RESOLVING PRONOUN REFERENCES

Jerry R. HOBBS*

Dept. of Computer Sciences, City College, CUNY, New York, U.S.A.

Received January 1977

Two approaches to the problem of resolving pronoun references are presented. The first is a naive algorithm that works by traversing the surface parse trees of the sentences of the text in a particular order looking for noun phrases of the correct gender and number. The algorithm clearly does not work in all cases, but the results of an examination of several hundred examples from published texts show that it performs remarkably well.

In the second approach, it is shown how pronoun resolution can be handled in a comprehensive system for semantic analysis of English texts. The system is described, and it is shown in a detailed treatment of several examples how semantic analysis locates the antecedents of most pronouns as a by-product. Included are the classic examples of Winograd and Charniak.

1. Introduction

1.1.

The importance of having the right algorithm for resolving pronoun references, or finding the antecedent of a pronoun, can be seen on American television in any episode of the George Burns and Gracie Allen re-runs, for much of Gracie's humor depends on her having the wrong algorithm.[1] For example, an episode is built around her misunderstanding of a fire inspector's warning:

> There's a pile of inflammable trash next to your car. You'll have to get rid of it.

In Jespersen (1954: 143) the problem of pronoun resolution rates brief mention:

> An ambiguity (not very serious) may sometimes arise when there are two antecedents to which *it* may refer: If the baby does not thrive on raw milk, boil it.

What counts as serious depends on one's point of view.

In this paper, two approaches to pronoun resolution are considered. The first is a simple, efficient, but naive algorithm working on the surface parse trees of the sentences in the text. Examination of several hundred examples from a variety of published texts shows that in spite of its obvious flaws, the algorithm works remarkably well.

In the second approach, it is shown how pronoun resolution happens in a total system for semantic analysis. And the word 'happens' is appropriate here. Charniak (1972) demonstrated rather convincingly that in order to do pronoun resolution, one had to be able to do everything else. In the latter part of this paper it is argued that once everything else is done, pronoun resolution comes free – it happens automatically. A system for the semantic analysis of English texts is described, and it is shown in a detailed treatment of several examples how semantic analysis locates the antecedents of most pronouns as a by-product.

The problems studied in this paper are addressed from a computational, analytic point of view, that is, within a clearly specified framework for analyzing texts rather than generating them. In some generative approaches, a noun phrase in deep structure is transformed into a pronoun by a pronominalization rule, under the condition of 'identity of reference' with another noun phrase. Little more is said about this condition; presumably the one generating the sentence is somehow aware of this identity of reference. However, from the computational, analytic point of view, we must be very precise about how a listener could *become* aware of this identity of reference. When he hears or reads a pronoun, how, out of all the possible structures which could serve as an antecedent, might he be able to pick the correct one. Our emphasis is directed toward developing algorithms for doing this. Even if it is the generation of texts one is interested in, the analytic point of view may be the correct one to take, for as Olson (1970) argues, the speaker elaborates his description of an entity just to the extent that will allow his listener to identify it.

* The author wishes to thank Eileen Fitzpatrick for her helpful suggestions. This work was supported by the CUNY Faculty Research Award Program Grants Nos. 11233 and 11655 and National Science Foundation Grant No. MCS-76-82164.

Present address: SRI International, Menlo Park, California, U.S.A.

[1] I am indebted to Eileen Fitzpatrick for calling this to my attention.

1.2. *Review of the natural language processing literature*

Most work by linguists on the problems of pronoun resolution has been concerned with elucidating syntactic constraints on the coreferentiality and non-coreferentiality of two entities occurring in the same sentence. Some of this work will be reviewed in Part 2. This section is a brief review of the most important work in natural language processing, in which the problem of locating antecedents has been addressed more directly.

Winograd (1972) was the first to write procedures for locating antecedents, in his system for manipulating and carrying on dialogs about a blocks microworld. He collects all possible referents and rates their plausibility on the basis of syntactic position, apparently in an order similar to that which the naive algorithm of Part 2 defines. A subject is favored over an object and both are favored over the object of a preposition. In addition, 'focus' elements are favored, where focus is determined from, among other things, the answers to *wh*-questions and from indefinite noun phrases in *yes-no* questions.

