschalk-Gleser Scoring Method

The Gottschalk-Gleser Content Analysis Method attempts to measure the intensity of certain psychological states, among them, hostility, anxiety, and hope. The standard procedure has been to ask the subject to talk for five minutes about any interesting or dramatic personal life experience. The interview is recorded, transcribed and then given to a technician to analyze. He performs his analysis by taking the interview one clause at a time comparing each clause to the categories listed in the scale for the psycholgical state he is interested in. The scale for Hostility Directed Outward is listed below (there are also scales for Hostility Directed Inward and Ambivalent Hostility.)

TABLE 1

Hostility Directed Outward Scale: Destructive, Injurious, Critical Thoughts and Actions Directed to Others

(I) Hostility Outward—Overt		(II) Hostility Outward—Covert	
Thematic Categories		Thematic Categories	
a 3*	Self killing, fighting, injuring other individuals or threatening to do so.	*a* 3*	Others (human) killing, fighting, injuring other individuals or threatening to do so.
b 3	Self robbing or abandoning other individuals, causing suffering or anguish to others, or threatening to do so.	*b* 3	Others (human) robbing, abandoning, causing suffering or anguish to other individuals, or threatening to do so.
c 3	Self adversely criticizing, depreciating, blaming, expressing anger, dislike of other human beings.	*c* 3	Others adversely criticizing, depreciating, blaming, expressing anger, dislike of other human beings.
a 2	Self killing, injuring or destroying domestic animals, pets, or threatening to do so.	*a* 2	Others (human) killing, injuring, or destroying domestic animals, pets, or threatening to do so.
b 2	Self abandoning, robbing, domestic animals, pets, or threatening to do so.	*b* 2	Others (human) abandoning, robbing, domestic animals, pets, or threatening to do so.
c 2	Self criticizing or depreciating others in a vague or mild manner.	*c* 2	Others (human) criticizing or depreciating other individuals in a vague or mild manner.
d 2	Self depriving or disappointing other human beings.	*d* 2	Others (human) depriving or disappointing other human beings.
		e 2	Others (human or domestic animals) dying or killed violently in death-dealing situation or threatened with such.
		f 2	Bodies (human or domestic animals) mutilated, depreciated, defiled.
a 1	Self killing, injuring, destroying, robbing wildlife, flora, inanimate objects or threatening to do so.	*a* 1	Wildlife, flora, inanimate objects, injured, broken, robbed, destroyed or threatened with such (with or without mention of agent).
b 1	Self adversely criticizing, depreciating, blaming, expressing anger or dislike of subhuman, inanimate objects, places, situations.	*b* 1	Others (human) adversely criticizing, depreciating, expressing anger or dislike of subhuman, inanimate objects, places, situations.
c 1	Self using hostile words, cursing, mention of anger or rage without referent.	*c* 1	Others angry, cursing without reference to cause or direction of anger; also instruments of destruction not used threateningly.
		d 1	Others (human, domestic animals) injured, robbed, dead, abandoned or threatened with such from any source including subhuman and inanimate objects, situations (storms, floods, etc.).
		e 1	Subhumans killing, fighting, injuring, robbing, destroying each other or threatening to do so.
		f 1	Denial of anger, dislike, hatred, cruelty, and intent to harm.

* The number serves to give the weight as well as to identify the category. The letter also helps identify the category.

Given the clause "I burned a hole in the sofa", a scorer using the Hostility Scale in Table 1 would mark the clause IA1 or "self injuring wildlife, flora, or inanimate objects or threatening to do so." The Roman numeral in the score identifies the hostility as being overt rather than covert. The letter helps identify the category. The number 1 also helps identify the category along with being the weight assigned to the clause. After the technician has gone through all the clauses in the sample he tabulates a final score. For details on how the final score is obtained and for more information on how these scales are used, we refer the reader to The Measurement of Psychological States Through the Content Analysis of Verbal Behavior, Gottschalk and Gleser, 1969, and Manual of Instruction for Using the Gottschalk-Gleser Content Analysis Scales, Gottschalk, Winget and Gleser, 1969.

Task

Before a computer can successfully mimic the action of a human scorer using the Gottschalk-Gleser Content Analysis Method it must derive a fairly sophisticated representation of the meaning of its input. This is in contradistinction to traditional schemes for computer-driven content analysis which have found it sufficient to utilize pattern matches against particular words (with and without inflectional

endings removed) or against word clusters in order to characterize the essential meaning of its input.

In the Gottschalk-Gleser Method the presence of a single word or group of words is seldom sufficient for characterizing what score a clause is to receive. Instead, a considerably more detailed analysis of the sentence is mandatory, requiring that the action or theme of the clause along with the actor and recipient of this action be identified. Once these structural elements are isolated then each can be further analysed as to the properties it possesses, and then a score for the clause can be determined.

An additional syntactic piece of information that must be gleaned from the structure of a sentence before it can be scored is the existence and scope of a negative. The critical part a negative plays is illustrated by the sentence, "My anteater does not like the red ants." Here the verb "to like", normally a non-hostile, non-scorable verb in the Hostility Scale, becomes scorable with the addition of the negative "not". A further example is the sentence, "We don't hate bears." This sentence would be scorable without the negative present as self expressing dislike of pets (IB1), but receives a different score with the negative present, denial of dislike, hatred, (IIF1).

The scope of the negative becomes important in more complex sentences like, "He wasn't telling the truth when he said that I like my congressman." If we were looking only at word clusters like "I like my congressman", we would miss the true intent of the sentence. We must in fact discern the relationships between these clusters; that is, we must find the scope of the negative in the first part of the sentence and discover it effect on the meaning of the second part. (In this example, it would allow the sentence to be scored IC3, self expressing dislike of others).

In developing a parsimonious theory for how to apply the Gottschalk-Gleser Content Analysis Method the above considerations (plus others) convinced us that we must first derive a description of a clause's syntactic deep structure before attempting to apply our content analysis scheme to it. In so doing, we hoped to avoid the fantastic proliferation of special patterns that would otherwise be needed to capture the various syntactic nuances of sentences.

The Parser

What was needed was an efficient parser that contained a sufficiently powerful grammar to characterize the subset of English that is often encountered in psychiatric

interviews. The parser chosen for this task is an experimental version of Woods' Augmented Transition Network Parser (197Ø). This parser was translated into UCI LISP (Bobrow et al, 1973) and was slightly modified so as to run on a medium size PDP-1Ø. Its grammar was enlarged to cover certain linguistic constructions that frequently occurred in our domain of discourse and a special core resident dictionary (of several hundred entries) was created for the interviews constituting our data base. (See Appendix 2 for a description of changes made to the Woods' parser.)

The parser creates a complete syntactic description of a sentence. As an example of its output, let us look at a parse for the sentence "He tore up the calendar page for the new month:

```
(SENTENCE* DECLARATIVE
           (NP   (PRONOUN he)
                 (NUMBER SINGULAR))
           (AUXILIARY (TENSE PRESENT))
           (VP   (VERB tear-up)
                 (NP (DETERMINER the)
                     (ADJ (NP (NOUN calendar)))
                     (NOUN page)
                     (NUMBER SINGULAR)
                     (PREP-PHRASE (PREPOSITION for)
                                  (NP (DETERMINER the)
                                      (ADJECTIVE new)
                                      (NOUN month)
                                      (NUMBER SINGULAR))))))
```

*abbreviations output by the parser have been expanded for readability

The parse may be thought of as forming a tree with SENTENCE at the top having four branches--DECLARATIVE (the sentence type); NOUN-PHRASE (the subject); AUXILIARY (tense); VERB-PHRASE (includes the verb, the object, and any verb modifiers). Input sentences in passive voice are transformed into active voice and hence receive the indentical structural description the active voice sentence receives. For example the following sentences receive the same syntactic description:

"Spinach is not liked by many children."
"Many children do not like spinach."

```
(SENTENCE DECLARATIVE
          NEGATIVE
          (NOUN-PHRASE  (ADJECTIVE many)
                        (NOUN child)
                        (NUMBER PLURAL))
          (AUXILIARY    (TENSE PRESENT))
          (VERB-PHRASE  (VERB like)
                        (NOUN-PHRASE  (NOUN spinach)
                                      (NUMBER SINGULAR))))
```

Although the two examples above represent only simple sentences (i.e. no embedded sentences, etc.) the parser is capable of handling substantially more complex sentences. However, at the present, our semantic routines for

interpreting (scoring) the output of the parser are restricted to handling non-embedded sentences.

Interpreting the Parse

In the Gottschalk-Gleser Scoring Method the score that a sentence receives depends on the action verb in conjunction with the noun phrases that function as actors and recipients of this action. In addition to isolating these grammatical constituents we must provide techniques for assigning "meaning" to each of these constituents. Fortunately, a relatively straightforward application of semantic features appears to suffice for this operation.

