Maintaining Consistency and Plausibility in Integrated Natural Language Understanding

Toyoaki Nishida, Xuemin Liu, Shuji Doshita, and Atsushi Yamada
Department of Information Science
Kyoto University
Sakyo-ku, Kyoto 606, Japan
phone: 81-75-751-2111 ext. 5396
email: nishida%doshita.kuis.kyoto-u.junet%japan@relay.cs.net

Abstract

In this paper, we present an inference mechanism called the *integrated parsing engine* which provides a uniform abductive inference mechanism for natural language understanding. It can (1) make plausible assumptions, (2) reason with multiple alternatives, (3) switch the search process to the maximally plausible alternative, (4) detect contradiction and tame conclutions which depend on inconsistent assumptions, and (5) update plausibility factor of each belief based on new observations. We demonstrate that a natural language understanding system using the integrated parsing engine as a subsystem can pursue a guided search for most plausible interpretation by making use of syntax, semantics, and contextual information.

1 Introduction

Natural language understanding involves lots of hard issues such as various types of ambiguities, indeterminacies caused by ellipses or fragmental utterances, or ill-formedness. Being confronted with these difficulties, it does not seem reasonable to seek for a method of *logically* deducing the speaker's intended meaning or plan from utterances. Instead, it is much more natural to characterize natural language understanding as an abductive process of exploring most plausible interpretation which can explain given utterances.

In this paper, we present an abductive inference mechanism, called the *integrated parsing engine*, for natural language understanding. The integrated parsing engine is able to:

- make plausible assumptions at appropriate time
- reason with multiple alternatives based on different sets of assumptions
- switch the search process to the maximally plausible alternative
- detect contradiction resulting from inconsistent assumptions and eliminate all conclutions which depends on these assumptions
- update plausibility factor of each belief based on new observations.

Thus, the integrated parsing engine is general enough to carry out linguistic and nonlinguistic inferences in a uniform manner, by drawing information from various sources: syntax, semantic, discourse, pragmatics, or real world.

In the remainder of this paper, we first describe mechanisms for maintaining consistency and plausibility. We then show how these two mechanisms interact to guide the inference process. Finally, we use an implemented example to demonstrate how the integrated parsing engine is used to interpret sentences by taking contextual factors into account.

2 Maintaining Consistency

The CME (Consistency Maintenance Engine) is a component of the integrated parsing engine responsible for maintaining consistency among beliefs. Basic design principles of the CME is based on de Kleer's ATMS (Assumption-based Truth Maintenance Engine) [de 86]. The CME maintains a set of alternative beliefs, each of which consists of a set of assumptions and their conclusions, as follows:

$$\begin{array}{ll} \textit{alternative 1} & \{A_{11},\ldots,A_{1m_1}\}\ B_{11},\ldots,B_{1m_1} \\ & \vdots \\ \textit{alternative n} & \underbrace{\{A_{n1},\ldots,A_{nm_n}\}}_{\textit{environment}}\ \underbrace{B_{n1},\ldots,B_{nm_n}}_{\textit{conclusions}} \end{array}$$

An external problem solver is assumed to exist which makes assumption, adds conclusion, and detects contradiction.

The main task of CME is to maintain alternative beliefs by removing all alternatives whose set of assumptions has turned out contradictory. Like ATMS, the CME takes advantage of the following monotonic property:

> if a contradiction is derived from a set of assumptions A, then contradiction is also derived from any set of assumptions B such that $B \supset A$.

Thus, if contradiction is derived from a set of assumptions $\{B,D\}$, alternative interpretations depending on sets of assumptions such as $\{B,C,D\}$, $\{A,B,D\}$, $\{A,B,C,D\}$, ... are removed. In addition, the CME keeps records of contradictory sets of assumptions to prevent any interpretation depending on them from being considered in future.

Unlike ATMS whose control regime is bread-first, our CME uses a tree called the environment tree, or the *E-tree* for short, to guide the search process. Each node of the E-tree represents an environment, a set of assumptions. Each arc of the E-tree represents that a lower node is derived from the upper node by making one more assumption. Thus in figure 1, E_0 is the root node, and it represents an environmnet without any assumption. Nodes below E_0 represent environments with one or more assumption added to its parent node's environment. Thus, $E_1 = E_0 \cup \{A_1\} = \{A_1\}$, $E_{11} = E_1 \cup \{A_{11}\} = \{A_1, A_{11}\}$, and so on.