In his system for general natural language inferencing, Rieger (1974) finds the antecedent of any definite entity by creating and narrowing down a candidate set of possible antecedents, based on the known properties of the entity. The difficulty with this approach for pronouns is that normally the only explicit property we have is new and not very useful for identification. If implied properties are included, then one runs into an overall problem with Rieger's system (shared by Charniak's). There are no controls on inferencing; everything that can be inferred is. When more than one candidate remains, the most recently referenced is chosen, where recency is determined by a 'system clock', i.e. determined by a complex and largely unspecified order of processing.

The chief value of Charniak's work has been to show just how difficult the pronoun resolution problem is. In particular, he showed how spurious the recency principle is. His thesis (Charniak 1972) contains a wealth of difficult cases in the guise of children's stories, which show that arbitrarily detailed world knowledge can be required to decide upon an antecedent. He points out that the knowledge required for pronoun resolution is just that which might be required by a conversational system which is asked questions about the stories. He internalizes these questions in the form of 'demons', which are axioms whose antecedents have been matched and which are looking for their consequents. However, he offers no solution to the problem of what questions are the appropriate ones to ask. Section 3.2 of this paper may be seen as an attempt to solve this problem.

Wilks (1974) has given a very nice partial solution to the problem of deciding among competing plausible antecedents, based on the use of selectional information to maximize the redundancy. In addition, he uses a bidirectional search through a data base of world knowledge to resolve pronouns. His approach, in general, is similar to that of operation 4 in Section 3.2, although he lacks a notion of salience.

2. The syntactic approach: The naive algorithm

In this section a naive algorithm for finding antecendents of pronouns is described and related to previous linguistic research. Results are presented of a statistical study of the algorithm's effectiveness on 300 examples of pronoun occurrences from three very different texts.

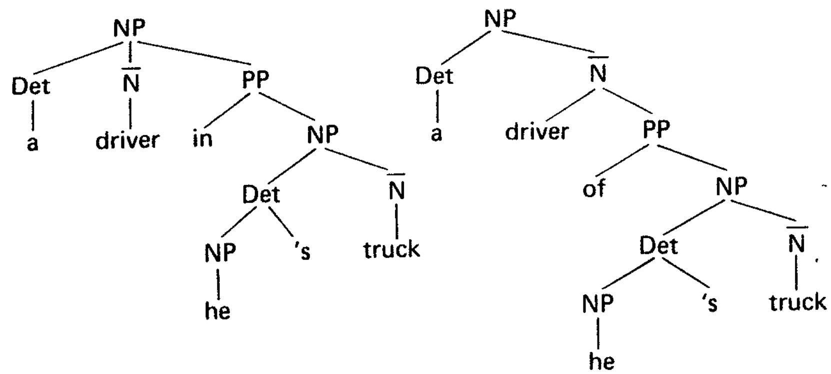
2.1. *The algorithm*

In what follows reference will be made to the 'surface parse tree'. By this is meant the tree that exhibits the grammatical structure of the sentence – its division into subject, verb, objects, adverbials, etc. – without permuting or omitting any of the words in the original sentence. That is, the terminal nodes of the tree taken in left-to-right order form the English sentence. It will be assumed however that certain syntactically recoverable omitted elements are available as antecedents, as described below.

It will be necessary to assume that an NP node has an $\bar{N}$ node below it, as proposed by Chomsky (1970), to which a prepositional phrase containing an argument of the head noun may be attached. Truly adjunctive prepositional phrases are attached to the NP node. This assumption, or something equivalent to it, is necessary to distinguish between the following two sentences:

(1) Mr. Smith saw a driver in his truck.
(2) Mr. Smith saw a driver of his truck.

In (1) 'his' may refer to the driver, but in (2) it may not. The structures we are assuming for the relevant noun phrases in (1) and (2) are shown in figs. 1a and 1b, respectively.



(a) (b)

Fig. 1.

The naive algorithm traverses the surface parse tree in a particular order looking for a noun phrase of the correct gender and number. The traversal order is as follows:

(1) Begin at the NP node immediately dominating the pronoun.

(2) Go up the tree to the first NP or S node encountered. Call this node X, and call the path used to reach it p.

(3) Traverse all branches below node X to the left of path p in a left-to-right, breadth-first fashion. Propose as the antecedent any NP node that is encountered which has an NP or S node between it and X.

