AN ADAPTIVE NATURAL LANGUAGE SYSTEM THAT LISTENS, ASKS, AND LEARNS

Parry L. Miliar
Massachusetts Instituta of Technology
Cambridge, Hassachueetts

Abstract

Whan a usar intaracts with a natural language system, ha may wall uaa words and expressions which ware not anticipated by tha systea designers. This paper describes a systea which can play TIC-TAC-TOE, and discuss tha game while it Is in progress. If the systea encounters new words, new expressions, or inadvertent ungrammatlcalltles, It attempts to understand what was meant, through contextual inference, and by asking intelligent clarifying questions of the user. The systea than records the meaning of any new words or expressions, thus augmenting its linguistic knowledge in the course of usar interaction.

1. INTRODUCTION

A number of systems are being developed which communicate with users in a natural language such as English. Tha ultimate purpose of such systems is to provide easy computer access to a technically unsophisticated person. When such a parson interacts with a natural language systea, however, he is quite likely to use words and expressions which were not anticipated. To provide truly natural interaction, the system should be able to respond intelligently when this happens.

Host current systems, such as those of Winograd [10] and Woods [11], are not designed to cope with such "linguistic input uncertainty." Their parsers fail completely if an input sentence does not use a specific, built-in syntax and vocabulary. At the other extreme, systems like ELIZA [9] and PARRY [2] allow tha user to type anything, but make no attempt to fully understand the sentence. The present work explores the middle ground between these extremes: developing a system which has a great deal of knowledge about a particular subject area, and which can use this knowledge to make language interaction a flexible, adaptive, learning medium.

In pursuing this goal, the present work is most closely related to work being done in the various speech recognition efforts [9, 7, 8, 12] which are studying how linguistic and semantic constraints can help deal with the ACOUSTIC error and uncertainty of speech. The adaptive systea, however. Is designed to deal with a much more LINGUISTIC type of uncertainty.

When people use unfamiliar words or expressions in conversation, we can usually deduce from context what is meant, and if not, we can at least ask intelligent clarifying questions. To allow the machine to do tha same, there must ba a very flexible interaction of syntax and semantics in the parsing/understanding process. There must be a different parser organization, and a more flexible use of linguistic and semantic constraints, than la present in current natural language systems.

The adaptive systea is a step towarda thls goal. The current implementation la a prototype, designed to illustrate many of these ideas, and to tie them together in a restricted system that is complete but not too complex. The system's domain of discourse is TIC-TAC-TOE: it plays a game and discusses the game In progress. It has a (currently lialted) set of semantic concepts, and attempts to interpret the user's sentences in terms of these. The set of semantic concepts is FIXED. The systea does not attempt to learn new concepts, merely new ways of describing concepts that are already familiar.

When the system encounters a new word or a new phrasing, it uses contextual information to determine tha possible meaning, and to ask clarifying questions which are as intelligent as possible. It then remembers the meaning of the new expression, and will use it Itself when it talks.

2. SAMPLE SESSION

The best way to introduce the system is to show it in operation. In the sample session that follows, user input is preceded by "U:", machine output by "M:", and comments added by the author are in small type, enclosed in brackets.

```
(MACHINE: I PLACE AN X IN THE TOP LEFT SQUARE)

(X - -)
(- - -)
(- - -)
```

[The machine makes its opening move.]

```
(USER: PLAY A ZERO IN THE CENTER SQUARE)

(M: IS ZERO A NOUN MEANING O)

(U: YES)
(M: OK)

(X - -)
(- O -)
(- - -)
```

[The new word "zero" is added to the machine's vocabulary.]

```
(M: I PLACE AN X IN THE BOTTOM RIGHT SQUARE)

(X - -)
(- O -)
(- - X)
```

(U: SLAP AN O IN THE BOTTOM LEFT CORNER)

(M: IS (TO SLAP SOMETHING SOMEWHERE) A WAY OF SAYING (TO PLACE SOMETHING SOMEWHERE))

(U: YES)
(M: OK)

(X - -)
(- O -)
(O - X)

[A naw surface frame (as described in taction 4) la added to the sat of surfaca frames which axpraas tha concapt of making a move.]

(M: I SLAP AN X IN THE TOP RIGHT SQUARE)

(X - X)
(- O -)
(O - X)

[In this sample session, the machine uses its newly acquired vocabulary at every opportunity.]

(U: HOW COME YOU PLACED AN X IN THE TOP RIGHT SQUARE)

(M: IS (HOW COME SOMEONE DOES SOMETHING) AN EXPRESSION MEANING (WHY DOES SOMEONE DO SOMETHING))

(U: YES)
(M: OK)

(M: TO BEAT YOU)

[A new surface frame is added to those expressing the concept 'WHY'. Syntactic features are associated with each such frame indicating such things as whether subject-verb inversion is to be performed.]