Verbs are assigned features called Verb-types (V1-V8) which are listed below:

Table 2

Verb-types

V1: Causing death or physical injury.
V2: Causing suffering or anguish; robbing; abandoning.
V3: Expressing adverse criticism, anger, blame, dislike, depreciation.
V4: Expressing vague or mild criticism or depreciation.
V5: Depriving; disappointing.
V6: Denial of anger, dislike, hatred, cruelty and intent to harm.
V7: Cursing.
V8: Dying or killed violently

These verb-types have been formulated from the Hostility Directed Outward Scale of the Gottschalk-Gleser Content Analysis Method. (See Table 1.)

In order to determine which verb-type a verb is to receive, the infinitive form of the verb is retrieved from the parse and looked up in a feature dictionary. Note that the parser performs the required morphological analysis for stripping the inflectional endings of the verb yielding its infinitive or root form.

Assigning semantic features to the noun phrases that function as actors and recipients is slightly more problematic. In the simplest case the noun phrase receives the semantic features of its head noun (determined by accessing the semantic feature dictionary with that head noun). If, however, the head noun of a noun phrase has no "relevant" semantic features but has a modifying prepositional phrase which contains as its head noun a noun that has "relevant" semantic features, then these features are lifted up to function as the semantic markers for the initial noun phrase. Since this algorithm is completely recursive, it can be repeatedly applied to prepositional phrases that modify previous noun phrases in a frantic effort to find some noun in the complex noun phrase that has useable semantic features. A few examples of this technique will help to illustrate the above discussion.

"the little boy"

```
(NOUN-PHRASE (DETERMINER the)
             (ADJECTIVE little)
             (NOUN boy)
             (NUMBER singular))
```

Look up "boy" in feature dictionary.
= (+OTHER +ANIMATE)

"the side of the car"

```
(NOUN-PHRASE (DETERMINER the)
             (NOUN side)
             (NUMBER singular)
             (PREP-PHRASE (PREPOSITION of)
                          (NOUN-PHRASE (DETERMINER the)
                                       (NOUN car)
                                       (NUMBER singular))))
```

Look up "side" in feature dictionary.
It has no relevant features so go
to prepositional phrase modifying it.
Look up the head noun "car" in the
feature dictionary.
= (+INANIMATE)

"her face"

```
(NOUN-PHRASE (DETERMINER the)
             (POSSESSIVE (NOUN-PHRASE (PRONOUN she)
             (NOUN face)
             (NUMBER singular))
```

Look up "face" in feature dictionary.
It has the special feature (+PART)
which tells the program to look for a
preceding possessive, and to look up
the head noun of that possessive.
= (+OTHER +ANIMATE)

The Scoring Algorithm:

Our scoring algorithm must decide what the score for the clause (sentence) is to be after examining the set of features assigned to the above-mentioned noun phrases and the head verb. In order to piece together this information and to make the correct decision as to what the score for the clause should be, our scorer utilizes an information structure for advice on how to interpret its inputs. This structure is called the Scoretree and it is indexed by the verb-type. For example, the scoretree associated with the verb-type V1 (causing death or physical injury) is:

TABLE 3

First node	Second node	Thematic Code Symbol
+SELF	+OTHER	IA3*
	+PET	IA2
	+WILDLIFE +FLORA +INANIMATE	IA1
+OTHER	+OTHER	IIA3
	+PET	IIA2
+SUBHUMAN	+SUBHUMAN	IIE1
NIL**	+OTHER +PET	IIE2
+INANIMATE +WILDLIFE +FLORA	NIL**	IIA1
NIL	+WILDLIFE +FLORA +INANIMATE	IIA1

(See Appendix 1 for a description of all eight scoretrees.)

At present each scoretree has only two levels. The first

level is made up of tests for the features of the subject of the sentence in question. The second level is composed of tests for the object. The scoring algorithm utilizes the scoretrees in the following fashion. First it determines which scoretree to access by determining the verb-type of the head verb in the clause. It then chooses a first level node by overlapping the features of the subject (or actor) noun phrase with the features of successive first level nodes until some partial match is achieved. It then searches the second level nodes accessible from the current first level node overlapping the features of the object (or recipient) noun phrase with successive nodes until another partial match is achieved. Associated with the second level node is the score that the total clause is to receive. (Documentation of the functions used in scoring is listed in Appendix 3.)

In order to unify the various steps that our system goes through in scoring a sentence we summarize below the results of each step in scoring the sentence, "The garbage man hit my dog with a belt."

The sentence is parsed yielding the syntactic deep structure:

```
(SENTENCE DECLARATIVE
          (NP (DETERMINER the)
              (ADJ (NP (NOUN garbage)))
              (NOUN man)
              (NUMBER SINGULAR))
          (AUXILIARY (TENSE PAST))
          (VP (VERB hit)
              (NP (DETERMINER the)
                  (POSSESSIVE (NP (PRONOUN I)))
                  (NOUN dog)
                  (NUMBER SINGULAR)
                  (PREP-PHRASE (PREPOSITION with)
                               (NP (DETERMINER a)
                                   (NOUN belt)
                                   (NUMBER SINGULAR)))))))
```

The head verb, subject (actor) and object (possible recipient) are isolated:

```
Head verb: HIT
Subject: (NP (DET THE) (ADJ GARBAGE) (N MAN))
Object: (NP (DET THE) (POSS (NP (PRO I))) (N  DOG)  (NU
SG)) (PP (PREP WITH) (NP (DET A) (N BELT) (NU SG))))
```

Each of the above three constituents is assigned a set of semantic features:

```
Head verb: V1
Subject: (+OTHER +ANIMATE)
Object: (+PET +SUBHUMAN +ANIMATE)
```

The feature of the head verb is used to determine the pertinent scoretree, i.e. V1 (see Table 3 for actual scoretree).

a) The first level node is determined from the features on the subject.

b) The second level node is determined from the features on the object.

The score associated with the second-level node is output: IIA2.

By not integrating the scoretree information directly into the scoring algorithm, we can easily augment and modify this crucial information without having to modify the scoring algorithm. For example, by just changing the scoretrees we can use the exact same scoring algorithm for all the different Gottschalk-Gleser scales!

Some Results

It is difficult to quantify just how well our present computerized scoring method performs. In testing our method out on one-hundred sentences taken from The Manual of Instruction for Using the Gottschalk-Gleser Content Analysis Scales (1969), we were able to parse and score correctly approximately 60% of them. The sentences below are a representative sample of the kinds of sentences that were handled.

I sort of upset them with my hours of coming in at night.
[Scored ID2 -- self depriving or disappointing other human beings]

For the present we have made the assumption that pronouns like "he, she, him, her, they", etc. refer to humans. This is a gross assumption and could in some instances, be erroneous. Note the sentence "The chickens hit their heads". It isn't likely that the chickens hit a human head.

He hit the man on the back of the head with a brick.
[Scored IIA3 -- Others (human) killing, fighting, injuring other individuals or threatening to do so]

Although the complete sentence is parsed, in the present scoring algorithm only the subject, the verb and the object--in this case just the first four words--are used to achieve a score.

The dock has been blown up.
[Scored IIA1 -- Wildlife, flora, inanimate objects injured, broken, robbed, destroyed or threatened with such (with or without mention of agent)]

The parse puts this sentence into active voice filling in a dummy subject. The scoretree used (V1) has an arc for looking for subjects of this type. The arc looks for a "nil" or unfeatured subject.

He almost knocked over two gas pumps.
[Scored IIA1]

The verb is parsed as "to knock over". For the present it is a V1 verb (causing death or physical injury), but if the direct object had been "a bank" it would be a V2 verb (robbing). See the discussion of this problem which follows.

He picked on my mother a lot.
[Scored IIC2 -- Others (human) criticizing or depriving other individuals in a vague or mild manner]

I don't trust him anymore.
[Scored IC2 -- Self criticizing or depreciating others in a vague or mild manner]

The negative "not" makes an unscorable action verb "to trust" into a scorable action.

I burned a hole in the sofa.
[Scored IA1 -- Self killing, injuring, destroying, robbing wildlife, flora, inanimate objects or threatening to do so]

"Hole" is featureless. The scorer looks to the prepositional phrase modifying it for features.

Problems Still to be Handled

It is a property of many verbs that they take on different senses depending on the semantic features of the noun phrase acting as their direct objects. A simple example using the verb "to fire":

1. He fired a gun.
2. He fired a man.
3. He fired a piece of pottery.

In some cases, as with the first two sentences, the change in meaning necessarily dictates a change in the verb's semantic features or verb-type. In sentence 1 the verb has the verb-type V1 (causing death or physical injury); in sentence 2 it has the verb-type V5 (depriving; disappointing).

Another example of a verb showing this property is "to cut":

He cut her hair.
He cut prices.
He cut a class.
He cut a deck of cards.
He cut the engine.
He cut a song.

It is clear that some way must be found to handle this problem. One way would be to extend the concept of a scoretree by creating a special scoretree for each verb of this type. This would be useful for verbs like "to fire"

that have multiple verb-types. The scoretree would allow for disambiguation of the verb's meaning (and therefore verb-type) by testing for various kinds of direct objects by looking either for specific words ("pottery") or for specific features (+WEAPON +PROJECTILE).

