KNOWLEDGE STRUCTURES AND LANGUAGE BOUNDARIES

Yorick Wilks
Dept. of Language & Linguistics,
University of Essex,
Colchester,
ENGLAND.

Abstract

The paper discusses the incorporation of richer semantic structures into the Preference Semantics system: they are called pseudo-texts and capture something of the information expressed in one type of frame proposed by Minsky (q.v.). However, they are in a format, and subject to rules of inference, consistent with earlier accounts of this system of language analysis and understanding. Their use is discussed in connection with the phenomenon of extended use: sentences where the semantic preferences are broken. It is argued that such situations are the norm and not the exception in normal language use, and that a language understanding system must give some general treatment of them.

Descriptive terms: natural language understanding, semantics, frames, pseudo-text, preferences, templates, formulas, thesaurus.

Introduction

This paper sketches how one might deal with extensions of word sense in a natural language understanding system (NLUS): that is to say, normal utterances that break preassigned semantic selection, or preference, restrictions. The proposals here extend the knowledge representation of the preference semantics NLUS (Wilks 1968, 1975) with pseudo-texts (FT) which are one type of frame structures in the zense of (Minsky 1975), but which are also consistent with the general assumptions of this particular NLUS. At. the end I shall describe, an implementation environment under construction, which may make possible some test of the relative contributions of PTs and very general pragmatic principles like "preference" (see below) to language extension.

To understand "preference" consider the following sentence, chosen, I promiseyou, at random from the front page of a daily newspaper: (The Times 5.2.76):

(1) Mr. Wilson said that the line taken by the Shadow Cabinet, that an Assembly should be given no executive powers would lead to the break-up of the United Kingdom.

This sentence presents no problems to the average reader of that newspaper, who is presumed to know what a cabinet is, and what the U.K. is. However, at each of the four underlined points, the noun would violate the normal semantic selection restrictions for the associated verb: lines, for example, would violate the normal "physical object" restriction on "take" and so on.

I shall refer to such restrictions as preference restrictions, because of the way the present NLUS is already able to accept natural language that violates preferences, as (1) does (see recap in next section for more detail). Such usage as (s) will be referred to as extended, or preference violating, and these will serve instead of the more literary and philosophical term "metaphorical".

It is an important assumption of this paper that such usage is the norm in ordinary everyday language use, and cannot be relegated to the realm of the exceptional, or the odd, and so dealt with by considerations of "performance". On the contrary it is, I would argue, central to our language capabilities, and any theory of language must have something concrete to say about it. Even if the newspaper usages above are "extended", I would suggest that anyone who could not grasp these extension could not be said to understand English properly (given adequate knowledge from which to extend, and we shall come to that.) It will be obvious already that the commitment to a norm implies a corresponding commitment to general everyday language as a proper topic for AI. This assumption needs defence, but there is no space for it here. However, one might bear in mind that, although non-general micro-worlds have been put forward as the E.Coli et AI, they may in fact turn out to be our phlogiston!

Semantic Sense-Projection

The process described in this paper is called projection: sense descriptions for words will be rewritten, in preference-violating texts, with the aid of the specific knowledge in PTs; which is to say that part of the PT is projected into the sense description for a word. So, for example in (1) some detailed political knowledge in a PT (see below) for "United Kingdom" could show that a breaking of such an entity could be caused, and we would then replace the sense description of "lead to the break-up" and providing a more appropriate sense description of "lead to for analysis of the rest of this text.

The essence of this process is finding in the appropriate PT, what it is that can normally be done with an entity of that type. As we shall see, matching this is not always a simple matter. If, with the aid of knowledge representations and sense descriptions, projection, even to a small degree, and generalized manner, we shall, at the same time, be able to explain why the same acceptable extenders use is not present in, say,

"He broke up evil",

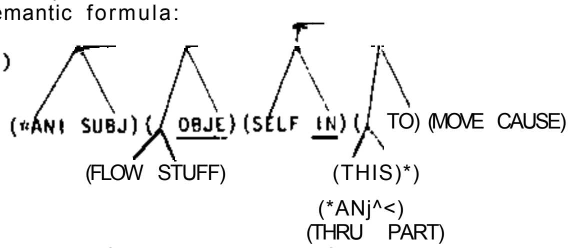
Thus the present task of explication will, to that extent* make concessions to the main linguistic goal of the last two decades, that of setting boundaries to acceptability or meaningfulness, at least in so far as (1) would interpretable on the basis of a knowledge base, and rules of extension, while the last sentence would not.

However,f some brief recap of the existing state of the NLUS is necessary for setting out these extensions.

Brief recap of preference semantics

In previous papers I have described an NLUS in which rules operate on semantic word-sense descritpions to build up text descriptions. The tules that insert sense descriptions into text descriptions are what 1 called "preferential": they seek preferred entities, but will accept the less preferred if necessary. A sense description for the action "drink" would be the semantic formula:

(2)



This is a formal structure of semantic primitives expressing the meaning of the action (see King and Wilks 1977): that drinking is a CAUSing to MOVE, preferably done by ANImate SUBject (-agent) and to a liquid (FLOW STUFF). TO a particular ANImate aperture (THRU PART), and INto the SELF (-the animate agent). The last primitive cause, is the head of the formula and its main primitive. For short we will write (2) as |drink|.

The text structures in the system are templates together with semantic ties between them, where a template is a network of formulas, containing at least an agent, action and object formula (or appropriate dummies in place of them). Thus the template for ^"The adder drinks water" will be written for short |the+adder drinks water] in which (2) is at the (action) node.

The process of setting up the templates allows the formulas to compete to fill noes in templates. Thus tge formula for the (snake-)adder goes to the agent node in the template above in preference to the (machine-)adder because (2) specifies, by (vANI SUBJ) that it prefers to be accompanied in a template by an animate agent formula. However, in the sentence:

(3) My car drinks gasoline.

the available formula for the first template node namely Q:ar3, is not for an animate entity, yet it is accepted because there is no competitor for the position. THE PURPOSE OF THIS PAPER IS TO INVESTGATE HOW THE SYSTEM MIGHT NOT MERELY ACCEPT SUCH A PREFERENCE-VIOLATING STRUCTURE F0R(3) BUT MIGHT ALSO INTERPRET IT.

An important later process is called extraction: template-like structures are inferred and added to the text representation even though they match nothing in the surface text. They are "deeper" inferences from the case structure of formulas in some actual template—where the case primitives are those underlined in (2). Thus, to the template for (3), we would add an extraction (in double square parentheses in abbreviated form):

(4) |gasoline in carJ J

which is an inference extracted from the containment subformula of (2) (SELF IN). Analogous extractions could be made for each case primitive in each formula in the template for (3).