(U: CAN YOU CURRENTLY BEAT ME)

(M: IS CURRENTLY AN ADVERB REFERRING TO THE TIME OF A MOVE)

[These are the only adverbs the system knows about. In fact, at present, the system ignores adverbs in its semantic processing. It asks the question because it wants to account for all the words in the sentence.]

(U: YES)
(M: OK)

(M: YES)

(U: AFTER YOU PLAYED AN X IN THE TOP WEST SQUARE COULD I BEAT YOU IF I PLAYED AN O IN THE BOTTOM RIGHT SQUARE)

(M: IS WEST AN ADJECTIVE MEANING)
EITHER
(1: CORNER)
(2: LEFT)

(U: 2)
(M: OK)

(M: NO)

[Notice that two top squaras contain an x. However, only tha left ona was playad while tha bottom right squara was fraa. Tharafora tha systaa concluded that tha top laft squara was baing dascrlbad. This example illustrates how contextual information can ba usad to ask as intelligent a clarifying question as possible.]

(U: HOW COME YOU PLACED AN X IN THE TOP EAST SQUARE)

(M: IS EAST AN ADJECTIVE MEANING)
EITHER
(1: RIGHT)
(2: WEST)

(U: 1)
(M: OK)

(M: TO BEAT YOU)

[Here, either top square might have been referred to.]

(U: WHEN I PLAYED AN O IN THE BOTTOM LEFT SQUARE COULD YOU BEAT ME)

(M: IS (WHEN SOMEONE DOES SOMETHING) AN EXPRESSION MEANING (AFTER SOMEONE DOES SOMETHING))

(U: YES)
(M: OK)

(M: YES)

[In the domain of this TIC-TAC-TOE discussion, "when" and "after" have the same meaning.]

[The next two sentences each contain two problems for the system to resolve.]

(U: WHEN YOU PLAYED AN X IN THE TOP RIGHT CELL COULD I KEEP YOU FROM BEATING ME)

(M: IS CELL A NOUN MEANING SQUARE)

(U: YES)
(M: OK)

(M: IS (TO KEEP SOMEONE FROM DOING SOMETHING) A WAY OF SAYING)
EITHER
(1: (TO PREVENT SOMEONE FROM DOING SOMETHING))
(2: (TO ALLOW SOMEONE TO DO SOMETHING))
(3: (TO FORCE SOMEONE TO DO SOMETHING))

(U: 1)
(M: OK)

(M: NO)

(U: <u>IF I LAY AN O ON THE RIGHT SIDE SQUARE HOW CAN YOU POLISH ME OFF</u>)

(M: IS (TO LAY SOMETHING SOMEWHERE) A WAY OF SAYING (TO SLAP SOMETHING SOMEWHERE))

(U: YES)
(M: OK)

(M: IS (TO POLISH SOMEONE OFF) A WAY OF SAYING) EITHER
(1: (TO BEAT SOMEONE))
(2: (TO LOSE TO SOMEONE))
(3: (TO TIE SOMEONE))

(U: 1)
(M: OK)

(M: BY LAYING AN X ON THE TOP CENTER CELL)

[If this same session were subsequently repeated, of course, the system would not need to ask any questions of the user.]

2.1 Comments on Currant Limitations

Thara ara a number of limitations to tha adaptivs systaa as it now stands. Some of these may ba apparant in tha sample session, but an introduction to tha *system* is not coaplata without discussing them explicitly.

(1) Tha number of concapts available to tha systaa at prasant is vary small. This, in fact. Is why tha *system's* first guess is usually tha corract one. If tha sentence is at all within tha systaa's comprehension, the options as to Its aaaning ara currently quite Halted.

(2) Tha range of expressive devices presently recognized is quite lialted as wall. For instance, tha system does not recognize relative clauses, conjunctions, or pronouns (except for I and you).

(3) The system currently deals only with TOTALLY UNFAMILIAR words and expressions in this adaptive fashion. It will not correctly handle familiar words which are used in new ways (such as a noun used as a verb, as in "zero the center squara").

(4) The system tries to map the meaning of new words and expressions into its specified set of underlying concepts. It then displays its hypotheses to the user, giving him only the option of saying yes or no. The user cannot say "no, not quite, it means ...". (Thus concepts like "the 'northeast' square" or "the 'topmost' square" would be confusing and not correctly understood.)

The present simple systaa has been developed with two goals In mind: (1) to explore the tachniques required to achieve adaptive behavior, and (2) to help formulate the issues which will have to be faced when incorporating these tachnlques into a much broader natural language system.