Another method which would work well with verbs having multiple meanings but only one scorable meaning would require that a list of words or features leading to unscorable senses of the verb be created for each of these problematic verbs. The elements of the list would be checked against the direct object of the sentence in question and if any of these words or features matched, then no scoretree would be accessed and the clause would be scored "nil". Because a special scoretree would not be created with this method, it is a much thriftier approach.

Another problem is that of retrieving recipients that are not direct objects. We will illustrate this with the verb "to shoot".

Let us look at the sentences:

1. I shot the man.
2. I shot the rifle.
3. I shot at the target.

The first sentence is scored without difficulty. The second sentence is trickier for it does not mean that a hole was

put through the rifle. The real recipient of the action is not the direct object as it is in the first sentence but some unspecified thing. In the third sentence, the object of the prepositional phrase is the recipient. A scoretree for the verb "to shoot" must look for recipients in several places and in a specific order. It must first see if there is a direct object. If there is one, then it must check to see if it is a weapon or a projectile from a weapon. If so, or if there is no direct object, then it must look for a prepositional phrase beginning with "at". The real recipient is probably the noun phrase within the "at" prepositional phrase. (Of course, we are always going to come up against such sentences as "He shot the gun at the park"--an unfortunate ambiguity). More work must be done on finding ways to characterize the placing of recipients since their discovery is critical to our scoring scheme.

The third problem in addition to direct objects changing the meaning of verbs and the unstable placement of recipients in a sentence is that of single words implying a scorable action. For example:

The Sudan raids caused lots of trouble.
(IIA3 -- Others killing others or threatening to do so)
The picture of the killer was in the paper.
(IIA3)
I found a lot of racial prejudice among them.
(IIC2 -- Others criticizing others in a vague or mild manner)
It was just one damned thing after another.
(IC1 -- Self cursing)

It is not clear that there are any systematic and rapid ways of scoring these words. No techniques have been developed yet. A brute force look-up procedure may be necessary which is very time-consuming.

Inferencing

Perhaps the most fascinating aspect of this research is the discovery of the amount of inferencing that a human scorer does -- something that present day computers are ill-equipped to handle. One interesting example of this phenomena is the sentence, "He knocked over the goldfish bowl." In this sentence the scorer implies a sequence of action not at all explicit that goes something like this: the goldfish bowl is knocked over dumping both water and fish on the floor; water spreads everywhere (a characteristic of liquids) causing harm to the fish who cannot live without being immersed in it. The scorer codes this clause as others killing or injuring pets (IIA2).

Another example: "When the lambs are big enough, they end up in the freezer." Using his knowledge of the way the world behaves and implying action not explicit in the sentence as he did above, the human scorer realizes the implication of lambs ending up in the freezer and scores it IIA2 (Others killing pets).

A much subtler sentence requiring knowledge of the views of college kids and the significance of troop trains is, "Some college kids tried to stop a troop train." The scorer scores this one IIC3 (others adversely criticizing, expressing dislike of others). He does not score it this way because of the action alone--trying to stop a train is not equal to expressing dislike of others. Instead he uses his knowledge of the anti-war, anti-military attitudes of "college kids" and the military aspect of "a troop train" and put these things together into a scenario of political demonstration which one implies as meaing "others adversely criticizing, expressing dislike of others".

Although the present day field of content analysis is unable to handle these problems requiring the storing of huge amounts of knowledge about the world, we can look to the rising field of artificial intelligence for ideas and possibly future solutions.

BIBLIOGRAPHY

Gottschalk, L.A. and G.C. Gleser, The Measurement of Psychological States Through the Content Analysis of Verbal Behavior, University of Calif. Press, Berkeley, Calif., 1969

Gottschalk, L.A., C. Winget, and G. Gleser, Manual of Instruction for Using the Gottschalk-Gleser Content Analysis Scales: Anxiety, Hostility, and Social Alienation-Personal Disorganization, University of Calif. Press, Berkeley, Calif. 1969

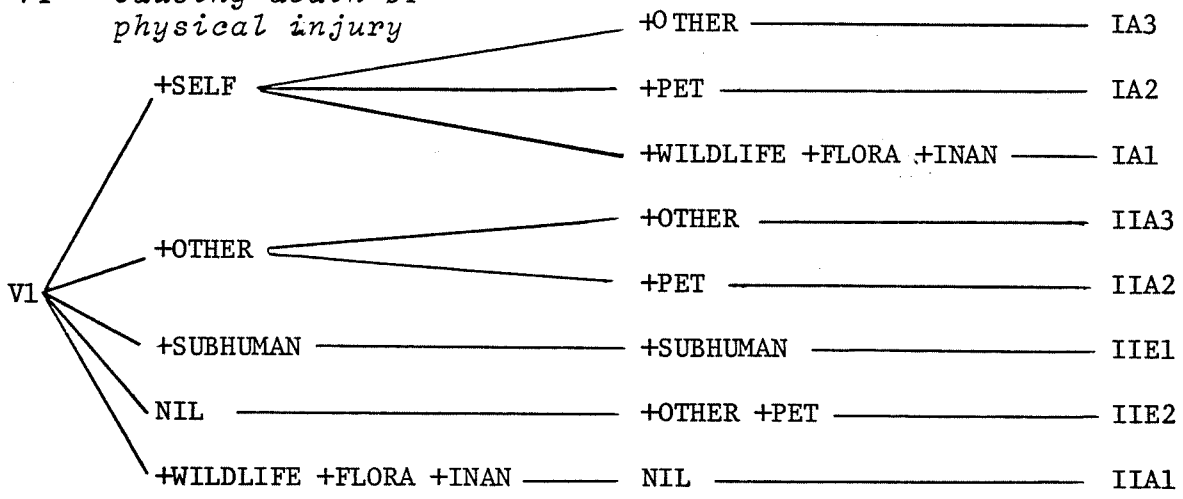
Woods, W.A., "An Experimental Parsing System for Transition Network Grammars," BBN Report 2362, Bolt Beranek and Newman, Inc., Cambridge, Mass. May 1972

Woods, W.A., R.M. Kaplan and B. Nash-Webber, "The Lunar Sciences Natural Language Information System: Final Report," BBN Report 2378, Bolt Beranek and Newman, Inc. Cambridge, Mass. June 1972

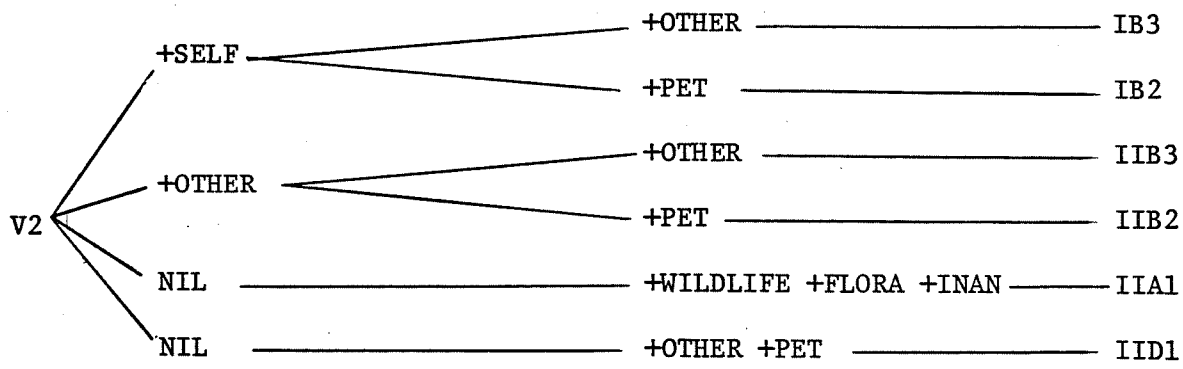
APPENDIX 1

Scoretrees for verb-types V1-V8

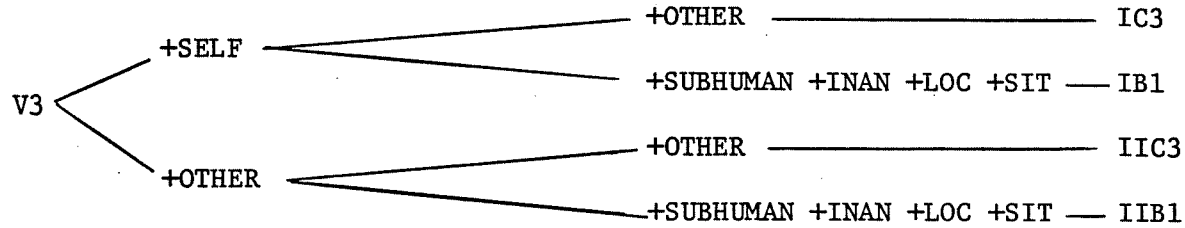
V1 - Causing death or physical injury



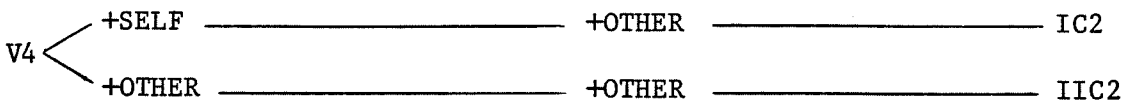
V2 - Causing suffering or anguish; robbing; abandoning