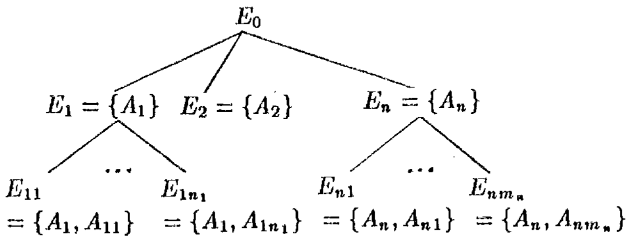

E_0
$E_1 = \{A_1\}$
$E_2 = \{A_2\}$
$E_n = \{A_n\}$
...
$E_{11} = \{A_1, A_{11}\}$
$E_{1n_1} = \{A_1, A_{1n_1}\}$
$E_{n1} = \{A_n, A_{n1}\}$
$E_{nm_n} = \{A_n, A_{nm_n}\}$

Figure 1: The E-tree

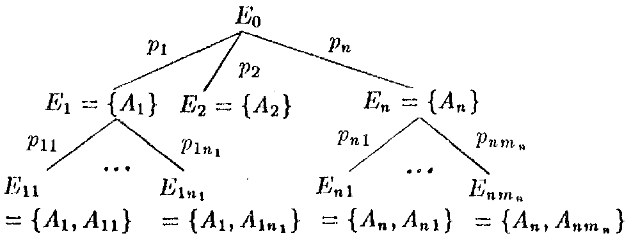

E_0
p_1
p_2
p_n
$E_1 = \{A_1\}$
$E_2 = \{A_2\}$
$E_n = \{A_n\}$
p_{11}
p_{1n_1}
p_{n1}
p_{nm_n}
...
$E_{11} = \{A_1, A_{11}\}$
$E_{1n_1} = \{A_1, A_{1n_1}\}$
$E_{n1} = \{A_n, A_{n1}\}$
$E_{nm_n} = \{A_n, A_{nm_n}\}$

Figure 2: The E-tree with Conditional Probabilities

We assume that a set of assumptions made at the same parent node are mutually exclusive. Although this is a rather strong assumption, it makes sense in natural language understanding since many assumptions being made during the natural language understanding process are mutually exclusive. Even if this is not the case, any set of assumptions can be transformed into a set of mutually exclusive assumptions by adding appropriate conditions. Although this is a cumbersome solution, it does not often take place in natural language understanding and most importantly it saves the amount of computation.

Note that the CME alone cannot determine which way to go when there is more than one possibility of extending the set of beliefs. This information is provided by the PME, as described in the next section.

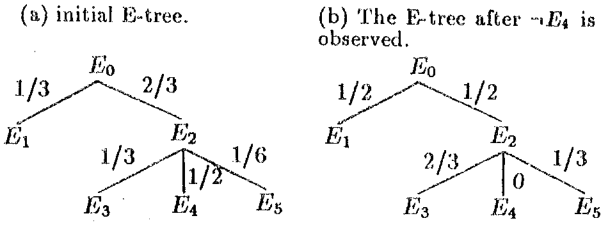
3 Maintaining Plausibility

The PME (Plausibility Maintenance Engine) maintains estimations of how plausible each environment is. This information is given as conditional probabilities and it is kept as annotations to each arc of the E-tree. Thus, in figure 2, which is a slightly more precise version of figure 1, p_i stands for $P(E_i)$, p_{ij} for $P(E_j|A_i)$, p_{ijk} for $P(E_k|A_i, A_j)$, *etc.*

It follows from the property of conditional probability that

$$P(E_i|\ldots E_j \ldots) = 0,$$

if $i \neq j$ and E_i and E_j are immediate children


(a) initial E-tree.

(b) The E-tree after $\neg E_4$ is observed.

Figure 3: A Sample E-tree with Annotation

of the same parent. Furthermore,

$$P(E_i|\ldots\neg E_j\ldots) = 0,$$

if E_j is a parent node of E_i.

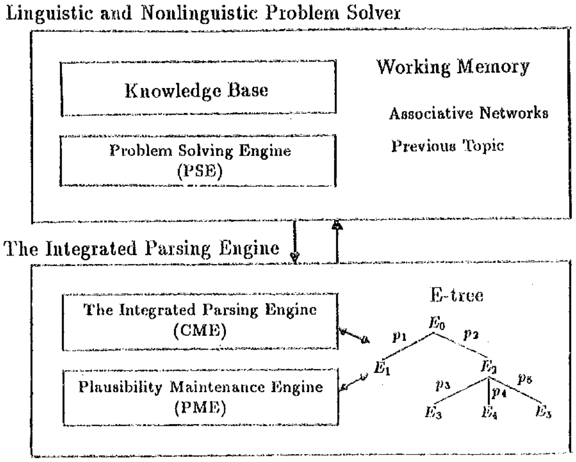
Initial value of p_i's are to be given from the external problem solver. The PME's role is to maintain estimation of prausibility by taking into account given observations. Currently we only take $\neg E$, the event of environment E running into contradiction, as an observation. We use a Bayes' law to modify $P(A)$ into $P(A|\neg E)$. Thus,