3. OVERIEW

Fig. 1 shows the various stages that the Adaptive Systam goes through in understanding a sentenca. In this section, we shall watch while tha systaa procassas tha santanca "How Come you placed an x in tha top right square."

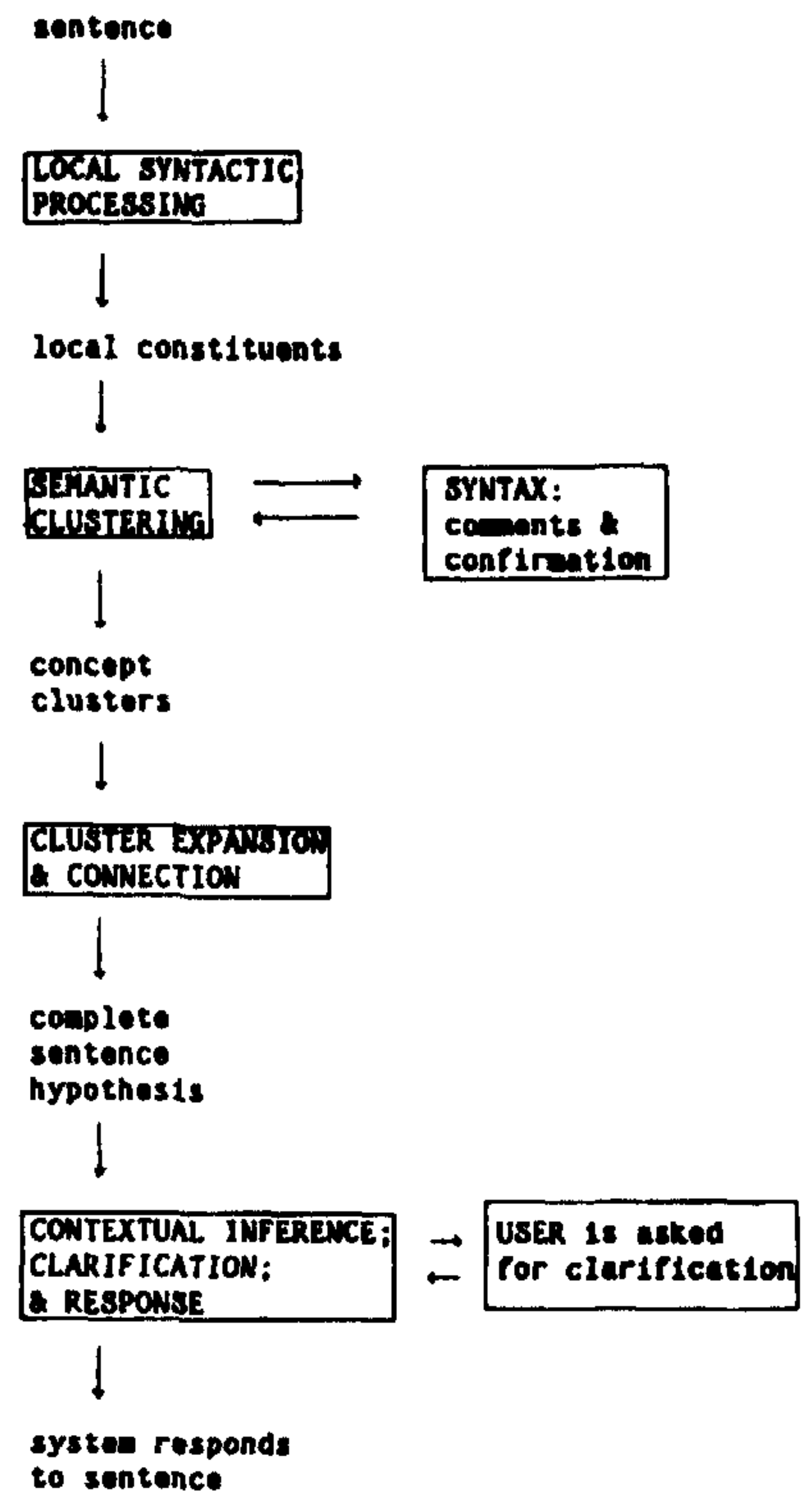

Fig. 1: Adaptive System Overview

(1) Local Syntactic Processing:
In this first stage, the system scans the antlra sentence looking for local constituents. These include "simple" noun phrases (NPs) and prepositional phrases *(pps),* ("simple* meaning "up to the head noun but not including any modifying clauses or phrases"), and verb groups (VGs) consisting of verbs together with any adjoining models, auxIIIlaries, and adverbs. In this instance, the system finds the two NPs, "you" and "an x", the PP "in the top right squara", and tha VG "placed".