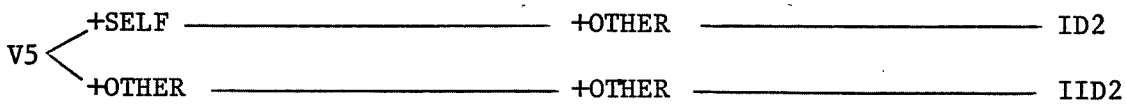
V3 - Expressing adverse criticism, anger, blame, dislike, depreciation



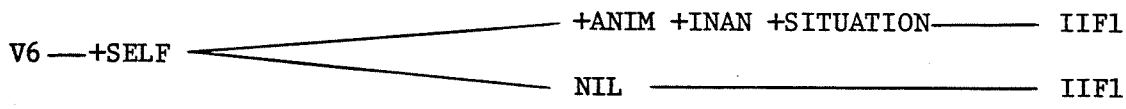
V4 - Expressing vague or mild criticism or depreciation



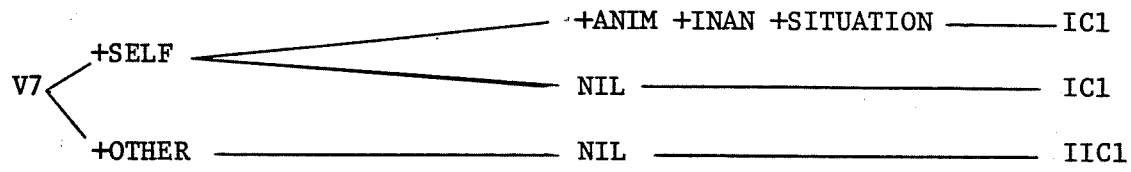
V5 - Depriving; disappointing



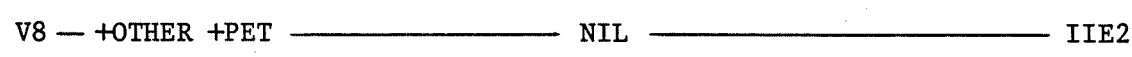
V6 - Denial of anger, dislike, hatred, cruelty and intent to harm



V7 - Cursing



V8 - Bying or killed violently



APPENDIX 2

Modifications Made to Woods' Parser

Woods' LSNLIS Parser was converted from BBN LISP to UCI LISP. The major problem encountered in this conversion was that of reducing the parser to a size small enough to run on a machine with much less core than originally envisioned. This reduction was accomplished by removing the following parts. SYSCONJ and all of the information saved for use by SYSCONJ, including the functions ALT, CONJGEN, CONJOIN, CONJSCOPE, CONJ STARTS, LONGBLOCK, POPCONJ, TRAIL, TRAIL1 and TRAILS were removed, as well as those parts of PARSER and STEP. Removing these items eliminates the ability to parse most occurances of conjunctions. (It is still possible to allow conjoined proper nouns by means of a grammar arc.) BACKUP which uses TRAIL information was removed, as were ASSIST and FORCER, which were deemed unnecessary for the sort of grammar we anticipated developing. SPOP and the Well Formed Sub String Facility (WFSS) were removed, along with a number of the utility routines used in the syntactic portion of Woods' system.

The Random Access Disk I/O routines necessary for maintaining a dictionary on DISK were written and added to UCI LISP, but were not used in the smaller versions of the parser. The system resulting from all of these reductions

requires some 5ØK of core when run interpretively (25K of which is the basic LISP system).

The files necessary to run the parser are: LOADIT.LSP, INIT, UTL.LSP, PAR.LSP, ARC.LSP, DIC.LSP, MOR.LSP, LEX.LSP, GMF.LSP, VARS.LSP, GMR.LSP and CHANGE.LSP. LOADIT.LSP contains an initialization function which loads the remaining files. INIT and UTL.LSP contain utility routines. PAR.LSP is the body of the parser. ARC.LSP and GMF.LSP contain the functions executed as pre-actions. DIC.LSP, MOR.LSP and LEX.LSP contain the tables and functions used in morphological stripping. VARS.LSP contains the global variables used by the parser, as well as some user interface functions. GMR.LSP is the grammar, and CHANGE.LSP consists of additions made in order to define new words.

In addition to all of the parser functions, before the parser is used a dictionary must be included. The present dictionary resides on GOTDEFS and provides examples of definitions of words.

States Deleted From the Grammar

The following states were deleted from our version of the Woods' grammar contained on the file GMR. With these deletions it is impossible to parse sentences requiring these forms of punctuation.

NP / NP /,ESP NP ,NP	multiple NP's separated by commas
NP //	handles ratios like "oxygen / rubidium"
NP NP; NPU ; NPU ;NP	multiple NP's separated by semicolons
PAREN PAREN PAREN	parenthesized expressions
S ; S ;S	multiple sentences separated by semicolons

Documentation of Changes Made
in the UCI Version of the Woods' Parser

```
(DEFPROP CHLIST
 (CHLIST COMPL/
         COMPL/ADVTYPE
         COMPL/S-ADV
         NP/
         NP/DET
         NP/HEAD
         NPR/
         R/WH
         S/
         VP/HEAD
         VP/NP
         VP/V
         VP/VP
         COMPFORMADV
         MODAL
         MORPHTABLE
         PARTNAMEP
         PLIST
         SUPFORMADV)
VALUE)
```

***Comments made following a function refer to the preceding function.

```
(DEFPROP NP/
 (NP/ (CAT DET
           T
           (COND ((GETF POSSPRO)
                  (ADDL ADJS (BUILDQ (POSS (NP (PRO *)))))
                  (SETRQ DET THE))
                 (T (SETR DET *)))
           (TO NP/ART))
      (CAT PRO
           T
           (SETR N (BUILDQ (PRO *)))
           (SETR NU (GETF NUMBER))
           (TO NP/HEAD))
      (CAT NEG (NULLR NEG) (SETR NEG *) (TO NP/))
      (CAT CONJ T (SETR NTYPE *) (TO COMPL/NTYPE))
      (JUMP NP/ART
            (OR (WRD MOST)
                (NOR (CAT (DET PRO NEG))
                     (WRD (WHOSE WHO WHAT WHETHER IF))))))
VALUE)
```

**Arc 3 deleted. Arc (CAT CONJ T ...) added to allow for any conjunction, not just "whether" or "if".

```
(DEFPROP NP/DET
 (NP/DET
  (MEM (MORE MOST)
       (AND (GET (NEXTWRD) (QUOTE ADJ)) (NOT (WRD THE DET)))
       (SETR MORE-MOST *)
       (TO NP/MORE))
  (CAT N
       (WRD (AVERAGE MAXIMUM MINIMUM MOST LEAST))
       (SETR N (BUILDQ (N *)))
       (SETR NU (GETF NUMBER))
       (TO NP/AVG))
```

**Arc 3 deleted. Specialized arc for Lunar Project.

```
(DEFPROP COMPL/
 (COMPL/
  (WRD FOR T (TO FOR/FOR))
  (WRD THAT T (SETRQ NTYPE THAT) (TO COMPL/NTYPE))
  (WRD THAN T (SETRQ NTYPE THAN) (TO COMPL/NTYPE))
  (JUMP FOR/NP
        T
        (COND
         ((NULLR SUBJ)
          (SETR SUBJ (BUILDQ (NP (PRO SOMETHING)))))))
  (CAT CONJ T (SETR ADVTYPE *) (TO COMPL/ADVTYPE)))
VALUE)
```

**Arc 5 added to handle compliments beginning with a conjunction other than "for, that, than" or "to". (All references to conjunctions are to subordinate not coordinate -- coordination is not handled at present.) e.g. He left before I could catch him.

```
(DEFPROP COMPL/ADVTYPE
 (COMPL/ADVTYPE (PUSH S/ T (SETR S *) (TO COMPL/S-ADV)))
VALUE)
```

**State COMPL/ADVTYPE added as a result of adding Arc 5 to COMPL/.

```
(DEFPROP COMPL/S-ADV
 (COMPL/S-ADV (POP (BUILDQ (ADV + +) ADVTYPE S) T))
VALUE)
```