(4) If node X is the highest S node in the sentence, traverse the surface parse trees of previous sentences in the text in order of recency, the most recent first; each tree is traversed in a left-to-right, breadth-first manner, and when an NP node is encountered, it is proposed as antecedent. If X is not the highest S node in the sentence, continue to step 5.

(5) From node X, go up the tree to the first NP or S node encountered. Call this new node X, and call the path traversed to reach it p.

(6) If X is an NP node and if the path p to X did not pass through the $\bar{\text{N}}$ node that X immediately dominates, propose X as the antecedent.

(7) Traverse all branches below node X to the *left* of path p in a left-to-right, breadth-first manner. Propose any NP node encountered as the antecedent.

(8) If X is an S node, traverse all branches of node X to the *right* of path p in a left-to-right, breadth-first manner, but do not go below any NP or S node encountered. Propose any NP node encountered as the antecedent.

(9) Go to step 4.

A breadth-first search of a tree is one in which every node of depth n is visited before any node of depth n + 1. Steps 2 and 3 of the algorithm take care of the level in the tree where a reflexive pronoun would be used. Steps 5–9 cycle up the tree through S and NP nodes. Step 4 searches the previous sentences in the text.

For the sake of concreteness, suppose we have the following context-free grammar for generating the surface structures of a fragment of English:

$$\text{S} \rightarrow \text{NP} \quad \text{VP}$$

$$\text{NP} \rightarrow \left\{ \begin{matrix} (\text{Det})\ \bar{\text{N}} \ \left(\left\{ \begin{matrix} \text{PP} \\ \text{Rel} \end{matrix} \right\} \right)^* \\ \text{pronoun} \end{matrix} \right\}$$

$$\text{Det} \rightarrow \left\{ \begin{matrix} \text{article} \\ \text{NP's} \end{matrix} \right\}$$

$\bar{\text{N}} \rightarrow$ noun (PP)*
PP $\rightarrow$ preposition NP
Rel $\rightarrow$ *wh*-word S
VP $\rightarrow$ verb NP (PP)*

Words in lower case letters mean any word of that category; parentheses, (...), indicate optional elements; the asterisk means 0 or more copies of the element that precedes it; braces, {...}, contain alternatives.

Figure 2 illustrates the algorithm working on the sentence

The castle in Camelot remained the residence of the king until 536 when he moved it to London.

Beginning from node NP_1, step 2 rises to node S_1. Step 3 searches the left portion of S_1's tree but finds no *eligible* NP node. Step 4 does not apply.

Step 5 rises to NP_2 which step 6 proposes as antecedent. Thus, '536' is recommended as antecedent of 'it'.

The algorithm can be improved somewhat by applying simple selectional constraints, such as

> Dates can't move;
> Places can't move;
> Large fixed objects can't move.

The utility of these constraints is limited. They never help with the pronoun 'he', since what one male human can do another can too. Even with 'it' the utility is limited since most English words can occur in such a wide variety of contexts. However, in the present example, they help.

After NP_2 is rejected, steps 7 and 8 turn up nothing, and control is returned to step 4 which does not apply. Step 5 rises to S_2, where step 6 does not apply. In step 7, the breadth-first search first suggests NP_3 (the castle), which selectional constraints reject. It then continues to NP_4 where it correctly settles upon 'the residence' as antecedent of 'it'.

If we were searching for the antecedent of 'he', the algorithm would continue, first rejecting NP_5 because of gender and finally lighting upon NP_6, the king.

When seeking an antecedent for 'they', the algorithm accepts plural and collective singular noun phrases and also collects selectionally compatible entities. In

> John sat on the sofa. Mary sat before the fireplace. *They* faced each other.

the algorithm would pick 'Mary' and 'John' rather than 'Mary' and 'the fireplace'. Also, when two plurals are conjoined, the conjunction is favored over either plural, as in

> *Human bones and relics* were found at this site. *They* were associated with elephant tusks.