All these are, of course, complex and contentous issues, that can only be summarised here so that we can get on to something else. They have however been programmed and described in detail in (Wilks 1975, 1976).

Since then a structural change (Wilks (1976a) as allowed a wider, and more specific, form of expression in formulas by allowing thesaurus items, as well as primitives, to function in them, No problems are introduced by doing this, provided that the thesaurus items are also themselves words in the dictionary, and so have thcirformulas defined elsewhere in their turn. One advantage of this extension is to impose a thesaurus structure on the whole vocabulary, and so render its semantic expression more consistent.

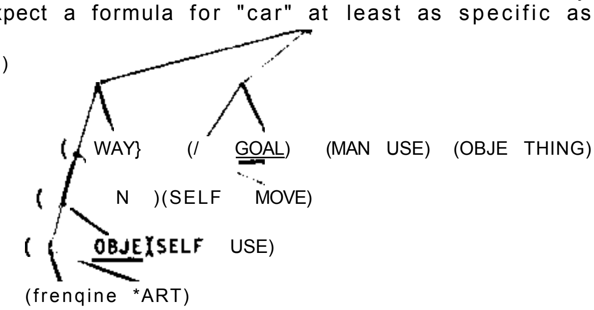
A thesaurus, Like Roget, is simply an organisation of a vocabulary into semi-synonymous rows, which are themselves classified hierarchically under heads, and even more generally, sections. Thus under some very general section name MOVE (■motion) we would find heads, two of which might be Aengine and vehicle. The former might be the name of a row of actual types of engine

(S) # 525 engine: turbine, internal combustion, steam......

where the number simply indicates the sequence position of # engine in the thesaurus. It is no accident that the most general section names like MOVE can be identified with the semantic primitives of the present system.

The organisation is imposed by requiring inclusion relations, between the formulas for word senses, corresponding to the thesaurus relations of the words. Thus, all the words in the row (5) would have a common subpart to their formulas, and that common subpart would be the dictionary formula for "engine , probable expressing in primitives no more than"a thing used by humans to perform some task, and self-moving in some way". If now thesaurus items can be inserted in formulas we may expect a formula for "car" at least as specific as:

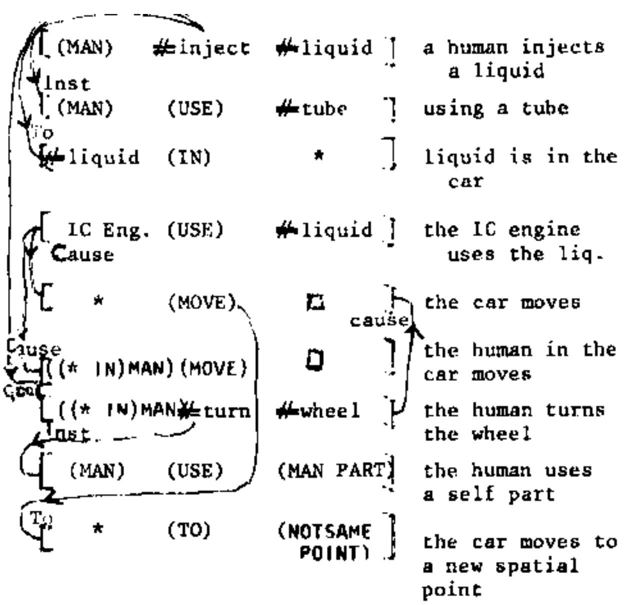
(6)



Language Boundaries and Projection

Let us return to examples like (3) for which the system constructs a template even though it contains a violated preference, and ask what should an intelligent system infer in such a situation?* I would suggest that cars can be said to drink in virtue of something a system might already know about them, namely that they have a fluid (gas/petrol) injected into them, and they use that in order to run. That is to say, the program should have access to a sufficiently ric h k nowledge structure to express the fact that cars stand in a relation to a particular fluid, a relation that is, of the "same semantic structure" as the relation in which a drinker normally stands to the thing drunk. All that may sound obvious, but how else are we to account for the naturalness of (3), but the relative unnaturalness (and uninterpretability) of "My car chews gasoline", and, the more distant, "My car carves the Sunday roast". One upshot of these proposals is to distinguish plausible (with respect to a knowledge base) preference violation from the implausible.**

The procedural upshot of the above would be to replace at least one formula in the template for (3) with another, either constructed by rule*** or drawn from the knowledge structure itself, to be called a pseudo-text (PT). Let us now postulate that "car" points not only to (6), i.e. |car] but that |car] in turn points to :



Part of the pseudo-text for "car"

The system already deals with certain preference violations, such as those constituting the ergative case paradigm ("The hammer broke the window" - see Wilks 1976b) and certain examples like "John got a shock", a class central to Riesbeck's thesis (see Schank (ed.) 1975).

** An important aspect of the interpretation of (3) is idiomatic, namely that the car uses _a lot of gas/petrol. This aspect of the meaning is beyond this, or I suspect any, general inference procedure.

*** In a fuller version of this paper (Wilks, in press) I describe the relation of this work to attempts, such as (Givon 1967), to give general rules for projection: rules operating on the dictionary and independent of contexts of use.

This structure is called a pseudo-text because it is of just the same format as the text representations produced by the present NLUS.It can be extended to taste to express as much specific information about cars as is thought appropriate. Given the parser for the present NLUS, it could even be input as real text about cars. The representation consists of the templates (explained loosely at the right), together with the self-explanatory) case and cause ties between them. In the templates,; denotes a dummy and * denotes the formula [car] that points to the object (7). The # prefixed items are thesaurus items, though the "IC engine" is simply a specific-dictionary word pointing to its own formula - - specificity is thus a matter of taste. So, for example, the thesaurus head #* liquid could be replaced by more explicit "gasoline". Items in round parentheses remain in primitive form.

It will be clear that the same information can be expresses in a number of different ways, and at different levels of generality; though the spirit of (Minsky 1975) suggests that they should be as specific as possible. The intention here is that THE PROCESSES THAT OPERATE ON SUCH ENTITIES AS(7) SHALL BE IDENTICAL WITH THOSE THAT OPERATE ON SUCH ENTITIES AS (7) SHALL BE IDENTICAL WITH THOSE THAT MANIPULATE REPRESENTATIONS DERIVED FROM INPUT TEXTS. The approach is thus the reverse of the conventional one: we seek to assimilate knowledge structures to text structures, rather than the reverse, on the grounds that the representation of language is the difficult task, and that the representation of knowledge as such makes no sense apart from that.