(2) Semantlc Clustering:
At this stage, tha clause-laval processing starts.

Unlike most systems, this clause-level processing is driven by SEMANTIC relationships, rather then by syntactic form. It uses a semantics-first "clustering", with a secondary use of syntax for comments and confirmation. In this example, all the local constituents found can be clustered into a description of a single concept: that of maklng a move. Section 4 describes the mechanics of this stage in more detail.

(3) Cluster Expansion and Connection:
During this stage, an attempt is made to account for each word in the sentence by expanding the concept clusters, and if there is more than one, by joining them together to form an entire multlclausal sentence. In this case, the concept cluster might be expanded in two ways.
a) One possibility might be that it is a "HOW" type question, and that "come" is some sort of adverb. However this possibility violates a semantic constraint, since the system is not set up to answer how a move is made; only how to win, how to prevent someone from winning, etc. Therefore this possibility is ignored.
b) The other possibility is that "how come" is a new way of describing some other clause function.

(4) Contextual Inference; Clarification; and Response:
During this final stage, any contextual information available Is brought to bear on areas of uncertainty, any necessary clarifying questions are asked, and the system responds to the sentence. In this example, the only uncertainty is the meaning of "how come". Since this is the main clause of the sentence, the possibility of its being an "If" or "after" clause are discarded. The remaining possibilities are "imperative", "how", "why", and "can". The system does not answer "how" and "can" questions In relation to making moves. Similarly, "imperative" does not make sense since the action described is a previously made move. Therefore the system asks If "How come someone does something" means "Why does someone do something". The user answers "yes", so the system stores this new way of asking "why", and proceeds to answer the question.

4. SEMANTICS-FIRST CLAUSE-LEVEL PROCESSING

One of the major differences between this approach to parsing and that of a top-down, syntax-driven system (such as Woods* or Winograd's) is the order in which syntactic and semantic processing is done at the clause level.

In a top-down system, a sentence must exactly match the built-in syntax before semantics can even be called and given the various constituents of a clause. This is clearly undesirable when one is dealing with input uncertainty, since one cannot be sure exactly how the user will phrase his sentence. One would prefer to let semantics operate first on any local consituents present, so that it can make a reasonable guess as to what is being discussed.

As semantically-related clusters of local constituents are found, syntax can be consulted and asked to comment on the relative grammatlcallty of the various clusters. If there are two competing semantic interpretations of one part of a sentence, and syntax likes one much better than the other, then the "syntactically pleasing" interpretation can be pursued first. Later, if this does not pan out, the syntactically Irregular possibility can be looked at as well. In this way, syntax can help guide the system, but is not placed in a totally controlling position.

A by-product advantage of this semantics-first approach is that the system can handle mildly ungrammatlcal input without any extra work. In addition, the semantics-first clustering approach lends itself quite naturally to handling sentence fragments.

In the remainder of this section, we describe how the adaptive system organizes its linguistic knowledge to implement this semantics-first approach. As we shall see, there are three components of this knowledge.

(a) The local recognizers which initially find local constituents. These recognizers are represented in Augmented Transition Network [11] form, are quite simple, and are not described further in this paper.
(b) Clause-level knowledge of how actions and clause-functions are described. This knowledge is expressed In a descriptive fashion which makes It easily manlpulable, and easy to add to.
(c) Clause-level syntactic knowledge which is expressed in a domain-Independent form.

4.1 Knowledge of how Actions are Described

Figure 2 Illustrates how the system stores its knowledge of how actions (or events) are described. This knowledge Is stored at two levels: the conceptual level, and the surface (or expressive) level.

```
CONCEPT: #PLACE

CONCEPTUAL SLOTS:

        P: player
        M: mark
        S: square

SURFACE FRAMES:

        VERB: place    (as in:
        AGENT: P       I place an x in the center)
        OBJ: M
        in: S

        VERB: play     (as in:
        AGENT: P       I play an x in the center)
        OBJ: M
        in: S

        VERB: play     (as in:
        AGENT: P       I play the center)
        OBJ: S
```

Fig. 2: Linguistic Knowledge about Actions

As shown in Fig. Z, the concept #PLACE represents the act of making a TIC-TAC-TOE move.
(a) On the CONCEPTUAL level, there are three "conceptual slots" indicating the actors which are involved in the action: a player, a mark, and a square.
(b) On the SURFACE, or expressive, level there is a list of surface frames each indicating one possible way that the concept can be expressed. Each surface frame consists of a verb plus a set of syntactic

case frames to be filled by the actors. (Notice that neither the conceptual slots nor the surface fraaes indicate explicitly the order In which the various constituents are to appear In a sentence.)

