

PARSING DIRECTLY INTO KNOWLEDGE STRUCTURES

Roger C. Schank, Michael Lebowitz, and Lawrence Birnbaum
Department of Computer Science
Yale University
Box 2158 Yale Station
New Haven, CT 06520

A new type of natural language parser is presented. The idea behind this parser is to map input sentences into the deepest form of the representation of their meaning and make appropriate inferences during the parsing process, using interest to guide the processing.

1. Introduction

Over the course of the last ten years, researchers in our project have designed and programmed a large number of parsers - programs that mapped English sentences into the Conceptual Dependency (CD) representation of their meaning. They have all had one methodological assumption in common: the parsing algorithm that they employed was to be as psychologically correct as possible. This assumption brought with it an operating principle which was (with one exception to be discussed later) always followed, namely that the parsing algorithm was a strict left-to-right, one-pass operation, without backup.

One of the major problems with most of our story understanding programs, including their parsers, was their inability to handle genuinely new texts for which they were unprepared. A new vocabulary item, domain of discourse, or previously unencountered syntactic construction would often throw things into disarray. It seemed to us that we could produce one of the desired end-products of a story understanding system, a summary, with a much more robust and much faster program. This was the reason behind the design of FRUMP [1]. FRUMP skims for what it is interested in, looking for precisely the items of information it wishes to include in its summary for any particular domain that it has knowledge about. However, since FRUMP is very top down it cannot respond to things it is unprepared for. Therefore it is not a complete understanding

This work was supported in part by the Advanced Research Projects Agency of the Department of Defense and monitored by the Office of Naval Research under contract N00014-75-C-1111.

system.

Ideally what we would like is a program with the robustness of FRUMP but with the ability to understand what it does not expect to see. Perhaps a human understander is top-down when appropriate, but bottom-up at other times. What is needed is an understanding system that can do both and be robust at the same time.

Many other researchers have investigated the problem of parsing natural language input into an internal representation. Virtually all of their methods concentrate on syntactic analysis. One very popular technique has been the Augmented Transition Network (ATN). Parsers of this sort have been discussed by Woods [10], Kaplan [2], and others. ATN's have tended to deal almost exclusively with syntax, perhaps with an occasional check of a few simple semantic properties of words. A more recent parser which views syntax parsing as an isolable sub-part of language understanding is by Marcus [3]. All of these programs view syntactic parsing as a process largely isolable from the rest of understanding. Our view, as described by Riesbeck [5] and Riesbeck and Schank [6] has always stressed the integration of semantics and syntax in parsing. A similar view was expressed by Wilks [9]. In this paper we will further contend that parsing should be integrated with the overall understanding process.

2. Processing Limitations

Now let us consider the design of a human-like parser. One important feature in such a parser is the speed with which humans can read text. Considering all the inferences, access of background knowledge, and other problems that an understander must deal with in the course of reading or listening to a sentence, people are very fast at the job. This is especially true

when we recognize that understanding is basically finished as soon as a sentence ends. This implies that high level processes cannot wait until the end, after parsing is finished, and also that human processing is integrated. People must be making inferences as soon as they begin to process a sentence. From this it follows that people will actively use whatever they discover, using high level conclusions to guide low level processing. Thus, as models of human processing, systems that first do a complete parse and then begin to make inferences are not sensible. It seems plausible that the processing requirements for complete understanding frequently overwhelm the total available processing capability (see [4]). This means some words must be processed very superficially in order to gain time to extensively process words of semantic, syntactic or inferential importance.

It is likely that people are deciding as they go what words to pay serious attention to and what to glide over. Such decisions can be explained on the basis of many factors. One obvious one is interest. That is, people are liable to pay attention to (that is, devote their processing time to) what interests them. We have discussed the concept of interest and its ramifications for the inference problem in [7]. The main conclusion there was that inference is controlled by interest. This is also likely to be true at the parsing level since we are now viewing the entire understanding process as an integrated phenomenon.

Taking advantage of interest to reduce the time spent on many words has certain implications on processing. One is that the understanding process may not be strictly left to right. Since the most important words often come at the end of a phrase, the preceding words may be skimmed over until they are known to be important and then 'gathered up.' To do this the understanding process must be top down in order to allow the understander to know what to ignore. Uninteresting words are simply stored in a buffer in short term memory until a word that requires them to be processed is found. When (and if) such a word is found, the words in the buffer are gathered up and analyzed more fully. Under this theory there is little wonder at the fact that understanders frequently cannot remember the actual words that they read. They may never have actually read them at all!

3. A Detailed Example

A front page story in the New York Times began, "An Arabic speaking gunman shot his way into the Iraqi Embassy here [Paris] yesterday morning, held hostages through most of the day before surrendering to French policeman and then was shot by Iraqi security officials as he was led away by the French officers."