It should be assumed that the algorithm is part of a larger left-to-right interpretation process which also recovers syntactically recoverable omitted material and records coreference and non-coreference relations. The algorithm then handles the case of Grinder and Postal's 'missing antecedents' (1971). In

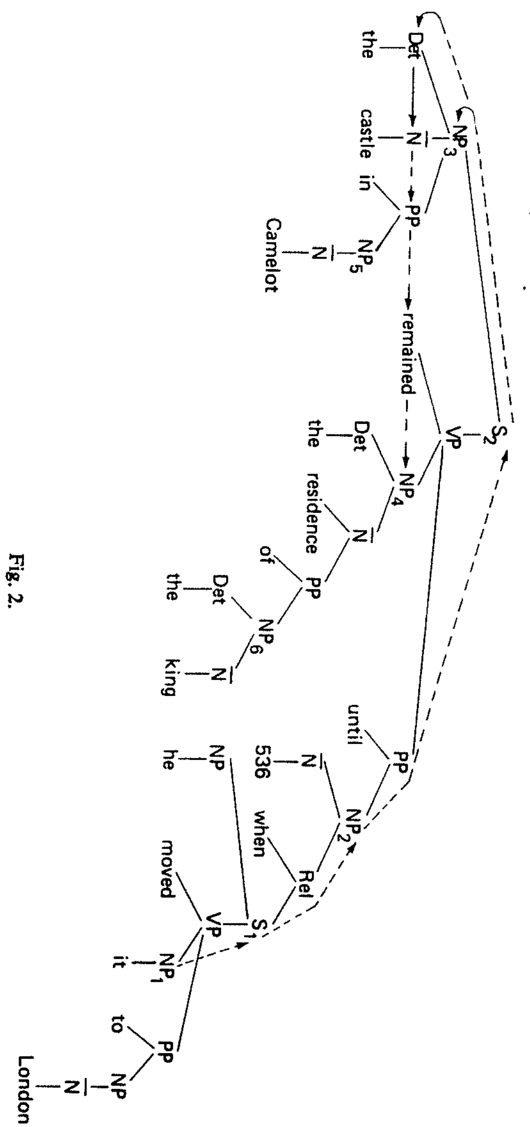


Fig. 2.

My uncle doesn't have a spouse, but your aunt does, and *he* is lying on the floor

the interpretation process first expands the second clause into

... but your aunt does have a spouse ...

and the algorithm then selects the aunt's spouse as antecedent of 'he'. The algorithm also avoids choosing 'the man' as antecedent of 'him' in

John said his mother would sue the man who hit *him*

for 'the man' is necessarily coreferential with the omitted subject of 'hit', which is necessarily non-coreferential with 'him'.

In dialogue it is assumed that the implicit 'A said to B ...' has been recovered before the algorithm is applied to quoted sentences and that rules are available to exclude the speaker and listener as possible antecedents of third person pronouns inside quotes.

The algorithm does not handle sentence pronominalization, as in

Ford was in trouble, and he knew *it*.

One might suggest that the algorithm be modified to accept an S node as the antecedent of a pronoun occurring in certain contexts. However, the problem of avoiding spurious antecedents would then be quite severe. In

The newspaper reported that Ford had claimed the economy was improving, but I didn't believe it

the algorithm allowing both S and NP nodes would recommend the following as plausible antecedents, in the given order:

The newspaper reported that Ford had claimed the economy was improving
the newspaper
Ford claimed the economy was improving
the economy was improving.

This is quite the opposite of one's intuitive feelings about which readings are preferred.

2.2. Relation of the algorithm to results from the generative transformational tradition

A great deal of work has been done in transformational grammar in recent years on stating precisely the conditions under which a noun or non-reflexive pronoun may *not* be coreferential with another element in the sentence. That is, the goal has been to state constraints of the form

A and B are necessarily non-coreferential if ....

This is equivalent to restricting the possible antecedents to a (usually still very large) set of entities. The problem of this paper is how to determine out of this remaining set what in fact really is the antecedent. It is important to be clear about the distinction between these two problems.

The easiest way for us to take the constraints into account would be simply to assume that there is a mechanism which applies them. Then any entity which the naive algorithm proposes is checked by the mechanism, which thus acts as a filter. This in fact is what will be assumed in Part 3. However, here it is shown that the two principal constraints are incorporated within the algorithm itself. There are classes of examples from the literature which are not and could not easily be handled by the algorithm, but they occur rarely in actual texts, and in view of the fact that the algorithm fails on much more natural and common examples, there seems to be little point in greatly complicating the algorithm to handle them.

The first constraint is that a non-reflexive pronoun and its antecedent may not occur in the same simplex sentence (Lees and Klima 1963; Langacker 1969; Jackendoff 1972). In the examples

John likes him.
John's portrait of him

'him' and 'John' cannot be coreferential. However, if an NP node precedes and is on a lower level than the pronoun, it is a possible antecedent, as in

John_i's father's portrait of him_i
After John_i robbed the bank, the police apprehended him_i.