When the systea processes a sentence, it fills the conceptual slots with local constituents found In the sentence. If It has found a faaillar verb, then it also gets any surface fraae(s) associated with that verb. At this point It calls syntax, asking for consents.

For Instance, if the input sentence is *I place an x in the corner", then all the conceptual slots of #PLACE would be filled, and the systea would pass the following string to syntax "agent verb obj pp". As a result, clause-level syntax does not see the actual constituents of the sentence, only the labels specified in the surface case fraae, plus Information indicating number, tense, etc.

An interesting aspect of this approach is that the clause-level syntax is entirely domain-Independent. It knows nothing about TIC-TAC-TOE, or even about the words used to talk about TIC-TAC-TOE. The surface frames allow semantics to talk to syntax purely in terms of syntactic labels. As a result, one could write a single syntactic module, and then insert it unchanged into many domains.

4.1.1 Using this Information

In this section, we describe in more detail how this knowledge can be used when processing a sentence.

(1) If the verb and constituents are faaillar:

If there is no uncertainty in a clause, then each constituent can be put into one of the conceptual slots, and any surface frames associated with the verb can be examined. The fraae indicates the case (agent, object, etc.) associated with each constituent when that verb is used. The fraae is used to create a string of case labels that are sent to syntax for comments.

For instance, if the sentence is "I place an x in the center square", the string passed to syntax is "agent verb obj pp". Syntax replies that the sentence follows normal order. Had the string been "verb obj pp", syntax would reply that the subject had been deleted. If the string was "do agent verb obj pp", syntax would reply that subject-verb inversion had taken place. Given "agent obj verb pp", syntax would reply that the object was out of position.

Thus syntax is set up to notice both grammatical and ungraomatical permutations in constituent order, and to comment appropriately. The system must then decide how to Interpret these comments.

For instance, if syntax replies that the object is out of position in the clause, or that there is incorrect agreement in number between subject and verb, the system may decide that the user has made a minor grammatical error, and allow the sentence to be processed anyway, especially if there is no better interpretation of the sentence. In this way, clause-level syntax plays an assisting role rather than a controlling role in the analysis of a sentence.

(2) If a constituent is unknown:

If an unknown constituent is present, then both the fraae and slot information can be used to halp resolve its meaning. For instance, suppose tha sentence is "I place a cross in the centar square*, and the word "cross* is unfamiliar.

Here, during tha semantic clustering, tha conceptual slots for a playar and a square can ba filled by "I" and "in tha cantar squara". but tha slot for a mark is unfilled. In addition, there is the unknown constituent "a cross".

A natural hypothesis, therefore, is that tha unknown constituent refers to a type of aark. Sinca the verb is familiar, a surface frame is available. Next, assuming the unknown constituent is a mark, the string "agent verb obj pp" can be passed to syntax. When syntax approves, this offers additional confirmation that the hypothesis Is probably right.

Subsequent evaluation of this hypothesis Indicatas that the sentance makes sanse only If tha mark referred to is an x, so tha system asks if "cross" is a noun meaning "x".

(3) If the verb is unknown:

If an unfamiliar verb is used, then there It no surface frame available to help guide tha analysis. Instead, syntax must ba usad in a different mode to propose what the surface frame should be.

Suppose the sentence is "I plunk an x in tha center square". Here, all the constituents can ba clustered into the concept 'PLACE, but-there is an unknown word, and no verb. The logical hypothesis is that the new word is a verb. A special syntactic module is therefore passed the following string •NP(P) verb(plunk) NP(N) PP(In,5)V This module examines the string and produces a new frame:

VERB: plunk
AGENT: P
OBJ: M
in: S

The system can than ask if "to plunk something somewhere" means "to place something somewhere", and upon getting an affirmative reply, can add tha new frame to those associated with tha concept #PLACE.

Since the system uses the surface frames to generate its own replies, it can now use this new frame itself whan it talks. Whan the system wants to generate a clause, it passes a selected frame, the constituents, and a list of syntactic features to a clause generator which outputs the specified form. (Thus, clause-level syntax can be used by tha system in three different modes: (1) to comment on the grammatlcality of a string of case markers, (2) to construct a new surface frame, and (3) to generate clauses whan the system itself replies.)

4.2 Knowledge of how Clause-Functions are

As illustrated in Fig. 3, knowledge of haw clause-function concepts are described is also expressed as two levels.

Each clause function has a conceptual slot indicating what types of action can be used with that clause type (in this case, tha action #PLACE), and a list of surface frames indicating different ways in which the concept can be expressed.