We should note too, that just as the thesaurus structure imposes a containment relation on the formulas of co-row-member words, so it also imposes a hierarchical relationship on PTs: that for # vehicle, for example, will be a less specific version of (7). Further up the thesaurus would be PTs for high-level sections: that for the primitive Man would be highly complex. But note there is no "inheritance of property" problem in this system: the formula for "amputee" would have head MAN and would specify the loss of limbs. Any inherited pseudo-text from MAN-asserting "two legs" -would be modified [amputee].

The system now uses (7) to make a projection, so as to derive an interpretation for (4), by seeking, in (7), templates matching the source template |ray+car drinks gasoline}: namely the first and "fourth lines of (7). The first match is in virtue of the similarity of [drink] and \# inject]—based on the expression in primitives, as in (2), of causing liquid to be in an entity of the same type as the agent. This would allow us to confirm, by projection, the "humanness of the drinker", that has already been noted by earlier extraction* routines that extracted out from [drink| independently of the PT (7). However, no projection is made at this stage onto |carj (though it might be later in the face of a sentence like"His thirst is never slaked following (4), that confirms the humanness projection) because,in the case of violations of the preferences of actions, as of "drink" in (4), the system ALWAYS PREFERS TO MAKE A PROJECTION ON TO THE ACTION ITSELF IF IT; CAN.

A stronger match is detected btween the |my+ car drinks gasoline| and the fourth line of (J) in v irtue of the containment of (^engine*) in |car], and of |gasoline| in #liquidj, which is evident in the formulas themselves. This results in the projection of the action node of the fourth line of (7), namely [use], on to |drink] in the template fir (3). This projection is taken to be strongly confirmed by the earlier match with the first line of (7), and is considered to carry over more sense that any alternative projection. The confirmation (of the match of the template to the fourth line of (7) by that of the first line) is necessary here, because |my+car leaks gasoline] would also match the fourth line, but no such projection would be appropriate. Conversely, no projection could be made for "My car drinks mud" from the fourth line, even with the confirmation of the first. The general rule for action projections then is: SEEK A PSEUDO-TEXT, FOR AGENT OR OBJECT, WITH A TEMPLATE MATCHING ON AGENT AND OBJECT NODES. PROJECT THIS GENERALLY IF THERE IS ALSO A PSEUDO-TEXT TEMPLATE MATCH TO THE ACTION ITSELF, FOR ANOTHER TEMPLATE IN THE SAME PSEUDO-TEXT.

We may note in brief because of pressure of space,above suggestion. First consider the more complex example presented by a recent headline:

(8) United Kingdom tries to escape Common Market Clearly, some projection would be appropriate here, of humanness on to the country (which would require a MAN-head for [U.K/];< [Common Market]. The knowledge required might be drawn from |escape| alone, by simple extraction and without recourse to the pseudo-texts for either of the entities of the U.K. joining, but not of leaving the Common Market. In such circumstances more historical facts are not enough, even when highly structured. We might conceivably be able to project some notion fclisassociatej onto[escape 1], from the U.K. PT given some more sophisticated matching criterion that placed relevance above negation in such cases (i.e. would match |escape"] with [associate] or [join].

Secondly,we might consider the problems presented by an example like:

(9) I see what you mean. Here the Jast clause breaks the preference expressed in |see| for a physical object. A system procedure wfll present the actual object of (9) to the top-level template simply as the primitive SIGN (the primitive for symbols and intensional representations of them) which has been obtained, by extraction, from the preferred object in[meanj .Thus the system is effectively dealing with the template sequence. [i see (SIGN)] |you man (SIGN)J . But what could we expect as a pseudo-text for something as general as SIGN, so as to use the above procedures to project on to [see]. If we take advantage of the hierarchical nature of the thesaurus, we might expect pseudo-texts at the very top level, associated with the section names - pure primitives like SIGN - just as specific pseudo-texts are associated with the lowest lever1 items in the thesaurus - row members like "car". The pseudo-text for a primitive like SIGN would be wholly "care structural": it would consist of no more than primitive concatentions, in template form, like MAN THINK SIGN*, the most general thing that can be said about what is normally done to signs. However, even something as general as this might suffice to project THINK correctly onto [see]. The interesting generality would come from using exactly the same projection procedures on the most general pseudo-texts like this, as on the most specific, like (7).

Thirdly, and this is treated at length in Wilks (1977a), we can consider a quite different type of projection for phrases like (10) a toy lion.

This comes from a much discussed class of examples ("plastic flower", "stone horse" etc.), where an obvious projection mechanism is to replace the head of the formula for the noun (BEAST) in [lion] in (10)) by the preferred object of predication in the qualifier — here *PKTS0B ill [toy]. This would be a very limited and general class of projections,not requiring access to PTs, but which might still provide a "projected formula" appropriate for examples like: (11) The cat walked round the toy lion.

* extractions, it will be seen, differ from projections in that they produce new template-like entities, rather than, as here, replacing formulas inside existing templates.

* those familiar with the system of Wilks (1968) 1965 etc.) will remember that these are the "bare template" structures actually used to obtain the initial template match. The suggestion here is that the "knowledge-aspect" of these highly-general structures is to be found as the pseudo-texts of primitives - as the latter function right at the top of the conceptual hierarchy imposed by the thesaurus.

Then he came back and sniffed it.

Where we might be helped to refer "he" and "it" correctly by the new, projected, formula |lion| whose head was no longer BEAST, and which could therefore no longer be the reference of "he" as a real lion would be.

A more radical and interesting development would be the construction of "PT repacking functions" specific to certain qualifiers. Thus, for example, such a function for "toy", if faced with the phrase "toy car" might repack (7) using a general rule to delete all constituent templates based on the action USE, as well as all those that are at end of a GOAL tie, since toy cars cannot, normally serve human needs, uses and purposes.

An Implementation Environment

It is proposed to implement these suggestions currently using a new base for the preference semantics system, expressed as a semantic augmented transition network. It is hoped that this more conventional re-expression of the semantics-driven pattern matching parser will make it more comprehensible and acceptable. The environment will be a question-answering system that discusses in English, the construction of the semantic representation itself for a text. It is hoped too that this will avoid the trivialities that lurk in answering questions about mini-worlds, where the answers to all possible questions sometimes seem preformed in the data base. Some simulation of parallel processes is hoped for, so as to distinguish (as independent matching processes) those aspects of the present system that correspond to local context (like preference) and those that correspond to a global context(like PTs and the use of the thesaurus generally), and to apply these independently to a text, as parallel branches of analysis, each seeking template-like structures in the input text. The interest, and the possibility of test of the relative usefulness of the branches, would come from switching off some particular branch and seeing if the analysis deteriorated.

Let us imagine such a dialogue as it might apply to the analysis of a real newspaper text like: (The Times: 5.2.76).