**State COMPL/S-ADV added as a result of adding Arc 5 to COMPL/.

```
  (CAT ADJ
       T
       (ADDL ADJS (BUILDQ (≥ (ADJ) # (*)) FEATURES))
       (TO NP/DET))
  (CAT N
       T
       (SETR N (BUILDQ (N *)))
       (SETR NU (GETF NUMBER))
       (TO NP/N))
  (CAT ADV T (SETR ADVS (BUILDQ ((ADVS *)))) (TO NP/ADV))
  (CAT V
       (OR (GETF PASTPART) (GETF PRESPART))
       (ADDL ADJS (BUILDQ (ADJ (PARTICIPLE #)) LEX))
       (TO NP/DET))
  (CAT V
       (GETF PRESPART)
       (SETR N (BUILDQ (N #) LEX))
       (SETRQ NU SG)
       (TO NP/N))
  (PUSH S/AUX
        (OR (CAT (NEG ADV)) (CHECKF V PRESPART))
        (SENDR SUBJ
               (COND ((WRD POSS (CAAR# (GETR ADJS)))
                      (SETQ TEMP (CADAR# (GETR ADJS)))
                      (SETR ADJS (CDR# (GETR ADJS)))
                      TEMP)
                     (T (SENDR SUBFLAG T)
                        (BUILDQ (NP (PRO SOMETHING)))))))
        (SENDRQ TYPE POSS-ING)
        (SETRQ NTYPE NOM)
        (SETR S *)
        (TO COMPL/S))
  (PUSH NPR/ T (SETR N *) (SETRQ NU SG) (TO NP/N)))
VALUE)
```

```
(DEFPROP NP/HEAD
 (NP/HEAD
  (VIR PP
       (NPREP)
       (ADDL NMODS *)
       (COND ((NULLR PPFLAG) (SETRQ PPFLAG T)))
       (TO NP/HEAD))
  (PUSH
   R/
   (AND (OR (WRD (WHO WHOM WHOSE WHICH THAT))
            (AND (WRD (WHICH WHOM WHOSE) (NEXTWRD))
                 (CAT PREP)))
        (OR (CADR# (GETR DET)) (WRD PL (GETR NU))))
   (SENDRQ TYPE REL)
   (SENDR WH (BUILDQ (NP (DET WHR) + (NU +)) N NU))
   (SENDR ANAPHORFLG (CADR# (GETR N)))
   (ADDL NMODS *)
   (TO NP/R))
  (PUSH
   R/WH
   (AND (CAT V) (GETR RELVPFLG))
   (!
    (COND
     ((WRD (WHICHQ HOWMANY) (CADR# (GETR DET)))
      (SENDR
       ANAPHORFLG
       (COND ((WRD ONES (CADR# (GETR N))) (GETR ANAPHORFLG))
             (T (CADR# (GETR N))))))))
   (SENDRQ TYPE QREL)
   (SENDRQ RELVPFLG T)
   (SENDR
    WH
    (BUILDQ
     (NP (DET WHR) # (NU +))
     (COND ((WRD ONES (CADR# (GETR N))) (GETR ANAPHORFLG))
           (T (GETR N)))
     NU))
   (ADDL NMODS *)
   (TO NP/NP))
  (JUMP
   NP/NP
   (OR (NOT *)
       (AND (WRD TO) (RFEAT INDOBJ V))
       (VPREP *)
       (NOR (NPREP *)
            (WRD (OF FOR))
            (AND (CAT PREP) (NOT (GETR N 2))))))
```

**(WRD TO) added. A stop-gap measure. Added so that the preferred parse favors "to" as beginning a To-Compliment after a noun phrase.

```
      (WRD TO))
 (LIFTR NPFEATURES (RESUMETAG NP/HEAD))
 (COND ((GETR PARTFLAG) (LIFTR ANAPHORFLG (GETR N) 2))))
(PUSH PP/
      (CAT PREP)
      (SENDR V (GETR V))
      (ADDL NMODS *)
      (COND ((NULLR PPFLAG) (SETRQ PPFLAG T)))
      (TO NP/HEAD))
(PUSH FOR/NP
      (WRD TO)
      (ADDL NMODS (BUILDQ (COMPL *)))
      (COND ((NULLR PPFLAG) (SETRQ PPFLAG T)))
      (TO NP/HEAD))
(TST R/NIL
     (AND *
          (GETR PPFLAG)
          (NOR (WRD (WHAT WHO WHOM WHICH THAT WHOSE))
               (GETR QDET)
               (AND (WRD BE (GETROOT * V))
                    (NOT (EQ * (QUOTE BEING))))))
     (SUSPEND 1)
     (SENDRQ TYPE REL)
     (SENDR WH (BUILDQ (NP (DET WHR) + (NU +)) N NU))
     (PUSH R/NIL)
     (ADDL NMODS *)
     (TO NP/R))
(JUMP
 NP/NP
 (NOR (NOT *)
      (AND (WRD TO) (RFEAT INDOBJ V))
      (VPREP *)
      (NOR (NPREP *) (WRD (OF FOR))))
 (LIFTR NPFEATURES (RESUMETAG NP/HEAD))
 (COND ((GETR PARTFLAG) (LIFTR ANAPHORFLG (GETR N) 2))))
(PUSH
 R/NIL
 (NOR (GETR PPFLAG)
      (WRD (WHAT WHO WHOM WHICH THAT WHOSE))
      (GETR QDET)
      (AND (WRD BE (GETROOT * V))
           (NOT (EQ * (QUOTE BEING))))
      (NULL STRING))
 (SENDRQ TYPE REL)
 (SENDR WH (BUILDQ (NP (DET WHR) + (NU +)) N NU))
 (ADDL NMODS *)
 (TO NP/R))
(PUSH
```

```
   COMPL/
   (AND (WRD THAT)
        (WRD (A THE) (CADR# (GETR DET)))
        (RFEAT FACTN HEAD))
   (ADDL NMODS (BUILDQ (COMPL *)))
   (TO NP/NP))
  (PUSH COMPL/NTYPE
        (RFEAT FACTN HEAD)
        (SENDRQ NTYPE THAT)
        (ADDL NMODS (BUILDQ (COMPL *)))
        (TO NP/NP))
  (JUMP NP/NP T))
VALUE)
```

```
(DEFPROP NPR/
 (NPR/ (CAT NPR T (SETR NPR (LIST *)) (TO NPR/NPR))
       (TST NPR/NAME
            (AND (NULL (CAT NPR)) (PARTNAMEP *))
            (SETR NPR *)
            (SETR NPR (BUILDQ (+ (NUMBER SG)) NPR))
            (TO NPR/NPR)))
VALUE)
```

**Arc 1 deleted. Specialized to Lunar Project.

**Arc (TST NPR/NAME ...) added to recognize capacitors (C_n), resistors (R_n), diodes (D_n), and transistors (Q_n), where n is a number.

```
(DEFPROP R/WH
 (R/WH (CAT ADV T (ADDL VMODS (BUILDQ (ADV *))) (TO R/WH))
       (PUSH PP/ (CAT PREP) (ADDL VMODS *) (TO R/WH))
       (PUSH NP/
             (NOR (CAT V) (GETR RELVPFLG))
             (COND ((GETR WH) (HOLD (GETR WH) NIL)))
             (SETR SUBJ *)
             (TO S/NP))
       (WRD THERE
            T
            (SETR THERE T)
            (SETR SUBJ (GETR WH))
            (TO S/NP))
       (JUMP S/NP
             (AND (GETR WH) (CAT V))
             (SETR SUBJ (GETR WH))))
VALUE)
```

**(SETR SUBJ (GETR WH)) added to last arc. Apparently just an error.

```
(DEFPROP S/
 (S/ (JUMP S/Q (QSTART) (SETRQ TYPE Q))
     (WRD PLEASE
          (NULL STACK)
          (ADDL VMODS (BUILDQ (ADV PLEASE)))
          (TO S/IMP))
     (JUMP S/IMP (AND (CHECKF V UNTENSED) (NULL STACK)))
     (JUMP S/DCL
           (NOR (CAT PREP) (QSTART) (NULL STRING))
           (SETRQ TYPE DCL))
     (CAT ADV
          (RFEAT NEGADV)
          (ADDL VMODS (BUILDQ (ADV *)))
          (SETRQ TYPE DCL)
          (TO S/NP))
     (CAT ADV T (ADDL VMODS (BUILDQ (ADV *))) (TO S/))
```

```
     (PUSH PP/
           (WRD OF)
           (HOLD * (QUOTE ((PARTITIVE))))
           (TO S/))
     (PUSH PP/
           (CAT PREP)
           (HOLD * (QUOTE ((FRONTED))))
           (TO S/QP1))
     (PUSH PP/
           (CAT PREP)
           (ADDL VMODS (BUILDQ (ADV *)))
           (TO S/))
     (PUSH NP/
           T
                 (ADDL (VMODS (BUILDQ ADV *)))
                 (TO S/))
VALUE)
```