We will examine this sentence word by word and consider the kind of processing that would be desirable in an integrated understanding scheme. Our model will use interest to enable it to be psychologically plausible, a requirement forcing it to be completely finished with each sub-part of the sentence as it is processed.

Since language is received as a stream of words, we can assume words become available in chunks, both visually (clauses) and aurally (delineated by pauses). Thus in our processing we can always assume that the next word of a sentence is available, often simplifying the problem of disambiguation.

An Arabic speaking gunman...

AN is a word that can be skipped initially. This means that it is looked up in the dictionary, and what is found there are instructions to go to the next word and place AN in short term memory (STM).

ARABIC is listed in the dictionary as a word that is skippable, so it is skipped and saved in STM. (The information that ARABIC, like many uninteresting adjectives, is skippable has already been compiled and it is simply looked up here. The procedure for determining what can be skipped is obviously one of the interesting problems in the issue of the development of language ability. Schank and Selfridge [8] discuss these issues.)

SPEAKING is also skippable as long as no potential actors have been encountered. A search for actors in STM finds none, so this word is also skipped.

GUNMAN is marked as a HIGH INTEREST ACTOR, demanding immediate processing. It causes us to create top down requests to find certain information, such as the answers to the following questions:

WHO is he? -- causes us to gather up saved
 adjectives and add them to
 the memory token for this GUNMAN

WHAT did he do? --
 in GUNMAN's permanent memory we find
 he SHOT someone. An inference that
 he shot or will shoot is made and
 we predict we will hear more about it.

WHO did he shoot? -- creates interest in the
 verb's syntactic object.

WHY did he shoot? -- makes us look for a reason

WHERE did this happen? -- makes us look for
 a location

WHAT SCRIPTS might this instantiate? --
 GUNMAN can indicate a script may be
 instantiated. Candidates are \$ROBBERY,
 \$TERRORISM, \$KIDNAP. We try to confirm.

The formulation of the above questions (as
 requests, see the next section) now guides the
 process of understanding this story. These
 requests relate to matters of parsing,
 inference, script application and goal pursuit
 - all levels of the understanding process.

 shot his way into the Iraqi Embassy...

SHOT is encountered and found to satisfy an
 expectation created by GUNMAN so we now build
 the first conceptual structure, a SHOOT event
 with the gunman token as actor, and an unfilled
 slot for the victim. The parser is now
 interested in finding the victim, which can be
 done by looking for the next main noun.

HIS is skipped and saved in STM.

WAY does not satisfy the expectation to fill
 the empty slot. WAY is listed as both
 skippable and pointing to a direction or
 location. A request is set up for the location
 and we attempt to find it.

INTO is skipped.

THE and IRAQI are skipped and saved in STM.

EMBASSY is found and set up as the location of
 the SHOOT event. Furthermore, EMBASSY is
 marked as a place of political significance.
 This piece of information causes us to
 instantiate the \$TERRORISM script that had
 (among others) been predicted from GUNMAN.
 Since EMBASSY is interesting, its requests are
 activated. One of these is for the nationality
 of the EMBASSY. IRAQI is found in STM as
 satisfying this request and is picked up.

Setting up \$TERRORISM causes us to lose
 interest in the representation of the isolated
 sentence and focuses us on setting up that
 script as the representation for the entire
 story. Thus, we now expect answers to the
 following questions:

Were HOSTAGES taken?
 Were demands made? (money?) (free prisoners?)
 Was any damage done?
 Were measures to counteract the terrorist
 taken? (return fire; arrest; free hostages)

 here yesterday morning...

HERE, in a news story, means to add the
 dateline to the location.

YESTERDAY is found to be a time word and is
 added to the time slot of the event.

MORNING is also handled in this manner.

 held hostages through most of the day....

HELD is skipped and saved since it matches none
 of the requests. The dictionary entry for HELD
 says only that it is a verb. No verbs were
 needed so we skip it. Only had it been
 inherently interesting would we have done
 detailed processing. The advantages of this
 system of parsing are clearly shown here. HELD
 is a highly ambiguous word that previously
 might have demanded great effort to
 disambiguate. An integrated understanding
 system simply goes on and waits for something
 interesting.

HOSTAGES satisfies an extant request. The TAKE
 HOSTAGES scene of \$TERRORISM is instantiated.
 At this point a check is made on the saved
 verb, HELD, to see if doing this is okay. HELD
 is found to be precisely the kind of word
 desired. The important point is that HELD
 never had to be disambiguated, which is nice
 because words like HELD really do not have any
 particular meaning. Its meaning is derived
 from its connection to HOSTAGES and HOSTAGES is
 understandable only through \$TERRORISM.