A clause-type frame currently includes any spacial words which introduce tha clause (la. "why" or "how come"), together with a list of syntactic properties which should be present in the clausa.

This list of syntactic properties night include SVIMV, "subject-verb inversion" (as in 'why dots someone do something'), or "subject deletion", "ING form", and "use of a particular preposition" (as In "fron doing something').

CONCEPT: #WHY

CONCEPTUAL SLOTS:

ACTION: #PLACE

SURFACE FRAMES:

Why ACTION(SVINV)	**(as in: "Why does someone do something")**
How come ACTION()	**(as in: "How come someone does something")**

Fig. 3: Knowledge about Clause Functions

These syntactic features, however, need not be inflexible rules. Sentence understanding can still proceed even If the syntactic features found by syntax do not exactly natch those specified by the clause-function frame. Thus, an inadvertent ungrammatically can readily be recognized as such, and processing can continue.

4.2.1 Using the Clause Function Knowledge

In this section we examine how this clause function Knowledge can be used.

(1) With no uncertainty:
If the input sentence is "Why did you place an x in the center square", then during the senentlc clustering the string "do agent verb obj pp" is passed to syntax, which replies that subject-verb inversion has taken place.
When examining the whole clause, the system sees that it exactly matches one of the surface frames for a #WHY-type question, since it starts with the word "why" and contains subject-verb inversion.
Suppose, however, the sentence had been "Why you place an x in the center square", or "How come did you place an x in the center square". Each of these sentences matches a surface frame for a #WHY-type question, except that in both cases subject-verb inversion is incorrect. In such a case, the system can, if it chooses, decide that the user has made a minor error, and allow the sentence to be processed anyway. The locally-driven semantics-first approach lets this happen in a natural way.

(2) A new surface frame:
Another problem arises when a new clause Introducer is encountered, as in: "Wherefore did you place an x in the center square". Here, as described in section 3, the system hypothesizes that this may be a new way of asking a #WHY-type question. Since syntax reports that subject-verb inversion has taken place, the system can therefore create a new surface frame:

Wherefore ACTION(SVINV)

to be added to the frames associated with #WHY.

4-3 Comments

In summary, the adaptive system stores Its linguistic Knowledge in a very accessible form. It is not embedded in the parsing logic. Knowledge of how actions and clause-functions are described is represented in a descriptive, manlpulable format. Syntax is domain independent, and is used only to make comments, with semantics playing the guiding role. This organization allows the parsing/understanding process to proceed in a flexible fashion.

5. FLEXIBLE ORGANIZATION OF SEMANTIC CONSTRAINTS

In an adaptive parsing system, semantic constraints must also be in a flexible, manlpulable form so that they can assist in the inference-making process. This contrasts with a non-adaptive system (a system where the syntax and vocabulary is assumed to be fixed). In e non-adaptive system, semantic constraints can often be included in an ad-hoc fashion, without really being explicit about how they interrelate.
To illustrate this difference, let us consider the following sentences.

(1) After you played the top WEST square could I beat you if I played the bottom right square.

(2) After you played the top right square could I beat you if I played the bottom WEST square.
[We shall assume that the word "west" is not known to the system.]

First let us see how a non-adaptive system would handle such sentences, assuming no unknown words. (See Fig. 4)

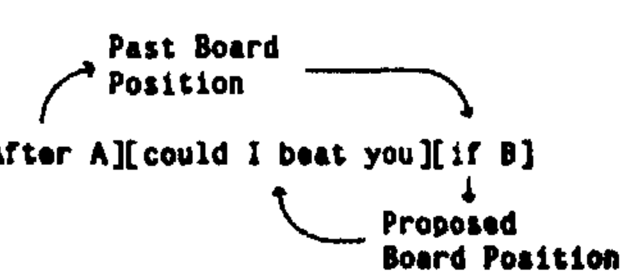

Fig. 4: Communication between Clauses

(a) When it encounters the "after" clause, it looks in its list of past moves to see if one matches the description given. If so, the resulting board position is used as context for the rest of the sentence. If not, the system aborts, giving an error message such as "You have referred to a non-existent past move".
(b) When processing the "if" clause, the system takes the board position produced by the "after" clause, and sees if the move referred to is legal in that context. If so, it produces a new board position. If not it aborts with an error message such as "You have proposed an illegal move".
(c) Finally, when processing the main clause, the system takes the board position produced by the "if" clause and examines it to answer the question.

Thus, the following semantic constraints art programed in: (a) that the move described in an "after" clause must refer to a previous Bove, and (b) the Bove described in an "if" clause wist be legal in its board position context.