**Arc 9 added to handle sentences beginning with prepositional phrases.
e.g. In the morning we went out.

**Arc 10 added to handle noun phrases used as adverbs at the front of a sentence.
e.g. Last night we went to the movies.

```
(DEFPROP VP/HEAD
 (VP/HEAD
  (CAT PREP
       (SETQ ATEMP (VPARTICLE V))
       (SETR V ATEMP)
       (SETR HEAD ATEMP)
       (TO VP/V))
  (PUSH NP/
        (AND (WRD BE V) STRING)
        (SENDRQ V BE)
        (SENDR ANAPHORFLG (GETR ANAPHORFLG))
        (SETR OBJ *)
        (TO VP/NP))
  (PUSH NP/
        (AND (VTRANS V) (NOT (WRD BE V)))
        (SENDR V (GETR V))
        (SENDR ANAPHORFLG (GETR ANAPHORFLG))
        (SETR OBJ *)
        (TO VP/NP))
  (PUSH S/Q
        (AND (VTRANS V) (WRD (WHICH WHO WHAT WHOSE)))
        (SENDRQ TYPE IDQ)
        (SETR OBJ (BUILDQ (NP *)))
        (TO VP/NP))
  (VIR NP (VTRANS V) (SETR OBJ *) (TO VP/NP))        **Deleted (RESUME V)
  (JUMP VP/NP
        (OR (RFEAT INTRANS V)
            (AND (VTRANS V) (GETR OBJ))
            (ADJVERB)))
  (WRD MORE
       (AND (RFEAT COPULA V) (GET (NEXTWRD) (QUOTE ADJ)))
       (TO VP/MORE))
  (PUSH COMPL/
        (AND (WRD THAT) (RFEAT THATCOMP V))
        (SETR COMPL *)
        (TO VP/NP))
```

**Substituted "ATEMP" for "TEMP" in Arc 1. Conflicted with the atom "TEMP" in PARSER. Apparently an error.

**Switched Arcs 3 and 6. Because of the addition of Arc 7 in VP/VP for noun phrases acting as adverbs, we wanted to make sure that if a verb could be both TRANS and INTRANS that VP/HEAD woudl try for TRANS first. Thus Arc 7 in VP/VP would be taken only as a last resort --

a. If the verb is INTRANS and a noun phrase follows the verb.
b. If the verb is TRANS (or TRANS and INTRANS) and two noun phrases follow the verb.

```
  (PUSH COMPL/
        (AND (WRD (FOR TO))
             (OR (RFEAT FORCOMP V) (RFEAT TOCOMP V)))
        (! (COND ((WRD TO) (SENDR SUBJ (GETR SUBJ)))))
        (SETR COMPL *)
        (TO VP/VP))
  (PUSH COMPL/
        (AND (WRD (FOR TO))
             (NOR (RFEAT TOCOMP) (RFEAT FORCOMP)))
        (! (COND ((WRD TO) (SENDR SUBJ (GETR SUBJ)))))
        (ADDL VMODS (LIST (QUOTE ADVMANNER) *))
        (TO VP/VP))
  (PUSH COMPL/NTYPE
        (SCOMP V)
        (SENDRQ NTYPE THAT)
        (SETR COMPL *)
        (TO VP/NP)))
VALUE)
```

**Added a check for a TOCOMP verb in Arc 9. Possibly just an oversight in the old version of GMR.

**Arc 10 added to handle sentences like:
He laughed <u>to relieve tension</u>.

```
(DEFPROP VP/NP
 (VP/NP (PUSH COMPL/
              (OR (AND (WRD (FOR THAT))
                       (EQUAL (GETR SUBJ)
                              (QUOTE
                               (NP (PRO IT) (NU SG)))))
                  (AND (WRD THAT) (GETR AGFLAG)))
              (COND ((GETR AGFLAG) (SETRQ AGFLAG NIL)))
              (SETR SUBJ *)
              (TO VP/VP))
        (PUSH FOR/NP
              (AND (RFEAT FORCOMP V) (WRD TO) (GETR OBJ))
              (SENDR SUBJ
                     (COND ((RFEAT SUBJLOW V) (GETR SUBJ))
                           (T (GETR OBJ))))
              (COND
               ((NOT (RFEAT TRANSCOMP V)) (SETR OBJ NIL)))
              (SETR COMPL *)
              (TO VP/VP))
        (PUSH FOR/NP
              (AND (WRD TO)
                   (NOT (RFEAT FORCOMP))
                   (GETR OBJ))
              (SENDR SUBJ
                     (COND ((RFEAT SUBJLOW V) (GETR SUBJ))
                           (T (GETR OBJ))))
              (COND
               ((NOT (RFEAT TRANSCOMP V)) (SETR OBJ NIL)))
              (ADDL VMODS (LIST (QUOTE ADVMANNER) *))
              (TO VP/VP))
        (CAT PREP
             (SETQ ATEMP (VPARTICLE V))
             (SETR V ATEMP)
             (SETR HEAD ATEMP)
             (TO VP/V))
```

**Arc 3 added to handle To'Compliments used as adverbs.
e.g. She kept the letter to show him later.

**"ATEMP" substituted for "TEMP" in Arc 5. Conflicted with the atom "TEMP" in PARSER.

```
        (CAT ADV T (ADDL VMODS (BUILDQ (ADV *))) (TO VP/NP))
        (PUSH NP/
              (AND (RFEAT INDOBJ V) (GETR OBJ))
              (ADDL VMODS (BUILDQ (PP (NIL TO) +) OBJ))
              (SETR OBJ *)
              (TO VP/VP))
        (PUSH COMPL/
              (AND (WRD (FOR THAT)) (RFEAT INDOBJ V))
              (ADDL VMODS (BUILDQ (PP (NIL TO) +) OBJ))
              (SETR OBJ NIL)
              (SETR COMPL *)
              (TO VP/VP))
        (JUMP VP/VP T))
VALUE)
```

```
(DEFPROP VP/V
 (VP/V (CAT V
            T
            (COND ((GETF PASTPART)
                   (COND ((AND (RFEAT COPULA V)
                               (VPASSIVE *))
                          (HOLD (GETR SUBJ)
                                (GETR NPFEATURES))
                          (SETR SUBJ
                                (BUILDQ
                                 (NP (PRO SOMETHING))))
                          (SETR AGFLAG T))
                         ((AND (NULLR ASPECT) (WRD HAVE V))
                          (SETRQ ASPECT (PERFECT)))
                         (T (ABORT))))
                  ((GETF PRESPART)
                   (COND ((WRD BE V)
                          (ADDR ASPECT (QUOTE PROGRESSIVE)))
                         ((WRD POSS-ING TYPE))
                         (T (ABORT))))
                  ((OR (NOT (GETF UNTENSED)) (GETR V))
                   (ABORT)))
            (SETR V *)
            (TO VP/V))
```

**Substituted (RFEAT COPULA V) for (WRD BE V) in Arc 1 to allow for any copula verb.

```
      (CAT ADJ
           (RFEAT COPULA V)
           (SETR V
                 (BUILDQ
                  (≥ (ADJ) (*) # #)
                  (COND ((GETF COMPARATIVE)
                         (QUOTE (COMPARATIVE)))
                        ((GETF SUPERLATIVE)
                         (QUOTE (SUPERLATIVE))))
                  (COND
                   ((WRD (APPEAR SEEM) V)
                    (QUOTE (SEEMING))))))
           (TO VP/ADJ))
      (MEM (MORE MOST)
           (AND (RFEAT COPULA V)
```

```
                (GET (NEXTWRD) (QUOTE ADJ)))
           (SETR MORE-MOST *)
           (TO VP/COMP-ADJ))
      (PUSH NP/
            (AND (GETR THERE)
                 (NULLR SUBJ)
                 (WRD (BE EXIST) V))
            (COND
             ((NOT (PNCHECK * (GETR PNCODE))) (ABORT)))
            (SETR SUBJ *)
            (TO VP/V))
      (JUMP VP/HEAD
            (GETR SUBJ)
            (COND ((NULLR V)
                   (COND ((WRD DO MODAL) (SETRQ V DO)
                                         (SETR MODAL NIL))
                         (T (ABORT))))
                  (T
                   (COND
                    ((AND (GETR THERE) (WRD BE V))
                     (SETRQ V EXIST)))))
            (SETR HEAD (GETR V)))
      (CAT ADV T (ADDL VMODS (BUILDQ (ADV *))) (TO VP/V)))
VALUE)
```

```
(DEFPROP VP/VP
 (VP/VP (WRD BY (GETR AGFLAG) (TO VP/AGT))
        (WRD BY
             (AND (WRD POSS-ING TYPE)
                  (NULLR OBJ)
                  (NULLR SUBFLAG))
             (SETR OBJ (GETR SUBJ))
             (TO ING/BY))
        (MEM (OF BY)
             (GETR SUBFLAG)
             (SETR SUBFLAG NIL)
             (TO ING/BY))
        (CAT ADV T (ADDL VMODS (BUILDQ (ADV *))) (TO VP/VP))
        (PUSH PP/ (CAT PREP) (ADDL VMODS *) (TO VP/VP))
        (PUSH COMPL/
              (CAT CONJ)
              (ADDL VMODS (BUILDQ (COMPL *)))
              (TO VP/VP))
        (PUSH NP/
              *
                    (ADDL VMODS (BUILDQ (ADV *)))
                    (TO VP/VP))
        (JUMP S/VP T))
VALUE)
```