(12) An ambulance driver, answering an emergency went through red traffic lights and collided with a car killing its driver. The interest here is how the program is to determine the reference of "its" as the car, rather than the ambulance driver. In what follows, the user interventions are in the upper case, notes are preceded and followed by square brackets.

END OF TEXT INPUT

1. An ambulance answering an emergency call
2. S went through red traffic lights
3. and S collided with a car
4. S killing its driver

[S marks missing template agents in the four fragments the text has been cut into for ease of reference"]

All fragments match templates.
Template 2 fails preferences for agent.
Now repacking S slots

2. The ambulance driver went through red traffic lights.

WHY?
a driver is mobile, a call not.

3. The abulance driver collided with a car.

Suggest:

4. The ambulance driver killed ? its driver

(system unsure of S in (4), it is marked as a problem]

WHY?
only active agent available

But now re-inferring as follows:

preference breaking in (2) drives inferences as follows:
traffic lights are aperture

"went through "means "passed

I select latter and project on to "went through"

|straightforward extraction in (2) plus access to pseudo-text for "traffic-lights"]

examining preferences in

4. S killing its driver.

"It" is lights (number wrong)
or
call
or
car
or
ambulance

Prefer car or ambulance for "it"

[calls don't have driversj

Confirm

4. ambulance driver kills car's driver

WHY?

First:

from traffic-lights pseudo-text

passing red traffic light

causes people hurt

['pass" has replaced "went through" by projection, remeraber]

[pseudo texts can have a "cause tied" template pair read out] this confirms kills car and ambulance driver.

Second:

Calling pragmatic over-ride

its-cars

"its" not twice coreferential

[a general Gricean* style pragmatics rule that maximises information in the fragment.]

Confirm its=cars

Representation complete.

The new point brought out of the hat at the end, as it were, shows the need for general pragmatic principles at the highest level. Just as we needed preference at the lowest. The general interest here would be the possibility of two clashing general principles in the analysis of a given text: preference seeking, in some sense, to minimise information (argued in Wilks 1975), and another seeking to maximise it.

Discussion

The cause-tied template pair, read out from the PT for fragment 4, shows that the function of the PT is not tied to just simple matching and inferences, and can span across texts of some length.

Moreover, in an actual implementation, the hierarchical organization of the PTs would plan a stronger role than appears here. As was suggested in connection with example (10), the PTs would not all be stored explicitly, but would be constructed as required * from the more general PT for the corresponding thesaurus sub-head term, together with appropriate slot fillers. Thus, the "ambulance example" would always access the *4#* vehicle PT initially and would attempt to deal with the text at that more general level, before proceeding to construct the "ambulance PT" from the *4h** vehicle one.

A number of very general comparisons and issues suggest themselves here. The PTs are clearly of the more static type of frame adumbrated in (Minsky 1975), in that, unlike those of (Charniah 1975) (Schonk 1975b) etc., they do not have a narrative line, temporally or causally ordered. Indeed, (7) is essentially unordered, merely connected, and, should unconnected sub-PTs exist in a PT, they may be considered connected by an unordered AND predicate.

So, since PTs are non-narrative, or in crude terms, for nouns not verbs, the question of parsing text expectationally with their aid does not arise (as has been suggested in Schonk 1975b, and argued against in Wilks 1977b), although this maybe a purely academic, question in that scripts are at present (Riesback 1977) being applied independently at parsing procedures. This leads to close analogies between the present paper and the proposals of (Granger 1977) to use knowledge-based techniques (though not actually the scripts avaible in the system he is working as part of) to understand unknown wordsin texts. The relation between unknown words and preference-violating known words is complex and beyond the scope of this discussion.

Another and very general and relevant issue is that of the relation of the "understood structure" derived for a text, and the memory structure to be associated with it. Ortony (1975) has argued persuasively that one must not just assume the two to be the same, in the face of much counter-evidence. In the present proposals, the "episode" structure for a text has the same format as a PT, or memory structure, but the two are not assumed to be the same (or even differently filled in copies) for any input.

The implementation environment, or debugging aid for NLUS's, is, like Moch AI, only hand-waving in the face of very difficult!, and ill-understood problems. It it has a main distinguishing feature, it is in the attempt to separate those projections where the highly specific PTs are helpful from those where they are not. In some cases general principles, like extraction or "pragmatic override", seem sufficient. The key interest of the proposed implementation would be the possibility of assessing the relative values of highly specific structures and general principles. This is in keeping with the "lazy system" assumption that has always been behind this NLUS** that a system should do no more processing and inference work than is necessary to deal with the analysis problem in hand, even if that problem is coping with extended usage.

It is hoped that a system able to project in this way, from both general and specific knowledge structures, and to relate such projections to the application of wholly general pragmatic principles might give insight into the complex role of knowledge in language understanding, and into the peculiar role of language boundaries that the preference restrictions symbolize.

* (Grice 1967) The principle of avoiding unnecessary redundancy: if "it=ambulance driver" the writer would have used "killed himself".

** Cf. the notion of "variable processing depth" in Bobrow & Winograd.

REFERENCES

Robrow, D & Winograd, T. KRL - an overview of a knowledge representation language. Cognitive Science U 1977

Givon, T. The structure of ellipsis. (Systems Development Corp.: Sta. Calif). 1967

Granger, R. Foul-up.ubmitted to Fifth IJCAI, 1977

Grice, H. Logice and Conversation, Unpublished mss. 1967

King, M. & Wilks, Y. Semantics, Preference and Inference. (Inst, for Semantic and Cognitive Studies: Geneva). 1977

Minsky, M. A framework for represenation knowledge in Winston (ed.) The Psychology of Computer Vision. McGraw Hill: New York). 1975, 211-277.

Ortony, A. How episodic is semantic memory? Proc. Theoret. Issues in Natural Language. Pricessing. BBN: Camb. Mass. 1975.

Riesbeck, C. (as Granger) 1977

Schank, R. (ed.) Conceptual Information Processing. (North Holland: Amsterdam). 1975a

Using knowledge to understand (as Ortony). 1975b

Wilks, Y. Computable Semantic Derivations. (as Givon). 1968

A preferential, pattern-matching, semantics for natural language understanding. Artificial Intelligence, 6, 1975.

De Minimis: the archaeology of frames. Proc. AISB Conference (Dept. of A.I., University of Edinburgh), 1976a.

Processing Case. Amer. Jn 1. Comput. Ling. 56. 1976b.

Making Preferences more active, Edinburgh Dept. of A.I., memo No.32, 1977a.

Wilks, Y. Frames, Scripts, Stories and Fantasies Pragmatics Microfiche, 1977b.