$$\begin{aligned} P(E_i|\neg E_j) & \\ &= \frac{P(\neg E_j|E_i)\cdot P(E_i)}{P(\neg E_j)} \\ &= \frac{(1-P(E_j|P_i))\cdot P(E_i)}{1-P(E_j)}. \quad (1) \end{aligned}$$

if E_i and E_j are brothers, (1) is further simplified to:

$$\frac{P(E_i)}{1-P(E_j)}. \quad (2)$$

For example, suppose it has turned out that environment E_4 is in contradiction and hence $\neg E_4$ is observed (figure 3(a)). The annotations to the E-tree are updated as in figure 3(b). Notice that the update of conditional probability can be done based on local information.



Linguistic and Nonlinguistic Problem Solver
Knowledge Base
Working Memory
Associative Networks
Previous Topic
Problem Solving Engine (PSE)
The Integrated Parsing Engine
The Integrated Parsing Engine (CME)
Plausibility Maintenance Engine (PME)
E-tree
E_0
p_1
p_2
E_1
E_2
p_3
p_4
p_5
E_3
E_4
E_5

Figure 4: The Structure of a Natural Language Understanding System with the Integrated Parsing Engine as a subsystem

4 Natural Language Understanding System Using the Integrated Parsing Engine as a Subsystem

The integrated parsing engine consists of the CME and the PME. The architecture of a natural language understanding system with the integrated parsing engine as a subsystem is shown in figure 4.

The knowledge base contains various types of information for language comprehension, including lexicon, morphology, syntax, semantics, discourse, pragmatics, commonsenses, and so on. The whole system is controled by the problem solving engine (PSE). The PSE can access to the knowledge base and use the integrated parsing engine as an aid to seek for most plausible interpretation. Input texts are analyzed in a sentence-by-sentence manner. The discourse structure is maintained as a previous topic in the working memory.

When it scans a new sentence, the PSE first initialize the E-tree with only the root node. Then the PSE repeats the following cycle:

(step 1) choose a leaf node with the highest probability as a working envirioment

(step 2) repeatedly derive conclusions from

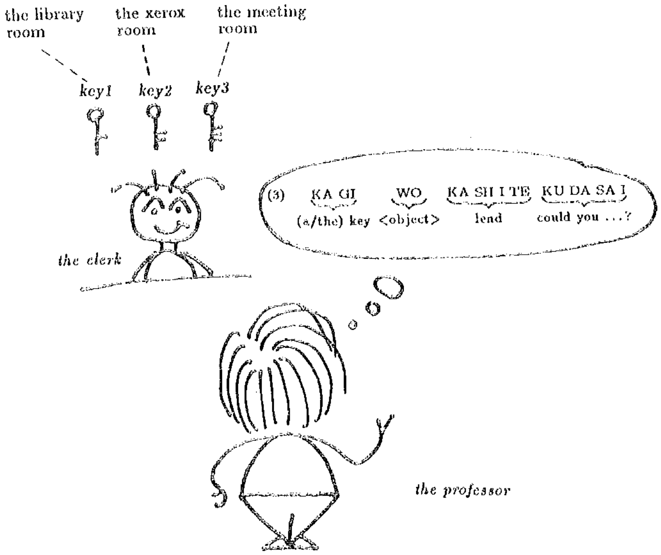


Figure 5: Sample Dialog Environment

believed propositions until either (a) the goal is achieved, (b) contradiction is derived, or (c) no more conclusion is derived unless making more assumption.

In case (a), the process halts.

In case (b), the process is passed to the PME, which modifies current estimation of plausibility so that this fact is reflected, then an alternative of meximum plausibility is chosen and is suggested to the CME.

In case (c), the process also is passed to the PME, which assigns plausibility to new nodes, and working environment is chosen again.

The integrated parsing engine has been written in Lisp. It is running with a small exmerimental grammar for Japanese. The next section shows how it works.

5 An Example

Suppose a dialog environment in which a professor speaks to a clerk to borrow a key of some rooms (figure 5) and utters the following Japanese sentence:

(3) $\underbrace{\text{KA GI}}_{\text{(a/the) key}}$ $\underbrace{\text{WO}}_{\text{<object>}}$ $\underbrace{\text{KA SH I TE}}_{\text{lend}}$ $\underbrace{\text{KU DA SA I}}_{\text{could you ...?}}$

"could you lend (me) (a/the) key?"