This constraint is accommodated by steps 2 and 3 of the algorithm.

The algorithm fails on the class of picture noun examples, however. In

(3) John saw a picture of him

it would interpret 'him' as John; yet if they were coreferential, 'himself' would have been used. Jackendoff has given an analysis of how reflexives are to be interpreted beyond the scope of simplex sentences. Unfortunately, the corresponding rule for how *non*-reflexives are *not* to be interpreted is incorrect. For there are cases where either the reflexive or non-reflexive pronoun may be used. Consider

*John_1 saw him_1.
?*John_1 saw a picture of him_1.
??John_1 saw a picture of him_1 hanging in the post office.
?John_1 saw that a picture of him_1 was hanging in the post office.
John_1 claimed that the picture of him_1 hanging in the post office was a fraud.

'Himself' is perfectly acceptable in place of 'him' in all five sentences. But apparently the more deeply the pronoun is embedded and the more elaborate the construction it occurs in, the more acceptable the non-reflexive becomes. Yet there is no precise boundary between where it is acceptable and where it is not.

Rather than complicate the algorithm excessively, we will simply let it fail on cases like (3).

The second principal constraint is the rule proposed by Langacker (1969) that the antecedent of a pronoun must precede or command the pronoun. A node NP_1 is said to command node NP_2 if neither NP_1 nor NP_2 dominates the other and if the S node which most immediately dominates NP_1 dominates but does not *immediately* dominate NP_2. The command relation was proposed by Langacker to take care of backward pronominalization examples:

After he_1 robbed the bank, John_1 left town.
That he_1 was elected chairman surprised John_1.

Step 8 of the algorithm, which searches the tree to the right of the pronoun, handles such cases.

There is a search downward in order to include examples like

That he_1 had done something terrible was disturbing to John_1.

But there is no search below S or NP nodes because of the apparent unacceptability of

?That he_1 had done something terrible disturbed John_1's teacher
*That he_1 had done something terrible disturbed the teacher who punished John_1.

This constraint will cause the algorithm to fail on several examples which have been discussed in the literature:

Mary sacked out in his_1 apartment before Sam_1 could kick her out. (Lakoff 1968; Culicover 1976)
Girls who he_1 has dated say that Sam_1 is charming. (Ross 1967)

However, this constraint never caused the algorithm to fail in the sample studied. If it were lifted, the performance of the algorithm would degrade drastically.

2.3. Statistical results

One hundred consecutive examples of pronouns from each of three very different texts were examined to test the performance of the naive algorithm. The pronouns were 'he', 'she', 'it', and 'they'. 'It' was not counted when referring to a syntactically recoverable 'that' clause or occurring in a time or weather construction. In applying the algorithm, it was assumed that the correct parse was available for each sentence. The texts were William Watson's *Early Civilization in China*, pp. 21–69, the first chapter of Arthur Haley's novel *Wheels*, pp. 1–6, and the July 7, 1975 edition of *Newsweek*, pp. 13–19, beginning with the article 'A Ford in High Gear'. The results of the study are summarized in table 1.

Overall, the algorithm worked in 88.3% of the cases. The algorithm together with selectional constraints worked 91.7% of the time. This is somewhat deceptive since in over half the cases there was only one nearby plausible antecedent. For that reason, the number of examples in which more than one plausible antecedent occurred are tabulated, with the number of times the algorithm worked. Of 132 such conflicts, twelve were resolved by selectional constraints and 96 of the remaining 120 were resolved by the algorithm. Thus, 81.8% of the conflicts were resolved by a combination of the algorithm and selection.

If we look at the results for Watson as typical of technical writing, things look even more encouraging. There were a very high number of conflicts, 76 out of 100 examples, but selection resolved eight of these and the algorithm worked on 62 of the remaining 68. The combination thus yields the correct antecedent in 92% of the cases where there is conflict.