A CONCEPTUAL THEORY OF QUESTION ANSWERING

Wendy G. Lehnert
Department of Computer Science
Yale University
New Haven, Ct. 06520

ABSTRACT

A theory of Q/A has been proposed from the perspective of natural language processing that relies on ideas in conceptual information processing and theories of human memory organization. This theory of Q/A has been implemented in a computer program, QUALM. QUALM is currently used by two story understanding systems (SAM and PAM) to complete a natural language processing system that reads stories and answers questions about what was read.

Keywords: natural language processing, computational question answering, conceptual information

1_ INTRODUCTION

If a computer is going to answer questions in a manner which is natural for human interaction, the computer must have knowledge of how people ask questions and what kinds of answers are expected in return. A competent question answering system must be based on a theory of human question answering that describes:

(1) what it means to understand a question
(2) how context affects understanding
(3) what kind of responses are appropriate
(4) how to extract answers from memory

A theory of conceptual question answering has been developed which addresses these four problems [Lehnert '77]. This theory has been implemented in a computer program (QUALM) which runs in conjunction with two story understanding systems, SAM [Cullingford '77] and PAM [Wilensky '76], enabling these systems to answer questions about the stories they read.

The theory of question answering proposed by QUALM is a theory of natural language processing. This distinguishes QUALM from many other question answering systems which are oriented towards information retrieval. Many systems which attempt to answer questions phrased in natural language have been designed in two pieces: (1) a memory retrieval system, and (2) a natural language interface. Very often the interface problem is considered secondary to the retrieval system and the two subsystems are designed as if they were theoretically independent of each other [Shortliffe '74, Woods '72].

The theory behind QUALM extends theories of memory processing which originated with the study of parsing [Riesbeck & Schank '77] and generation [Goldman '75]. Conceptual Dependency [Schank '75] has proven to be a strong representational system for the task of question answering. Parsing and generation strategies based on Conceptual Dependency were naturally adopted for question answering without significant alterations. This approach to question answering which utilizes existing theories of natural language processing constitutes a major departure from the information retrieval viewpoint where natural language is considered to be merely a "front end" for a question answering system.

In order to understand questions, QUALM must interface with a conceptual analysis program that parses an English question into its Conceptual Dependency representation [Schank '75]. In SAM and PAM, QUALM interfaces with a parser designed by Christopher Riesbeck [Riesbeck & Schank '76]. In order to produce answers in English, QUALM also needs a generator that can translate Conceptual Dependency representations into English. The generator used by SAM and PAM is based on a generator designed by Neil Goldman [Goldman '75]. All of the processing specific to answering questions occurs on a conceptual level that is language independent. If QUALM interfaced with a Russian parser and a Chinese generator, it would be able to understand questions stated in Russian and produce answers to these questions in Chinese. No changes in QUALM are required to accommodate different languages since the question answering processes are independent of language.

2. CONCEPTUAL CATEGORIES FOR QUESTIONS

When QUALM initially receives a question from the parser, the question Is represented as a Conceptual Dependency conceptualization. This conceptualization must then be categorized into one of thirteen possible Conceptual Categories. The Conceptual Categories for questions are:

(1) Causal Antecedent	(7) Instrumental/Procedural
(2) Goal Orientation	(8) Concept Completion
(3) Enablement	(9) Expectational
(4) Causal Consequent	(10) Judgemental
(5) Verification	(11) Quantification
(6) Disjunctive	(12) Feature Specification
(13) Request	

The conceptual parse of a question represents a very literal or naive understanding of the question. Conceptual Categorization constitutes a higher level of interpretation which is designed to determine exactly what the questioner really means. For example, if a stranger walks up to John on the street and asks:

QI: Do you have a light?

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

John would parse this question into a conceptualization equivalent to asking:

Q2: Do you have in your immediate possession an object capable of producing a flame?

If John does not interpret the question any further than this, he could answer:

Al: Yes, I just got a new lighter yesterday.

and then walk away. This sort of response indicates that John did not have a complete understanding of the question. He understood it on a preliminary level, but he did not understand it in terms of what the questioner had intended. His misinterpretation can be explained as faulty Conceptual Categorization. What John understood to be an inquiry deserving a yes or no answer, should have been understood as a request deserving a performative action. The person asking Ql didn't just want to know if John had a light; he wanted John to offer him a light (flame). In terms of Conceptual Categories, we would say that the question should have been interpreted as a Functional Request rather than a Verification Inquiry.

If a question is not categorized correctly, it will be impossible to produce an appropriate response.

RIGHT: Q3: How could John take the exam?
(an Enablement question)
A3a: He crammed the night before,
(an Enablement answer)

WRONG: Q3: How could John take the exam?
(an Enablement question)
A3b: He took it with a pen.
(Instrumental/Procedural answer)

Q3 is asking about the enabling conditions for taking an exam. In order to take an exam, one has to be prepared for it, presumably be a student, and so forth. Q3 suggests that the questioner does not believe John satisfied some necessary enabling condition. An appropriate answer to Q3 will address this questioned enablement (He crammed the night before, or he bribed an administrator). A3b does not address the Enablement conditions at all. A3b answers the question on a much lower level of instrumentality, indicating that the question was understood to be an Instrumental/Procedural question instead of an Enablement question.

RIGHT: Q4: How did John die?
(Causal Antecedent question)
A4: He caught the swine flu.
(a Causal Antecedent answer)

WRONG: Q4: How did John die?
(Causal Antecedent question)
A4b: Well, first he was alive,
(an Enablement answer)

This time A4b indicates that Q4 was understood to be an Enablement question. A necessary enablement for dying is being alive. But Q4 should not have been interpreted to be asking about the enabling conditions for dying. QA is more reasonably understood to be asking about the cause of John's death: Was it an accident? Was he ill? Did he kill himself?

RIGHT: Q5: How did John get to Spain?
(Instrumental/Procedural question)
A5a: He went by plane.
(Instrumental/Procedural answer)

WRONG: Q5: How did John get to Spain?
(Instrumental/Procedural question)
A5b: He wanted to see Madrid.
(a Causal Antecedent Answer)

An appropriate answer to Q5 would specify the transportational means which was instrumental to John's getting to Spain (he took a cruise, he flew, etc.) But A5b tells us what caused John to go to Spain. A5b answers a Causal Antecedent question instead of an Instrumental/Procedural question.

When Q3-5 are represented in Conceptual Dependency, it is easy to see which Conceptual Category should be assigned to these questions. In QUALM, parsed conceptualizations are run through a discrimination net which assigns a Conceptual Category to each question. But Conceptual Categorization does not constitute complete understanding of a question. Each conceptual question is subject to further interpretive processing before a memory search for an answer can begin.