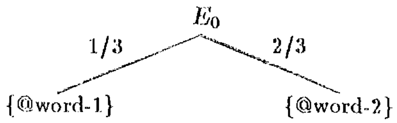


Figure 6: E-tree after assumptions *@word-1* and *@word-2* are made

The referential meaning of this sentence is ambiguous if there is more than one key in a given situation. Suppose three keys are there: *key1* for a library room, *key2* for a xerox room, and *key3* for a meeting room.

Although sentence (3) is ambiguous in normal contexts, it becomes much less so if it follows sentences like:

(4) HO N WO KO PI I SHI TA I NO DE SU GA
"I'd like to xerox some books."

Even if no previous sentence is spoken, sentence (3) is acceptable in a situation where the speaker and the hearer mutually believe that the xerox room is accessed so often that "the key" is usually used to refer to *key2*, the one for the xerox room.

Note that the omission of the patient case does not matter in usual situations, since there is a strong default that the filler of this case is the speaker.

Now let us show how sentence (3) is analyzed in a context where sentence (4) was previously uttered. The task of analyzing input starts from recognizing words. Lots of ambiguities arise in this phase. For sentence (3), 'KA' might be a single word 'KA' (postposition marking interrogative) or a part of a longer word 'KAGI' (key). Since longer match is considered to be more plausible in general case in Japanese analysis, we assign larger number of probability to the latter possibility. Following this analysis, the PSE makes the assumptions to the integrated parsing engine:

@word-1 (take the sequence 'KA' as a word):
⇒ probability 1/3.

@word-2 (take the sequence 'KAGI' as a word):
⇒ probability 2/3.

Accordingly, the CME extends the initial E-tree as in figure 6. Since, the environment E_1 has the highest plausibility, the CME chooses it for the next environment and control is returned to the PSE.

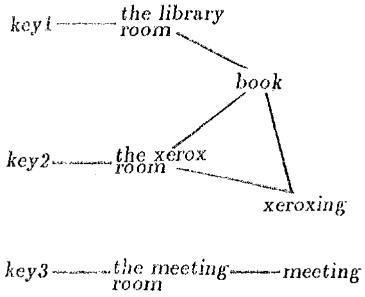


Figure 7: An Associative Network between Concepts

Now the PSE tries to derive further conclusion in the chosen environment. After having recognized that the part of speech of the word 'KAGI' is noun, the PSE tries to find out the referent of the noun and realizes that three ambiguities arise in this situation. Again, the PSE calls the CME to make assumptions. At the same time, the PSE is called for to assign estimated conditional probabilities to each assumption.

Currently, the system uses an associative network as shown in figure 7 to determine plausibility. Nodes of this network represent either a concept or an instance, and arcs mean that the two concepts or instants at its both ends have a certain relation. Those items which have dense connections to previous subjects are considered to be plausible as a referent. In our example, since the node *xerox* is marked as the previous subject *key2* is considered most plausible, while *key1* is less plausible and *key3* much less. Thus, the following assumptions are made: [1]

@referent-1 (consider 'KAGI' to refer to *key1*):
⇒ probabiliy 1/3.

@referent-2 (consider 'KAGI' to refer to *key2*):
⇒ probabiliy 1/2.

@referent-3 (consider 'KAGI' to refer to *key3*):
⇒ probabiliy 1/6.

In case no previous utterance is given, the PSE will consult information given as *a priori* measurements.

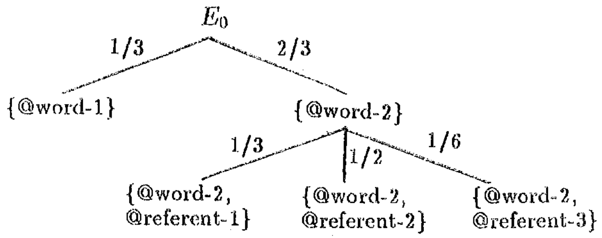
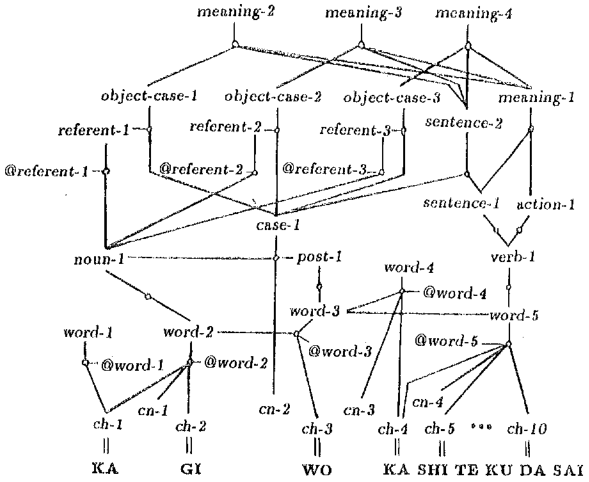


Figure 8: E-tree after assumptions about the referent of 'KAGI' (key) are made



Notice that all part of this network is not explored in actual processing.