Klapholz and Lockman (1975) put forward the hypothesis that the antecedent is always found within the last n sentences, for some small n. Charniak (1972) was more explicit and proposed, with reservations, $n = 5$. This is clearly wrong. However, it is of interest to know how often this heuristic does hold and for what values of n it holds often enough to be useful. Therefore, statistics were also collected on how close the antecedent was to the pronoun. It turns out that it is possible to make a much stronger statement than either Klapholz and Lockman or Charniak suggested. With n 'less than one', a very large majority of the antecedents will be found. Let the *candidate sets* $C_0, C_1, \ldots, C_n$ be defined as follows:

$$C_0 = \begin{cases} \text{the set of entities in the current sentence and the previous sentence} & \text{if pronoun comes before main verb} \\ \text{the set of entities in only the current sentence} & \text{if pronoun comes after main verb.} \end{cases}$$

$$C_1 = \begin{cases} \text{the set of entities in the current sentence and the previous sentence.} \end{cases}$$

$C_n =$ the set of entities in the current sentence and the previous n sentences.

The frequencies with which antecedents were found in $C_0, C_1, \ldots$, is also given in table 1, and they show that the hypothesis is very strong.

Ninety percent of all antecedents are in C_0 while 98% are in C_1. Yet there is no useful absolute limit on how far back one need look for the antecedent. One antecedent occurred nine sentences before the pronoun. The pronoun 'it', especially in technical writing, can have a very large number of plausible antecedents in one sentence – one example in Watson had thirteen. Any absolute limit we impose might therefore have dozens of plausible antecedents and would hardly be of practical value.

Table 1
Summary of the three texts.

	#	C_0	C_1 $-C_0$	C_2 $-C_1$	C_3 $-C_2$	C_9 $-C_8$	Algorithm works	Algorithm works after selection
he	139	126	10	2	0	1	130	130
she	7	7	0	0	0	0	7	7
it	71	64	4	1	2	0	55	59
they	83	74	9	0	0	0	73	79
Total	300	271	23	3	2	1	265	275

	Conflicts before selection	Algorithm correct	Conflicts after selection	Algorithm correct
he	31	22	31	22
she	0	0	0	0
it	48	33	44	33
they	53	43	45	41
Total	132	98	120	96

These results show that the naive approach is quite good. Computationally speaking, it will be a long time before a semantically based algorithm is sophisticated enough to perform as well, and these results set a very high standard for any other approach to aim for.

Yet there is every reason to pursue a semantically based approach. The naive algorithm does not work. Any one can think of examples where it fails. In these cases it not only fails; it gives no indication that it has failed and offers no help in finding the real antecedent.

Moreover, the semantic approach described in Part 3 is all processing that must be done anyway in the analysis of texts.

3. The semantic approach

3.1. A system for semantic analysis of English texts

It is well known that understanding natural language requires a great deal of world knowledge. We will assume this knowledge is available in the form of predicate calculus axioms. Here we describe a system for the semantic analysis of English texts which consists of several 'semantic operations' that draw inferences selectively from the collection of axioms. The inferences drawn are just those which are required to interpret general words in context, determine the relations between sentences, and resolve anaphora.

The input to the semantic analyzer is the text, which we assume syntactic analysis or semantic interpretation rules have already reduced to a logical notation which exhibits functional relationships. In addition, we will assume the syntactically derivable coreference and non-coreference relations have been detected and recorded; any antecedent proposed by the method described below is checked against these relations.

The text in the logical notation consists of (1) a set of entities – X_1, X_2, X_3, ..., representing the entities referred to in the text; (2) the set of kernel statements which describe properties of the entities by applying predicates, corresponding roughly to English words, to the entities; and (3) an indication of which statement in a sentence is asserted and which are grammatically subordinate, by means of the symbol '|', which is read 'such that' or 'where'. For example, the sentence

The boy is on the roof of the building

would be represented (ignoring tense and definite articles)

$$\mathrm{on}(X_1 \mid \mathrm{boy}(X_1),\ X_2 \mid \mathrm{roof}(X_2,\ X_3 \mid \mathrm{building}(X_3))).$$

(The X_1 such that X_1 is a boy is on X_2 where X_2 is the roof of X_3 which is a building.) The information content of this sentence consists of the statements

$$\mathrm{on}(X_1, X_2),\ \mathrm{boy}(X_1),\ \mathrm{roof}(X_2, X_3),\ \mathrm{building}(X_3).$$

In the course of semantic processing, the text is augmented by inferences which the semantic operations determine appropriate, and entities standing for anaphors are merged with the entities standing for their antecedents.