3. INFERENTIAL ANALYSIS

Complete understanding of a question often involves inferences in addition to Conceptual Categorization. When interpretation of a question does not include analysis by inference, answers may be produced which are technically correct, but completely useless. Suppose John is mixing cake batter and he asks his wife:

Q6: Now what haven't I added?
A6: A pound of dog hair and an air filter.

She's probably right; he probably didn't add a pound of dog hair and an oil filter. But her answer is inappropriate because John was "obviously" asking for what he hadn't added that he should have added. The intent of this question is obvious only when an interpretive inference mechanism can be invoked to supply an implicit constraint. There is an entire class of questions that require the same type of inferential analysis:

Q7: Who isn't here?
(Who isn't here who should be here?)
Q8: What did I forget to buy?
(What didn't I buy that 1 should have?)

In each of these questions, an inference must be made that specifies appropriate constraints for potential answers. When Q7 is asked by a professor upon entering his class, appropriate answers refer to members of the class. When Q8 is asked in the context of shopping for a dinner party, appropriate answers refer to those things that are needed for dinner.

The Universal Set Inference, a general inference mechanism, is needed for questions of this class. This mechanism examines the context of a question and determines appropriate constraining factors. But before this mechanism can be invoked, some process must be responsible for recognizing which questions require this particular inference mechanism. The Universal Set Inference should not be summoned for questions like:

Q9: Who is coming to your party?
Q10: Isn't this the book you wanted?

The successful application of an interpretive inference mechanism relies on the ability to know when that mechanism is needed. This is one way Conceptual Categorization is exploited. One of the thirteen Conceptual Categories is the class of Concept Completion questions. These correspond roughly to fill-in-the-blank questions. During the interpretation of a question, the Universal Set Inference is applied if and only if:

(1) the question is categorized as a Concept Completion question, and
(2) the conceptual question has MODE - NEG

Q6-8 each satisfy these requirements. While the lexical statement of Q8 does not appear to be negated, the conceptual representation for Q8 is equivalent to asking "What didn't I remember to buy?" which is encoded as an MTRANS with negative MODE. Q9 is a Concept Completion question but it fails to meet the criteria because it has a non-negative MODE. Q10 fails because it is a Verification question instead of a Concept Completion question.

A useful system of categorization will provide simple test criteria for inference mechanisms of the sort just described. Different questions require different processing. A strong categorization system can recognize which processes are required for a given question and dictate subsequent processing accordingly.

±1. CONTEXT-SENSITIVE INTERPRETATION

In the last section we claimed that one general inference mechanism, the Universal Set Inference, could be invoked to establish appropriate constraints on Concept Completion questions with MODE - NEG. This inference mechanism relies on the context in which questions are asked.

Q7: Who isn't here?

requires contextual information that implicitly specifies who should be here. If this question is asked by a professor in a class, it means "Which of my students aren't here?" If it is asked by a host at a party, it means "Who Isn't here who was invited?" Without contextual information, it is impossible to know what implicit constraints are appropriate.

Specific constraints on questions can be derived from whatever scripts [Schank & Abelson '77] are actively operating in a given context. When a script is active, its script-defined roles and role instantiations [Cullingford '77] delineate the universal set these questions implicitly reference.

The Universal Set Inference

Question Category: Concept Completion
Question Criteria: MODE value - NEG
Contextual Criteria: there is an active script

If these test criteria are satisfied, interpretive constraints are imposed by those roles defined in the active script(s).

question	scriptal context	constraining roles
Who isn't here?	party	guests
What didn't I add?	cooking	ingredients
Who hasn't bid?	bridge game	bridge players

Many questions must be understood in terms of their surrounding context. It is therefore crucial to be able to characterize contextual information in terms of general knowledge structures (Schank & Abelson '77) so that general interpretive inference mechanisms can be designed which are sensitive to context without being context-specific. That is, a contextually sensitive processing mechanism should be applicable in different contexts. A theory of question answering that needs to propose a new set of processing strategies for each new context encountered is not much of a theory.

5. CONTENT SPECIFICATION

Once a question has been sufficiently understood, retrieval processes can begin to look for an answer. The first part of the retrieval process decides how much of an answer is needed. Consider the following story:

John went to a restaurant and the hostess gave him a menu. When he ordered a hot dog the waitress said they didn't have any. So John ordered a hamburger instead. But when the hamburger came, it was so burnt that John left.

If asked:

QI4: Did John eat a hot dog?

There are many possible answers. When SAM reads this story, SAM can answer Q14 three different ways:

A14a: No.
A14b: No, the waitress told John they didn't have any hot dogs.
AI4c: No, the waitress told John they didn't have any hot dogs and so John ordered a hamburger.

These answers are all different in terms of the amount of information they convey. in fact, answers can vary not only in terms of their relative content, but in terms of the kind of content they communicate. For example, if Q14 had been answered "Yes," in the context of our story where John didn't eat a hot dog, then the content of this answer would be described as wrong.

The decision-making processes that determine what kind of an answer should be returned are part of Content Specification. Content Specification takes into account the Conceptual Category of each question and intentionality factors that describe the "attitudinal" mode of the entire system in order to determine how a question should be answered. A system of descriptive instructions are produced by Content Specification to instruct and guide memory retrieval processes as they look for an answer.

The primary challenge involved in Content Specification is precisely how these instructions to memory retrieval are formalized. It is not enough to say "give a minimally correct answer," or "bring in everything you can find that's relevant." The instructions generated by Content Specification must tell the retrieval heuristics exactly how to produce a minimally correct answer and exactly what has to be done to do to come up with everything that's relevant.

One type of Content Specification mechanism that guides retrieval heuristics are Elaboration Options. Each Elaboration Option has four parts: an intentionality Threshold, a Question Criterion, an Initial Answer Criterion, and Elaboration Instructions. Intentionality refers to variables within the system that are set with suggestively-named values like "talkative," "cooperative," "minimally responsive," etc. The intentionality Threshold specifies what sort of Intentionality must be assigned to the system in order for an Elaboration Option to be attached to the question. The Question Criterion describes what Conceptual Category must be assigned to the question in order for it to receive the Elaboration Option. If either the Intentionality of the system or the Conceptual Category of the question fail to meet the specifications of the Intentionality Threshold and the Question Criterion, then the Elaboration Option is not used. The Answer Criterion specifies the type of conceptual answer which the memory search must initially return in order for the Elaboration Option to be executed. And the Elaboration Instructions specify exactly how an elaboration is to be extracted from memory and integrated into the conceptual answer.

To see exactly how an Elaboration Option works, we will discuss one of the simpler Elaboration Options that has been implemented in QUALM. Consider the following story:

> John went to New York by bus. On the bus he talked to an old lady. When he left the bus, he thanked the driver. He took the subway to Leone's. On the subway his pocket was picked. He got off the train and entered Leone's. He had some lasagna. When the check came, he discovered he couldn't pay. The management told him he would have to wash dishes. When he left, he caught a bus to New Haven.