In an adaptive systea, these constraints can be used in an active fashion to help infer the meaning of new words. For instance, if sentence (1) is to make sense, then "the top west square" must refer to a square played while the bottom right square was free (ie. was still a legal aove). Thus, as illustrated in the sample session, If two top squares contain an x, but only one was played before the bottom right square was filled, then that Bust be the square being described. Similarly, in sentence (2), "the bottom west square" Bust refer to some square that was free when the top* right square was played.

Thus the semantic constraints which are added almost as after-thoughts in a non-adaptive system to handle the unlikely event of a user typing NONSENSICAL input, become a central part of the inferential aechanlsm in an adaptive system, for the very reason that they allow the system to discard NONSENSICAL INTERPRETATIONS of the meanings of new words.

These two sentences also illustrate that such constraints can operate globally between widely separated parts of the sentence. Furthermore, they can operate equally effectively in two (or more) "directions". In other words, certainty in the "if" clause can help resolve uncertainty in the "after" clause, or alternately, certainty in the "after" clause can help resolve uncertainty in the "if" clause.

Notice that there is also a third "direction" in which these constraints can be used. Consider the following sentence.

(3) WHEN you played in the center square could I beat you if I played the top right square.

If "when" is an unknown word, then the system must try to infer what clause-function it might refer to. In so doing, the system checks whether the "when" clause describes a previous move whose subsequent board position allows the aove described by the "if clause. If so then "when" alght aean "after". If not, then "when" presumably means something else. Here we see certainty inside the two clauses helping resolve uncertainty as to the function of one of the clauses.

The clear implication of these exaaples is that these semantic constraints Bust be incorporated in a much more systematic and flexible fashion than is necessary in a non-adaptive system.

5.1 <u>Implementing</u> these Constraints

The adaptive system handles this problem as follows:

(a) If there is uncertainty in the "arter" clause, then it produces not ONE, but rather a LIST of board positions (corresponding to different possible interpretations of what the clause Beans). Associated with each is the uncertain constituent (in this case, the NP "the top west square") and the referent of that constituent that corresponds to that board position possibility (ie. the actual square which the NP alght be describing). Each of these board positions are fed in turn to any "if" clauses in the sentence. During this proctss, the semantic constraints might invalidate soae of these board positions. This leaves the system with a more selective group of referents that the NP Bight be describing. In this way, the user can be asked a clarifying question that Is as "Intelligent" as possible.

(b) A similar process takes place if the "if" clause contains uncertainty. A list of possibilities is compiled by the "if" clause, tested in further "If* clauses (if any), and only then is the user asked for clarification.

This method is ONE way of allowing such semantic constraints to help the Inference process, and thereby let the system ask as intelligent questions as possible. There are a great many intra-utterance constraints of this sort, whose nature and implications have never been systematically explored. The development of adaptive parsing techniques will provide motivation to explore these more fully.

This example, in fact, illustrates a such aore general phenomenon. In natural language, there are many levels of syntactic and semantic knowledge which contribute to the understanding of a sentence. These all interact, and therefore constrain each other in complex ways. In a fully adaptive system, one must be prepared to help resolve uncertainty on ANY of these levels by taking advantage of certainty on many other levels. To allow this, one Is forced to think out fully and systematically exactly how all this knowledge interacts. This could be one of the greatest contributions of studying language froa an adaptive standpoint.

6. <u>SENTENCE. CLARIFICATION AND EVALUATION</u>

In this section, we describe how semantic constraints con be used to resolve input uncertainty. After the semantic clusters have been formed, cluster expansion and connection groups these into one or more sentence hypotheses. Each sentence hypothesis is a set of clauses which span the entire sentence. The system must then "evaluate" these clauses (ie. determine their meaning).

The clauses are evaluated in the following order. First any "after" clause is evaluated. Then any "if" clauses ore evaluated left to right. Finally the main clause is evaluated. Thus if the sentence is

"[if A][after B][could C][if D]"

the clauses would he evaluated in order B, A, D, C.

In the process of this evaluation, any uncertainty present is resolved. The uncertainty might be that a constituent contains an unknown word, or it might be due to the use of a new action or clause-function surface frame.

Constituent Uncertainty:

Sometimes constituent uncertainty can be resolved from information available in a single clause. Sometimes information from several clauses is used.

For example, consider the clause "After I placed a circle in the center square". If the word "circle" is unknown, then as described in section 4, previous processing indicates that this is an "after" clause doscribing a move, and that "a circle" probably refers to some mark. To evaluate this clause, the system examines the previous moves and presumably finds that "I" indoed did play In the center square and that the mark involved was an "0M. Thus the likely moaning of "circle" is resolved from information available entirely within this clause.