Figure 9: Dependency of Beliefs

The E-tree now becomes as in figure 8, and {@word-2, @referent-2}, which is the most plausible environment at this point, is chosen as the next environment. The analysis is continued this way until the semantic representation is obtained for the whole sentence. The interpretation obtained this case is:

$$\left[\begin{array}{lcl} \textit{event} & = & \textit{asking-for} \\ \textit{actor} & = & <\textit{the speaker}> \\ \textit{object} & = & \textit{key2} \end{array}\right]$$

Figure 9 shows the dependency structure of beliefs related to this analysis.

Notice that the efficiency of the analysis is significantly improved when strong expectation exists. For example, although character 'SHI' in sentence (3) has many possible interpretations in Japanese, the system is not annoyed by those ambiguities, since this part of the sentence just goes as expected. The system may come to suspect it only when most of its expectation fails.

[1] Currently we use a very simple algorithm for assigning those value: when there are three alternatives, the densest connection receives the value (1/3), the second (1/2), and the third (1/6), regardless of how closely they are related to each other. We plan to develop a much more precise method in a near future.

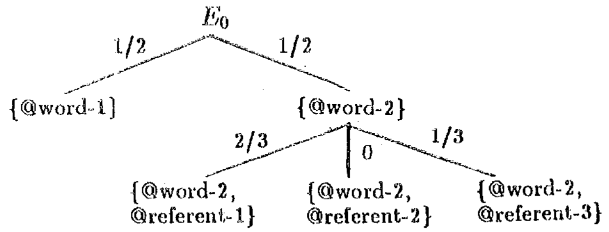


Figure 10: E-tree after assumptions about the proposed interpretation based on {@word-2, @referent-2} is rejected

Now suppose the above interpretation is rejected for some reason, say by explicitly negated by the speaker. Then the system will eventually produce an alternative interpretation taking *key1* as a referent, by changing annotations to the E-tree as in figure 10.

6 Related Work

This paper was inspired by a number of works. A massively parallel parsing by Waltz and Pollack [WP85] has demonstrated the effect of integration through a uniform computation mechanism (marker passing) in context-dependent comprehension of discourse. They have pointed out the importance of non-logical, associative relation between concepts. Charniak has pointed out the abductive nature of language comprehension. Charniak's Wimp [Cha86] uses a marker passing mechanism as a basis of abductive inference engine for language comprehension. But it is not used alone; it is augmented by a logical process called path proof. In a parser used in Lytinen's MOP-TRANS [Lyt86], a mechanism is provided to allow close interaction between syntax and semantics, while keeping the modularity of the system. Another thing to note is that Lytinen's integrated parser makes use of strong semantic expectation to constrain the search.

The integrated parsing engine presented in this paper takes advantages of these preceding works. Unlike Waltz and Pollack, and like Charniak and Lytinen, our integrated parsing engine has a hybrid architecture for logical and non-logical inferences. What is novel with our integrated parsing engine is the method of integrating and maintaining logical and non-logical information obtained from various source. In addition, the integrated parsign engine provides a concise and high level mechanism for abductive reasoning. We have carefully chosen a set of reasonably high-level functions necessary for abductive reasoning. This serves to much simplifying natural language understanding system than otherwise.

7 Concluding Remarks

We have presented an inference engine for integrated natural language understanding, based on a characterization of natural language understanding as an abductive process. The essence of our approach is connecting consistency maintenance engine and plausibility maintenance engine closely enough to allow their dense interaction. Although we have shown rather "low level" issues, we believe the same idea is applicable to "higher level" problems such as inferring speaker's intention and plan.

References

[Cha86] Eugine Charniak. A neat theory of marker passing. In *Proceedings AAAI-86*, pages 584–588, 1986.

[de 86] Johan de Kleer. An assumption-based tms. *Artificial Intelligence*, 28:127–162, 1986.

[Lyt86] Steven Lytinen. Dynamically combining syntax and semantics in natural language processing. In *Proceedings AAAI-86*, pages 574–578, 1986.

[WP85] D. Waltz and J. B. Pollack. Massively parallel parsing: a strongly interactive model of natural language interpretation. *Cognitive Science*, 9:51–74, 1985.