After reading this story SAM answers:

Q15: Did John go to New York?
A15: Yes, John went to New York by bus.
Q16: Did John eat?
Alb: Yes, John ate lasagna.
QI7: Did someone pick John's pocket?
A17: Yes, a thief picked John's pocket.
Q18: Did John pay the check?
A18: Yes, John paid the bill.

These are answers SAM gives when it has a talkative Intentionality. If SAM were running with a less than talkative Intentionality, each of these questions would have been answered with a simple "Yes." The longer answers (A15-18) are the result of the Verification Option. This is a very simple Elaboration Option which is defined as follows:

<u>The Verification Option</u>

Intentionality Threshold: Talkative
Question Category: Verification
Answer Criterion: initial answer is Yes
Elaboration Instructions:

> final conceptual answer is "Yes, *X*" where *X* is the conceptualization found in the story representation that matches the question concept.

The retrieval heuristics for a Verification question search the story representation for a conceptualization matching the conceptual question. If it finds a match, the initial answer is Yes. A conceptualization from the story representation doesn't have to correspond to the question concept exactly in order to match it; it may contain more information than the question concept. This is why A15-17 appear to volunteer information. A15 tells how John went to New York, A16 says what John ate, and A17 asserts who stole John's wallet.

6. FINDING AN ANSWER

Retrieval heuristics vary for each Conceptual Category of questions. A number of interesting problems arise in designing processes that extract information from memory. We will briefly outline three such problems which reflect the scope and depth of the difficulties involved.

6.1 Integrative Memory Processing

Expectational questions are interesting because they cannot be answered on the basis of a story representation alone. Expectational questions correspond roughly to why-not questions. These questions require "integrative" memory processing. The term integration is very often used in the context of adding new information to memory. A single unit of information is "integrated" into a larger memory structure. But in the context of retrieving information from memory, an integrative process is one which combines information from different sources to produce new information.

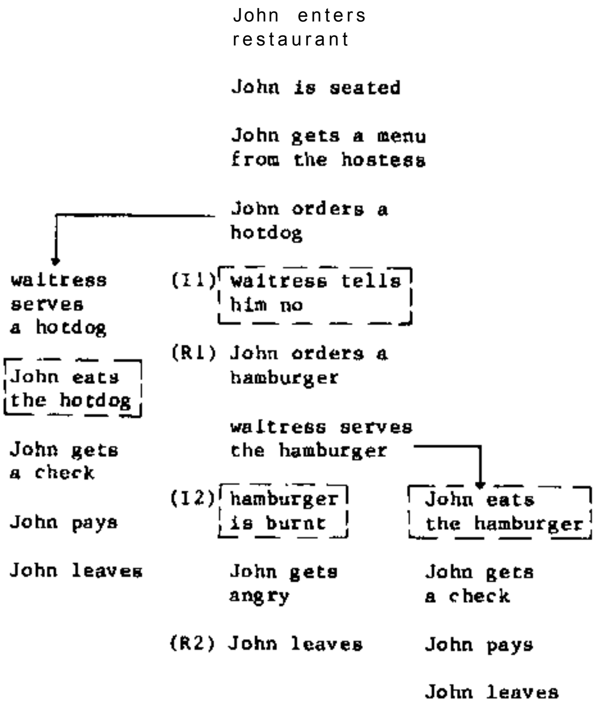
After reading the burnt-hamburger story, SAM answers:

Q19: Why didn't John eat a hot dog?
A19: Because the waitress told John they didn't have any hot dogs.
Q20: Why didn't John eat the hamburger?
A20: Because the hamburger was burnt.

These questions are answered by an integrative process that combines the story representation with predictive mechanisms in order to reconstruct expectations that were alive at some time during the understanding process. When John ordered a hot dog we had an expectation that he would eat a hot dog until we heard there were none. When John ordered a hamburger we expected him to eat a hamburger until we heard that the hamburger was burnt and John just left. Expectational questions ask about expectations which were aroused at some point during the understanding process and then subsequently violated by an unexpected turn of events. Had we asked "Why didn't John swim across the lake?" the question would seem unreasonable since we never had any expectations about John going swimming or crossing a lake.

The theories of memory representation implemented in SAM and PAM adhere to the premise that a story representation should encode information about things that happened in the story. This includes inferences about things that probably occurred (but weren't explicitly mentioned) as well as conceptualizations for events that were explicitly described in the input story. But Expectational questions ask about things that didn't happen. To answer an Expectational question, we must use the same predictive processes used during story understanding to reconstruct failed expectations which were alive at some time during understanding. The reconstruction of failed expectations is achieved by an integrative memory process called ghost path generation.

The generation of ghost paths cannot be fully understood without a fundamental understanding of script application [see Cullingford '77]. But some sense of what goes on should be apparent from the following diagram. In this diagram, the chain of events in the center corresponds very roughly to information in the story representation that SAM generated at the time it read the burnt-hamburger story. The two chains on either side correspond to the two ghost paths needed to answer Qi9 and Q20.



6.2 Answer Selection

While Expectational questions are interesting because they cannot be answered on the basis of a story representation alone, there are many questions that do not need information outside of the story representation which are still difficult to answer. Causal Antecedent questions are complicated in this respect. A Causal Antecedent question is one which asks for the reason behind an event. After reading the Leone's story, consider the following answers;

Q21: Why did John wash dishes?
A21a: Because he couldn't pay the check.
A21b: Because he had no money.
A21c: Because he had been pickpocketed on the subway.

SAM answers Q21 "Because he had no money." But is this the best answer of the three? What factors determine the superiority of one answer over another?

Effective answer selection entails making assumptions about what the questioner knows. Anyone who asks Q21 can be assumed to know that John washed dishes. If we go on to assume that the questioner knows two more things: (1) John washed dishes in a restaurant, and (2) washing dishes in a restaurant is classically what happens when someone eats and then can't pay, then the questioner can infer: (3) John couldn't pay the check. If the questioner can figure out for himself that John couldn't pay the check then A21a does not tell him anything he doesn't know to begin with. A good answer must take into account what the questioner does and doesn't know, and address the knowledge state of the questioner by telling him something new.

A21b is a weak answer for the same reasons that A2la was weak. If someone knows that John couldn't pay a check, they can reasonably infer that John didn't have (enough) money. Both inferences:

1) John couldn't pay the check.
2) John didn't have any money.

can be made by the questioner on the basis of general world knowledge and knowing that John washed dishes in a restaurant. But there is no way the questioner can infer that John was pick-pocketed on the subway without additional knowledge of the story. Therefore A21c is the best answer to Q21 as long as we assume the questioner has knowledge about the world and can make inferences on the basis of that knowledge.