Each of the potentially large number of world knowledge axioms is associated with a particular word, or predicate; hence the collection of axioms will be called the 'Lexicon'. For example, stored with the word 'bank' would be the fact that a bank is a building

(4) $(\forall y)(\mathrm{bank}(y) \supset \mathrm{building}(y))$

and with 'building' the fact that a building has a roof

(5) $(\forall y)(\exists z)(\mathrm{building}(y) \supset \mathrm{roof}(z, y))$.

The general form of the axioms is

(6) $(\forall y)(\exists z)(p(y) \supset (q(y, z) \supset r(y, z)))$

where p is the word or predicate with which the inference is associated, y represents its explicit parameters, z stands for the entities whose existence is also implied, and q(y, z) represents the other enabling conditions which must be checked before the conclusions r(y, z) can be drawn. If $p(X_1)$ occurs in the text and (6) is determined to be appropriate, then $q(X_1, X_i)$ is looked for in the text for some entities X_i, and if it is found, the conclusion $r(X_1, X_i)$ is drawn by adding it to the text.

Inferences are not drawn freely, but only in response to the specific demands of semantic operations. These demands take two forms:

Forward inferences: from $p(X_1)$ try to infer something of the pattern $r(z_1, z_2)$.

Backward inferences: Find something in the text from which $p(X_1)$ could be inferred.

Since the Lexicon is potentially quite large, the axioms are divided into clusters roughly according to topic. The clusters are given an initial measure of salience according to their anticipated relevance to the text at hand. The measures of salience are modified in the course of semantic processing in response to changes in topic in the text in the following way: When an axiom in a cluster is used, the entire cluster is given maximum salience; while the axioms in a cluster are not being used, its salience decays. All searches of the Lexicon initiated by the semantic operations are conducted in cluster order.

It may seem at first glance that this device is purely for efficiency. But in fact it is a crucial part of the analysis mechanism. When an operation calls for a chain of inference, there are usually quite a few chains of inference possible, each of which may lead to a different interpretation or a different antecedent. The changing salience measures on the axioms together with the lengths of the chains of inference define a dynamic ordering on the set of chains of inference. The operation then picks the appropriate chain of inference which is first in this ordering at the time the operation is invoked. In this way the inferencing process and hence the interpretation it produces are made highly context dependent.

3.2. The semantic operations and pronoun resolution

There are four principal semantic operations:

(1) *Detecting intersentence connectives:* The implicit relations between sentences in a paragraph are detected by comparing the current sentence and a previous sentence against a small set of common patterns. These patterns are stated in terms of inferences to be drawn from the two sentences. The patterns include Contrast, Cause, Violated Expectation, Temporal Succession, Paraphrase, Parallel, and Example. Only the first three are relevant to the examples below and are described here.[2]

Explicit conjunctions cause certain patterns to be strongly preferred – 'and' promotes Temporal Succession and Parallel, 'because' promotes Cause, a dash and 'i.e.' promote Paraphrase, and 'but' promotes Contrast and Violated Expectation.

One variety of the Contrast pattern might be stated as follows:

From the assertions of the current sentence and a previous sentence, try to infer statements S_1 and S_2, respectively, where
(a) the predicates of S_1 and S_2 are contradictory or lie at opposite ends of some scale;
(b) one pair of corresponding arguments of S_1 and S_2 are identical;
(c) the other pairs of corresponding arguments are 'similar' but different.

The Violated Expectation pattern is matched if from the assertion of the previous sentence a statement S can be inferred, while from the assertion of the current sentence, not-S is inferred. The latter inference is drawn and the former is suppressed.

[2] All the patterns are described in Hobbs (1976b).

Among the axioms in the Lexicon will be axioms stating our knowledge about causes and effects. A strong form of the Cause pattern is

Find a causal chain from the purported cause to the purported effect.

A weaker form is

(7) Find a causal chain from a prominent fact inferable from the purported cause to a prominent fact inferable from the purported effect.

Here in a sense we are required to establish the causal link between supersets containing the items rather than between the items themselves. This allows us to recognize as having valid causal links, texts which are not just instantiations of potential theorems in the system.