THROUGHOUT is found to point to either a time
 or place so a request is made for a time or
 place word. However, at this point our
 understanding system knows that this
 information is not as interesting as the
 outstanding expectations, because there are

death-related requests active (see [7]). Thus the rest of this phrase is virtually ignored due to lack of interest.

MOST, OF and THE are skipped and saved in STM.

DAY is saved and ignored. It satisfies the low level request for a time word.

before surrendering to French policemen...

BEFORE is a time ordering word that predicts a new event and marks its time relative to the preceding event.

SURRENDERING is a word that is marked as an important part of a number of scripts, including \$TERRORISM. The surrender scene of \$TERRORISM is instantiated and requests are fired off looking for reasons for this action, the captors, etc. Predictions are made about certain words that might follow. 'To' would mark off captors, for instance.

TO tells us that the captors are coming (from the prediction).

FRENCH is skipped and saved in STM.

POLICEMEN is marked as an ACTOR so STM is consulted to gather up any modifiers. POLICEMAN is a possible captor, so it fulfills an expectation.

and then was shot by Iraqi security officials..

AND says a new event is coming.

THEN orders the event.

WAS specifies that the actor stored in STM is the conceptual object of the new event. This sets up requests for the actor, indicated by 'by', and the action.

SHOT is found to be interesting and is treated similarly to the way that GUNMAN was except that we expect only the things that were particular to the SHOOT action. Thus we have:

WHO was shot? -- GUNMAN
WHO shot him? -- not answered
WHY was there a shooting? -- not answered
WHERE did this happen? -- already known

WHAT SCRIPTS does this instantiate? --

SHOOT can also instantiate a script. Candidates are \$ROBBERY, \$TERRORISM, \$KIDNAP, ordinarily. But in a context set up by \$TERRORISM, none of the above fit easily. We look for plans and goals.

WHAT were the RESULTS of this action?--

A request is set up to find the results. If not satisfied the usual results are inferred - the victim's death.

BY tells us the actor of the SHOOT will follow.

IRAQI and SECURITY are skipped and saved in STM.

OFFICIALS ends the processing of the noun group. It satisfies the requests for WHO did the shooting and, as we now have an actor we ask about why he would kill a TERRORIST. There is no obvious reason, so we are surprised by this event. We seek to explain it by postulating a REVENGE or SHUT HIM UP type theme, but this is a guess.

as he was led away by French officers.

We are basically done now as no further requests need to be satisfied immediately. We are still interested in the goals of each of the actors, however, so WHY requests are still alive.

AS is known to be a time co-occurrence word. Since we are only interested in anything that occurred at the same time if it is itself interesting, we can now skip ahead looking for something interesting.

HE and WAS are skipped.

LED is uninteresting and is both noticed and then skipped.

AWAY, BY, THE, and FRENCH are all skipped.

OFFICERS is skipped because there are no requests asking for it.

The period tells us we are done.

The final representation for this sentence:

\$TERRORISM	UNEXPECTED RESULT
ACTOR Arab gunman	-----
PLACE Iraqi Embassy	\$SHOOT
SCENES	ACTOR Iraqi officers
\$HOSTAGES some	OBJECT - Arab gunman
\$CAPTURE	RESULT
ACTOR French policemen	STATE dead
OBJECT Arab gunman	ACTOR Arab Gunman
PLACE Iraqi Embassy	

4. An Integrated Partial Parser

We will now look briefly at the integrated partial parser (hereafter IPP) we have written to implement the theories we have been describing here. This program was written to handle a limited class of stories, namely newspaper stories about terrorism and related areas. We have not tried to address all the issues involved in parsing, instead concentrating on the areas crucial to integrated partial processing. The parser, written in LISP on a DEC System 20/50, currently handles over 50 stories taken directly from newspapers. We are in the process of increasing its vocabulary and its domain of interest.

IPP is structured about a classification of words, which determines on how each word type should be processed by an understander, independent of strict syntactic classification. The immediate fate of a word upon being read determines its primary classification. There are three classes. The first consists of words requiring immediate action. These are interesting words which add crucial information to our story representation and generate expectations which guide further processing. The class is subdivided into words which build event structures (Event Builders, or EBs) and words which fill in slots in event structures (Token Makers, TMs).

EBs build the event structures which provide the framework for a final representation. Events structures which are predicted or interesting by themselves are instantiated and their expectations activated. Events which are not predicted, and not interesting are virtually ignored.

The discovery of a TM causes a token to be created. If the token satisfies an expectation, is interesting, or an interesting modifier is in STM, then the modifiers in STM are applied to the token. If the token is interesting the expectations included in the

TM's dictionary entry, are activated. These expectations may, for example, look for plausible scripts.

The second class of words do not require immediate attention, but may be needed later. These words are processed by a skip and save strategy, keeping them in one of several short term memory locations until, and unless, they are needed. This typically occurs when an interesting EB or TM is found. Frequently these words will never be used again, making the total effort needed to process them negligible.