If we assumed that the questioner knew nothing about restaurants, A21a would be the best answer. If we assumed that the questioner knew about restaurants but didn't understand about paying for things, A21b would be the best answer. It is impossible to judge various answers to a question without knowing (or assuming) something about the person being addressed.

<u>6.3 Conceptual Organization</u> of <u>Knowledge</u>

When people answer questions, their answers sometimes tell us something about the form and organization of conceptual information in human memory. For example, consider the following story:

> John was sitting in a dining car. When the train jerked, the soup spilled.

Suppose we ask:

Q22: Where was the soup?

This is a specification question that can be answered a number of ways. Two common answers are:

A22a: In a bowl.
A22b: On the table.

A much less natural answer would be:

A22c: On a plate.

A22c seems to be very odd answer which conjures up an image of a soup puddle on a plate. This is not the scene most people envision when hearing the story. Most people imagine the soup in a bowl on a plate on a table.

The acceptable and unacceptable answers to Q22 tell us something about human memory organization. It never occurs to people to answer "On a plate." Furthermore, when this answer is given it provokes a wrong image of soup resting directly on a plate. But "On the table," is a natural answer. Why is it that "On a plate," is a bad answer but "On the table," is perfectly reasonable? The soup does not rest directly on the table any more than it rests directly on a plate. Why is it acceptable in one case but not the other? This phenomenon must be accounted for in terms of memory organization.

When people hear this story they assume a causality between the train jerking and the soup spilling. (If asked "Why did the soup spill?" people will answer "Because the train moved.") This causality relies on the fact that the soup is physically connected to the train in some way. This physical connection can only be recognized by constructing a path of physical objects between the soup and the train. This path of connections must be accessed In order to answer Q22. If a path is constructed the same way people build one, it will be easy to retrieve answers to Q22 that seem natural. If the path is built differently, we may end up with an answer like A22c.

Suppose we construct a path like the following:

```
A BAD PATH:   soup (inside-of)
              bowl (on-top-of)
              plate (on-top-of)
              tablecloth (on-top-of)
              table (on-top-of)
              floor (inside-of)
              dining car (part-of)
              train
```

With this memory representation it is not clear how we can extract the answers A22a and A22b without also getting answers like "On a plate," or "On a tablecloth." There is nothing in this memory representation that tells us where the good answers are. What we need is a memory representation that makes it easy to find a bowl and a table but hard to retrieve a plate.

```
A BETTER PATH:  soup (inside-of)
                bowl (part-of)
                placesetting (part-of)
                tablesetting (on-top-of)
                table (part-of)
                dining area (part-of)
                dining car (part-of)
                train
```

This path suggests a very simple retrieval heuristic to produce the answers A22a and A22b: trace the path looking for objects which are connected by either "inside-of" or "on-top-of" links.

The closer a memory representation is to human memory organization, the easier it will be to produce answers that make sense to people. A system of memory representation for physical objects has been proposed [Lehnert '77] which is designed to facilitate inference and retrieval problems of the sort just described. Conceptual descriptions of objects in this system are based on decompositions into a set of seven object primitives in much the same way that Conceptual Dependency [Schank '75] describes actions by decomposing them into a set of primitive acts.

7. CONCLUSIONS

The overall question answering process can be intuitively approached in two stages: understanding the question (interpretation) and finding an answer (memory retrieval). Each of these stages is likewise divided into two parts:

INTERPRETATION:
- [1] Conceptual Categorization
- [2] Inferential Analysis

MEMORY RETRIEVAL:
- [3] Content Specification
- [4] Searching Heuristics

[1] Conceptual Categorization guides the subsequent processing by dictating which specific inference mechanisms, elaboration options, and retrieval heuristics should be invoked in the course of answering a question.

[2] Inferential Analysis is responsible for understanding what the questioner really meant when a question should not be taken literally.

[3] Content Specification determines how much of an answer should be returned in terms of detail and elaborations.

[4] Searching Heuristics do the actual digging in order to extract an answer from memory.

All of the processes within these four phases are specific to question answering per se and are language-independent, operating within a conceptual representation system.

QUALM represents a theory of question answering which is motivated by theories of natural language processing. Within the context of story understanding, QUALM has provided a concrete criterion for judging the strengths and weaknesses of story representations generated by SAM and PAM. If a system understands a story, it should be able to answer questions about that story in the same way that people do. Although the computer implementation of QUALM is currently limited to the application of answering questions about stories, the theoretical model [Lehnert '77] goes beyond this particular context. As a theoretical model QUALM is intended to describe general question answering, where question answering in its most general form Is viewed as a verbal communication device between people.

While many of QUALM's question answering techniques are designed for answering questions about stories, QUALM is not limited to stories about a specific content domain. QUALM is applicable to any story that can be understood in terms of scripts and plans [Schank & Abelson '77]. This limitation is not content-specific; it is dependent on the general knowledge structures that are used in text understanding. When new scripts and plans are added to the knowledge base for SAM and PAM, questions can be answered about stories using this new knowledge without any additional alterations to QUALM.

BIBLIOGRAPHY

Cullingford, R. E. (1977) Organizing World Knowledge for Story Understanding by Computer, (thesis) Department of Engineering and Applied Science. Yale University, New Haven, Ct.

Goldman, N. (1975). Conceptual generation. In R. C. Schank, ed: Conceptual Information Processing. North Holland Amsterdam.

Lehnert, W. (1977). The Process of Question Answering. (thesis) Research Report //88. Department of Computer Science, Yale University, New Haven, Ct.

Riesbeck, C. and Schank, R. (1976). Comprehension by Computer: Expectation-Based Analysis of Sentences in Context. Research Report //78. Department of Computer Science, Yale University, New Haven, Ct.

Schank, R. C. (1975). Conceptual Information Processing. North Holland, Amsterdam.

Schank, R. C. and Abelson, R. P. (1977). Scripts, Plans, Goals and Understanding. Lawrence Erlbaum Associates. Hillsdale, N.J.

Shortliffe, E.H. (1974). MYCIN: A Rule-Based Computer Program for Advising Physicians Regarding Anti-Microbial Therapy Selection. (thesis) Stanford Artificial Intelligence Laboratory Memo-AIM25i. Stanford University, Stanford, CA.

Wilensky, R. (1976). Using Plans to Understand Natural Language. Proceedings of the Annual Conference of the Association for Computing Machinery. Houston, Texas.

Woods, W.A., Kaplan, R.M., Nash-Webber, B. (1972). The Lunar Sciences Natural Language Information System: Final Report. BBN Report No. 2378, Bolt, Beraneck and Newman, Inc. Cambridge, Mass.