**Arc 6 added to allow adverb compliments that are sentences:
I climbed the mountain <u>because it was there</u>.

**Arc 7 added to allow adverb compliments that are noun phrases:
e.g. He hit a homer <u>last week</u>.

```
(DEFPROP COMPFORMADV
 (LAMBDA (ENDING) (EQ (CAR ENDING) (GET * (QUOTE ADV))))
FEXPR)
```

**Function added to test if the dictionary entry for a particular adverb includes a comparative ending.

```
(DEFPROP MODAL
 (LAMBDA NIL
  (MEMB * (QUOTE (MIGHT DO WILL-MODAL SHALL CAN MAY MUST))))
EXPR)
```

**"MUST" was added as a modal verb.

```
(DEFPROP MORPHTABLE
 (MORPHTABLE (N ((S) NIL N (PLURAL -S) (NUMBER PL))
                ((E S) NIL N (PLURAL -ES) (NUMBER PL))
                ((I E S) (Y) N (PLURAL -ES) (NUMBER PL))
                ((E N) NIL N (PLURAL -EN) (NUMBER PL)))
             (V ((E S) NIL
                       V
                       (CTYPE ES-ED)
                       (TNS PRESENT)
                       (PNCODE /3SG))
                ((I E S) (Y)
                         V
                         (CTYPE ES-ED)
                         (TNS PRESENT)
                         (PNCODE /3SG))
                ((S) NIL
                     V
                     (OR (CTYPE S-ED) (CTYPE S-D))
                     (TNS PRESENT)
                     (PNCODE /3SG))
                ((E D) NIL
                       V
                       (OR (CTYPE S-ED) (CTYPE ES-ED))
                       (TNS PAST)
                       (PASTPART))
                ((I E D) (Y)
                         V
                         (CTYPE ES-ED)
                         (TNS PAST)
                         (PASTPART))
                ((E D) (E)
                       V
                       (CTYPE S-D)
```

```
            (TNS PAST)
            (PASTPART))
      ((I N G) NIL
               V
               (OR (CTYPE S-ED) (CTYPE ES-ED))
               (PRESPART))
      ((I N G) (E) V (CTYPE S-D) (PRESPART)))
(ADJ ((E R) NIL
            ADJ
            (COMPFORM ER-EST)
            (COMPARATIVE))
     ((E R) (E)
            ADJ
            (COMPFORM R-ST)
            (COMPARATIVE))
     ((I E R) (Y)
              ADJ
              (COMPFORM ER-EST)
              (COMPARATIVE))
     ((E S T) NIL
              ADJ
              (SUPFORM ER-EST)
              (SUPERLATIVE))
     ((E S T) (E)
              ADJ
              (SUPFORM R-ST)
              (SUPERLATIVE))
     ((I E S T) (Y)
                ADJ
                (SUPFORM ER-EST)
                (SUPERLATIVE)))
(ADV ((E R) NIL
            ADV
            (COMPFORMADV ER-EST)
            (COMPARATIVE))
     ((E R) (E)
            ADV
            (COMPFORMADV R-ST)
            (COMPARATIVE))
     ((I E R) (Y)
              ADV
              (COMPFORMADV ER-EST)
              (COMPARATIVE))
     ((E S T) NIL
              ADV
              (SUPFORMADV ER-EST)
              (SUPERLATIVE))
     ((E S T) (E)
```

```
                                ADV
                                (SUPFORMADV R-ST)
                                (SUPERLATIVE))
                        ((I E S T) (Y)
                                   ADV
                                   (SUPFORMADV ER-EST)
                                   (SUPERLATIVE))))
VALUE)
```

**(ADV ...) section was added to recognize comparative and superlative endings on adverbs.

```
(DEFPROP PARTNAMEP
 (LAMBDA(NM)
  (AND (MEMQ (CAR (SETQ NM (EXPLODE NM))) (QUOTE (C R D Q)))
       (NUMBERP (READLIST (CDR NM)))))
EXPR)
```

**Function which outputs "T" if the input is a cathode (C_n), resistor (R_n), diode (D_n) or a transistor (Q_n).

```
(DEFPROP PLIST
 (LAMBDA(X)
  (PROG (Y Z)
        (SETQ Z (SETQ Y (CONS NIL NIL)))
        (SETQ X (CDR X))
   L1   (COND ((NULL X) (RETURN (CDR Y)))
              ((NOT
                (MEMB (CAR X)
                      (QUOTE
                       (PNAME VALUE
                              MACRO
```

```
                                    EXPR
                                    FEXPR
                                    SUBR
                                    FSUBR
                                    LSUBR))))
                    (NCONC Z (CONS (CAR X) (CONS (CADR X) NIL)))
                    (SETQ Z (CDDR Z))))
            (SETQ X (CDDR X))
            (GO L1)))
EXPR)
```

**"VALUE" added to the list (PNAME MACRO EXPR FEXPR SUBR FSUBR LSUBR). Apparently an error.

```
(DEFPROP SUPFORMADV
 (LAMBDA (ENDING) (EQ (CAR ENDING) (GET * (QUOTE ADV))))
FEXPR)
```

**Function added to test if the dictionary entry for a particular adverb includes a superlative ending.

APPENDIX 3

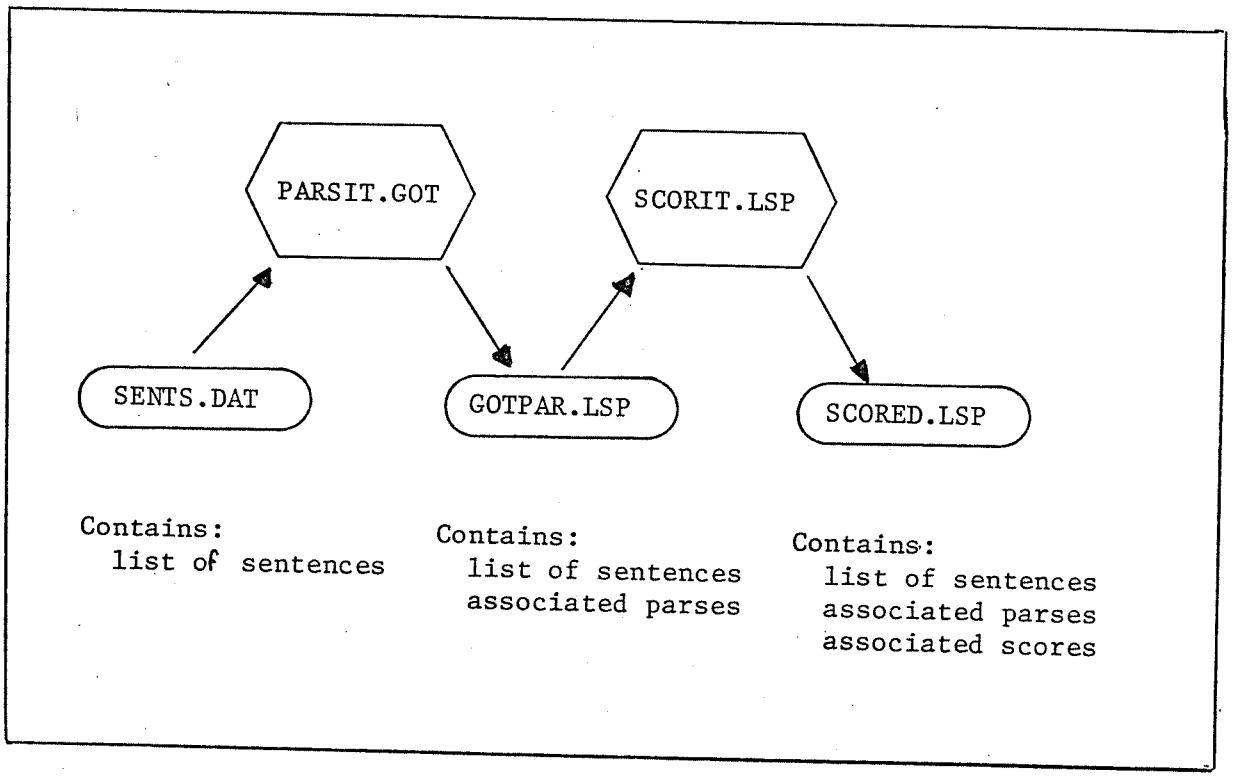
Documentation Of Functions Used in Scoring

Documentation for PARSIT.GOT
SCORIT.LSP
SENTS.DAT
GOTPAR.LSP
SCORED.LSP

Because of our core limitations on the UCI PDP-1Ø, we are forced to divide the scoring activity into two parts:

1. We first parse a list of sentences contained in the file SENTS.DAT, writing each parse out onto another file, GOTPAR.LSP.
2. We then read in these parsed sentences one at a time, score it, then write it out onto another file, SCORED.LSP.

The chain of movement is as follows:



Documentation for GOTFNS.LSP

This file contains the functions used in scoring.