The final class of words are those we skip entirely. The only processing effort is in identifying them. While few words are skippable in all domains, in any particular domain, a surprisingly large number of words can be totally skipped without degrading understanding. Clearly it is advantageous for an understander if this class is as large as possible, since processing time for skippable words is negligible.

Much of IPP's processing knowledge is implemented in the form of requests [5]. A request is a form of production, or test-action pair. If the test of an active request is found to be true, then the corresponding action is performed.

The tests and actions performed by IPP requests perform only a limited set of functions. As actions, requests may build new conceptual structures, fill slots in conceptual structures with other conceptual structures, activate new requests, and de-activate requests no longer appropriate. The tests may check for concepts (tokens or events) of a specified type, specific lexical items, or lexical items satisfying some property.

5. Computer Runs

What follows are computer runs of IPP on stories from the New York Times.

*(IPP S1)

Input:

(AN ARABIC SPEAKING GUNMAN SHOT HIS WAY INTO THE IRAQI EMBASSY HERE THIS MORNING HELD HOSTAGES THROUGHOUT MOST OF THE DAY BEFORE SURRENDERING TO FRENCH POLICEMEN AND THEN WAS SHOT BY IRAQI SECURITY OFFICIALS AS HE WAS LED AWAY BY FRENCH OFFICERS)

Output:

** MAIN EVENT **

SCRIPT	\$TERRORISM		
ACTOR	ARAB GUNMAN	PLACE	IRAQI EMBASSY
CITY	PARIS		
SCENES			
SCENE	\$HOSTAGES	SCENE	\$CAPTURE
PLACE	IRAQI EMBASSY	PLACE	IRAQI EMBASSY
ACTOR	ARAB GUNMAN	OBJECT	ARAB GUNMAN
		ACTOR	POLICEMEN

** UNEXPECTED EVENTS **

ACTION	\$SHOOT		
ACTOR	IRAQI OFFICIALS	OBJECT	ARAB GUNMAN
AFTER	SCENE \$CAPTURE		
RESULT			
STATE	DEAD	ACTOR	ARAB GUNMAN

673. msec CPU (0. msec GC)

*(IPP S2)

Input:

(A GUNMAN WHO DIVERTED A VERMONT BOUND BUS WITH MORE THAN TWENTYFIVE PASSENGERS FROM THE BRONX TO KENNEDY INTERNATIONAL AIRPORT AND KILLED TWO HOSTAGES SURRENDERED ON A RUNWAY LATE LAST NIGHT ENDING A DAYLONG SIEGE OF TERROR AND GUNFIRE)

Output:

** MAIN EVENT **

SCRIPT	\$HIJACK		
ACTOR	GUNMAN	CARRYING (GREATER THAN 25)	
VEHICLE	BUS	PASSENGERS	
RELATED			
ACTION	DO	RESULT	
ACTOR	GUNMAN	STATE	DEAD
		ACTOR	TWO HOSTAGES
SCENES			
SCENE	\$CAPTURE		
OBJECT	GUNMAN		
ACTOR	POLICE		

718. msec CPU (0. msec GC)

References

- [1] DeJong, G. F. (1977) Skimming newspaper stories by computer. Research Report 104, Department of Computer Science, Yale University.
- [2] Kaplan, R. M. (1975) On process models for sentence analysis. In D. A. Norman and D. E. Rumelhart, eds., Explorations in Cognition. W. H. Freeman and Company, San Francisco.
- [3] Marcus, M. (1977) A theory of syntactic recognition for natural language. Unpublished Ph.D. thesis, MIT.
- [4] Norman, D. A. and Bobrow, D. G. (1975) On data-limited and resource-limited processes. Cognitive Psychology, Vol. 7, pps. 44-64.
- [5] Riesbeck, C. K. (1975) Conceptual analysis. In R. C. Schank (ed.), Conceptual Information Processing. North Holland, Amsterdam.
- [6] Riesbeck, C. K. and Schank, R. C. (1976) Comprehension by computer: Expectation-based analysis of sentences in context. Research Report 78, Department of Computer Science, Yale University.
- [7] Schank, R. C. (1978) Interestingness: Controlling inferences. Research Report 145, Department of Computer Science, Yale University.
- [8] Schank, R. C. and Selfridge, M. (1977) How to learn / what to learn. Fifth International Joint Conference on Artificial Intelligence, August 1977, Cambridge, Massachusetts.
- [9] Wilks, Y. (1973) An artificial intelligence approach to machine translation. In R. C. Schank and K. Colby, eds., Computer Models of Thought and Language. W. H. Freeman and Co., San Francisco.
- [10] Woods, W. A. (1970) Transition network grammars for natural language analysis. Communications of the ACM. Vol. 13, p 591.