On the other hand, as described in section 5, with the clause "After I placed an o in the top WEST square", if "west" is unknown then several squares may bo being described. Section 5 describes how the system allows information from other clauses to constrain the square being described. When as rostricted as possible a list of squares is finally determined, it is passed to a routine whose job is to resolve noun phrase uncertainty. In this case, the routine will examine the different adjective-concepts (such as #RIC.HT, #LEFT. #CORNER. IBOTTOM, etc.) to see which might describe possible squares, and then ask.the user which was meant.

New Surface Frame Uncertainty:

If a new surface frame is used to express a familiar action (ie. a new verb and/or prepositions marking the NPs), the different possible meanings are automatically considered when the system attempts to fill the conceptual slots of different action concepts. Even if only one concept has the appropriate slots, the system asks to be sure. Sometimes, however, several actions have the same conceptual slots (such as "I beat you", "I tie you", and "1 lose to you"). In this case, of course, the system must ask which was meant.

If a new surface frame is used to express a familiar clause function, the system currently uses the following simple constraints to delimit its possible meaning. Each sentence is assumed to have one and only one main clause. "After" clauses must describe a previous move. "If clauses must describe some possible move (either in the current game context, or in the context of any "after" clause in the sentence). "Why" clauses must refer to a previous move. "Can" and "Mow can" clauses must refer to the action of beating, tieing, losing, preventing, allowing, or forcing. Imperative clauses must refer to a possible current move. In our simple domain, these constraints usually delimit the meaning of a new surface clause-function frame to a single possibility.

This section has described how the adaptive system can constrain input uncertainty. Clearly, in a more complex domain, one would want to use much more sophisticated constraints. The important point is that the locally-driven, semantics-first design provides a very natural framework for incorporating such constaints. The purpose of the present work is to take a first step in exploring the design of such a system.

7. CONCLUSION

Language communication is an inherently adaptive medium. One sees this clearly if one takes a problem to a lawyer and spends time trying to assimilate the related "legalese". One also sees it in any conversation where a person is trying to convey a complicated idea, expressed in his own mental terms, to someone else. The listener must try to relate the words he hears to his own set of concepts. Language has, presumably, evolved to facilitate this sort of interaction. Therefore it is reasonable to expect that a good deal of the structure of language is in some sense set up to assist in this adaptive process. By the same token, studying language from an adaptive standpoint should provide a fresh perspective on how the various levels of linguistic structure interact.

REFERENCES

[I] Davies, D.J.N., and Isard, 5.D., 'Utterances as Programs,' presented at the 7th International Machine Intelligence Workshop, F.dinburg, June 1972.

[2] Enea, H., and Colby, K.M., 'Ideolectic Language Analysis for Understanding Doctor-Patient Dialogs', Proceedings of the 3rd IJCAI, Stanford, August 1973.

[3] Fillmore, C.J., 'The Case for Case', in 'Uiuversals in Linguistic Theory', Bach and Harms (Eds.), Holt, Rinehart, and Winston, Inc., Chicago 1968.

[4] Joshi, A.K., and Weischedel, R.M., 'Some Frills for the Modal TIC-TAC-TOE of Isard and Davies: Semantics of Predicate Complement Constructions,' Proceedings of the 3rd IJCAI, Stanford, August 1973.

[5] Miller, P.L., 'A Locally Organized Parser for Spoken Input', Comm. ACM 17, 11 (Nov. 1974), 621-630.

[G] Miller, P.L., 'An Adaptive System: for Natural Language Understanding and Assimilation', RLE Natural Language Memo No. 25, MIT, February 1974.

[7] Reddy, D.R., Erman, L.D., Fennell, R.D., and Neeley, R.B., 'The HEARSAY Speech Understanding System', Proceedings of the 3rd IJCAI, Stanford, August 1973.

[0] Walker, D.E., 'Speech Understanding through Syntactic and Semantic Analysis', Proceedings of the 3rd IJCAI, Stanford, August 1973.

[9] Weizenbaum, J., 'Eliza- a Computer Program for the Study of Natural Communication between Man and Machine', CACM 9, 1972.

[10] Winograd. T. Procedures as a Representation of knowledge in a Computer Program for Understanding Natural Language, MAC-TR-84, Project MAC, MIT, Cambridge, Mass., February 1971.

[11] Woods, W.A., and Kaplan, R.M., 'The Lunar Sciences Natural Language Information System', BBN Report No. 2205, Bolt, Beranek, and Newman Inc., September 1971,

[12] Woods, W.A., and Nakhoul, J., 'Mechanical Inference Problems in Continuous Speech Understanding', Proceedings of the 3rd IJCAI, Stanford, August 1973.