Vocabulary

VERB TYPE: A property of verbs (e.g. V1, V2, V3)

SCORETREE: A property of verb types (e.g. The scoretree for V8 is ((+OTHER +PET) NIL . IIE2))

FEATURES: A property of nouns and noun phrases (e.g. +SELF +FLORA +INAN)

SCORE: A letter-number combination assigned to a sentence according to the Gottschalk-Gleser Content Analysis Scale for Hostility Directed Outward.

Functions

GETFEATS: Function which retrieves the features for a particular noun.

GETVTYPE: Function which retrieves the verb type for a particular verb.

GETSCORETREE: Function which retrieves the scoretree for a particular verb type.

SCORE: Function which takes a sentence as its input and outputs the score tor that sentence.

GOWALK: Function which has three inputs -- the verb's scoretree, the features of the subject, and the features of the object. It outputs the score for the sentence. It calls WALK, WALK1, WALK2, and WALK3.

FEATMATCH: Function which outputs "T" if the features on a particular branch of the current scoretree match or intersect with the features of a particular noun phrase. Otherwise FEATMATCH outputs "NIL". FEATMATCH calls MEMFEAT and INTERSECTP.

SUBPARSE: Function which takes as its input the parsed sentence and searches it, setting SUBJ, OBJ, VERB and NEG to the appropriate sub-structures of the parse.

NPFEATS: Function whose input is a parsed noun phrase and whose utput is a list of features for that noun phrase.

MAPCOND: Function that does successive MAPC's until the function FN returns a non-nil value. The value of MAPCOND is that first non-nil value encountered for FN.

Documentation for PRPDCT.DAT

This file contains the dictionaries used in scoring. It also contains the functions for bringing them into core.

ADDALL: Function that adds the properties "SCOREFEATS", "VTYPE", And "SCORETREE" to the appropriate atoms.

VTYPEDCT: Atom whose value is a list of verbs with their associated verb types.

FEATSDCT: Atom whose value is a list of nouns with their associated features.

SCORETREEDCT: Atom whose value is a list of verb types with their associated scoretrees.

Documentation for ALPHA.LSP

This file contains the function ALPHA which takes an atom as its input (i.e. VTYPEDCT, SCORETREEDCT, FEATSDCT) and outputs in alphabetical order the list of elements that is the atom's value.

APPENDIX 4

Scoring an Interview

The following interview is from Manual of Instruction for Using the Gottschalk-Gleser Content Analysis Scales, p. 86. A human technician has coded the interview for Hostility Directed Outward (see Table 1 for a description of the scale). We will describe how the computerized scoring system would handle each clause keeping in mind how it might differ from a human scorer. We think that this description is vital to understanding both the limitations of the current parser and the limitations of our computerized content analysis system in general.

The Interview:

1. Well, this is going to be a story
2. about when I was out in the Children's Home.
3. Once we were doing something on bicycles
4. and the manager grabbed us and started to whip us. [SCORED IIA3]
5. And then I argued with him [SCORED IC2]
6. so he made us all three stay in bed all day. [SCORED IID2]
7. And then another time we was out in the park
8. and I was shooting arrows at another boy. [SCORED IA3]
9. I hit him in the cheek [SCORED IA3]
10. so I got whipped again.
11. After that I just ran away. [SCORED IB1]
12. Me and a buddy stole a car [SCORED IB3]
13. but I got mad at him [SCORED IC3]
14. and we split up.
15. But the cops got both of us anyway and brought us back [SCORED IIC3]
16. He got sent to reform school then. [SCORED IIC3]
17. I ran over a dog [SCORED IA2]

18. while I was driving
19. and I didn't feel so damn good about that. [SCORED IC1]
20. Then the front tire got a flat [SCORED IIA1]
21. and that's
22. when they caught me.
23. I hope
24. I don't get sent to any reform school,
25. cause I'm pretty sure
26. I wouldn't like it at all. [SCORED IB1]
27. Every day is just alike, the same ordinary things
28. unless you get into fights or something. [SCORED IA3]
29. They have a system of swats [SCORED IIA3]
30. if the boys do something wrong [SCORED IIC2]
31. or sometimes they take away their privileges. [SCORED IID2]

Comments on Scored Clauses:

4. This is not a complete sentence because of the "and" at the beginning. Since the parser only handles complete sentences, the "And" must be removed for it to parse. Also we cannot handle conjunctions in our trimmed-down version of the Woods' parser. Even with the parsing problems solved, we are still unable to score the sentence because it contains an embedded clause -- a to-complement, "to whip us". Our scoring algorithm does not now handle the extraction of features or scores from embedded clauses. Such a facility should not be too difficult to add. It should be able to handle:

a) Clauses as subjects:
Getting badges in the Boy Scouts is a farce.
+EVENT
SCORED IB1

b) Clauses as objects:
I shot what looked like a rabbit
+WILDLIFE
SCORED IA1

c) For-complements:
I wanted him to kill the spider
IIA1
SCORED IIA1

d) To-complements:
He wanted to smash the vase
IIA1
SCORED IIA1

5. "And" must be removed to parse. Scorable in current system.

6. "So" must be removed to parse. Also "us all three" is not an acceptable noun phrase. In addition, the parser does not accept verb complements of this form. Not scorable anyway. How does one decide tht being made to stay in bed is a punishment? We could agree to always score sentences where an individual is being made to do something "disagreeable". But how do we classify what is disagreeable? If the sentence had been "He made us stay on the sidewalk", I think most would agree that that is not such a disagreeable thing to be made to do. What is the difference between "staying in bed" and "staying on the sidewalk"? In fact, staying in bed is not necessarily such a disagreeable thing to be made to do (most people would enjoy it) unless (and this is the crux of the issue) it is intended to be a punishment. Discovering the intent of an action is far beyond the scope of this project.

8. "And" must be removed to parse. The problem with "shoot" has been discussed in the section of this paper titled "Problems Still to be Handled".

9. Scorable in present system.

11. Scorable if we are willing to say that "run away" always means "to express dislike of a place by leaving it". If we were to do this the sentence "Seeing that the bomb was about to explode, the boy ran away" would be scored IB1. This problem often comes up. For example, the verb "to take" sometimes means "to steal" as in the sentence, "The man took the watch from the display when the jeweler wasn't looking." Contrast this with "The jeweler took the watch from the display window". I think one would find it difficult to state in very precise terms what makes one action stealing and the other perfectly acceptable. Expand this to all senses of the verb "to take" and the problem becomes immmense:
He took a bath. (stole a bath?)
He took a peek. (stole a peek?)
He took a ride. (stole a ride?)

12. Parser does not handle conjunction. Otherwise the clause is scorable in the present system.

13. "But" must be removed to parse clause. Scorable if a special scoretree is developed for the verbs "mad", "angry", "sore" and the like. ["Get" is a copula verb here and the parser rewrites the predicate adjective "mad" as the verb.]

15. Not scorable unless a very specialized scoretree for "to get" can be developed (or for "to bring back"). It would have to differentiate between this sentence and ones like "My father got both of us and brought us home." Problem is somewhat similar to that of scoring sentence 11.

16. Scorable because of the negative aspect of where he got sent. That is, if it had read "He got sent to the candy store" it wouldn't have been scored. We need to tag undersireable places and objects with a feature like +NEGATIVE.
In addition, we see here a sentence that requires that we look further than just the subject, verb and object to score it; we need to look at the adverb phrase "to the reform school". We do not have any way of doing that now although the task seems to point to using a special scoretree of some sort.

17. Scorable in present system.
Verb is "ran-over".

19. "And" must be removed to parse clause. Need some facility for parsing swear words -- or any single scorable word. Problem described in "Problems Still to be Handled".

2Ø. A strange way of saying that the front tire became flat. Not scorable unless a very specialized scoretree is developed.

26. Scorable. No referencing capabilities exist except for that done in parsing. Thus "it" is assumed to be an inanimate object. In fact, in this sentence it is an EVENT -- being in reform school -- but the score for either case is IB1.

28. "Unless" must be removed to parse. Need a specialized scoretree for "to get" that includes a search for the preposition "into" plus words like:
"a spat" = IA2
"an argument" = IA2
"a fight" = IA3

29. "A system of swats" implies "someone swats someone". It isn't practicle to search all clauses looking for this specific phrase. (It isn't likely that the phrase would come up again anyway.) It would be an interesting project to develop a way of retrieving the verbal action and actors that have been lost in converting an action into a noun phrase.
System where people swat people = system of swats

3Ø. "If" must be removed to parse.
Ambiguous by itself. Does it mean:
a) The boys do something in a wrong manner.
b) The boys do something (and that something is wrong).

31. Scorable if specialized scoretree for "to take-away" is developed. Must include a special check to see if what is being taken away is desirable. "They took away their pain" would not be scored because pain isn't desirable. "They took away their privileges" would be.
Also there should be developed a method of handling possessive pronouns. For example: "They took away their rights" means "they took away rights from them which would be scored differently than "They took away my rights".