(2) *Predicate interpretation:* It is typical for the most common English words to be useable in a wide variety of contexts. In large measure, this is possible because they are defined not so much in terms of the inferences that can be drawn from them, but in terms of their effect on the inferences that are drawn from the words with which they are grammatically related. For example, normally from the use of the word 'horse' one can infer certain facts about the entity described, such as that a typical activity might be racing and that it moves within a certain range of speed when it races. However, when we encounter 'slow horse', the word 'slow' alters what we may infer about the horse's speed when racing. A general definition for 'slow' will first instruct us to find the most prominent motion associated with the entity it modifies (and here the notion of context-dependent salience comes in). It then qualifies what we may infer about the range of speed of this motion. Such a definition will work for not only 'slow horse', but also 'slow walk', 'slow race', and 'slow watch', and if we are allowed to interpret 'motion' metaphorically, it also works for 'slow student' and 'slow business'.

This suggests a method for the interpretation of general words or predicates in context. A definition consists of two parts. The first part specifies the demands the predicate makes on its arguments, expressed in terms of inferences to be drawn from the explicit properties of the arguments. The second part specifies how these inferences are to be modified, or what further information is provided by the predicate. The operation which searches the Lexicon for these inferences and draws them with the appropriate modifications we will call 'predicate interpretation'.

Consider two more words which are used in examples below:

(a) 'Into' is a two-argument predicate. We must verify that its first argument (what the prepositional phrase modifies) is a motion and that its second argument (the object of the preposition) is a real or metaphorical region. We may then infer that the endpoint of the motion is within the region.

(b) For the predicate 'reduce', we must first locate the most salient scale with a real or metaphorical vertical orientation associated with its object. We can then infer that a downward motion occurs on this scale.

The operation of predicate interpretation allows us to recover omitted material such as *the part from the whole:*

He landed on (the roof of) the building

because 'on' demands a surface for its object; and *missing quantity words:*

(The price of) Coffee is higher this month

since 'higher' requires a real or metaphorical vertical scale.

This operation has a default feature. If a proof of the required inference can't be found, the inference is drawn anyway. At worst it will be instantiated with new and unknown entities for its arguments. The next operation will normally correct for this.

Predicate interpretation frequently aids in pronoun resolution, as will be seen in the next few paragraphs, and in Section 4.1.

(3) *Knitting:* When a statement is instantiated whose predicate is identical to that of a statement already in the text, then the first guess we make is that this is a redundancy. That is, redundancy is expected to be the normal state of affairs. If no obvious inconsistency is found, we assume the two statements are the same, and we merge them, thereby merging the corresponding arguments.

Knitting frequently leads to the antecedents of pronouns and implicit entities being found before operation 4, whose work it is to find antecedents, is even called. For example, suppose the following text is being processed:

(8) The boy walked into the bank. Moments later he was seen on its roof.

Let the bank be represented by the entity X_1 and the 'it" of 'its' by X_2. Predicate interpretation on 'into' forces us to verify that its object, X_1, is an enclosure. This could be done by using the fact (4) that a bank is a building and the fact that a building is an enclosure. Since the entire chain of inference is instantiated the statement

(9) building(X_1)

becomes part of the text. The second sentence is processed, and predicate interpretation on 'roof' demands verification that X_2 is a building. This cannot be verified by the other properties of X_2 since it has no other properties. Therefore the default feature applies and

(10) building(X_2)

is simply assumed. The redundancy of (9) and (10) is presumed, knitting occurs, and the two statements and thereby their corresponding arguments X_1 and X_2 are merged. Thus has the antecedent of 'it' been identified as 'the bank'.

(4) *Identifying entities:* This operation seeks to identify the so far unidentified entities. Entities referred to in a text may be arranged in a hierarchy according to their degree of specification:

(a) proper names, including "you" and "I";
(b) other noun phrases, including those with definite, indefinite, and demonstrative articles;
(c) third person pronouns;
(d) zeroed arguments and implicit entities.

When a proper noun is encountered, it is identified with any entity in the previous text described by the same proper noun, or if there is none, a new entity is introduced. When a noun phrase tagged by the indefinite article is encountered, a new entity is introduced. The identification procedure for definite noun phrases is described in Hobbs (1975, 1976b). Its search step is similar to the search step described below for pronouns.

In order to find the antecedent of a pronoun a backward search through the Lexicon is conducted for a chain of inference that begins at some statement in the text and ends with a known property of the pronoun. Suppose example (8) had escaped the first three operations, and a search for the antecedent of 'it' was necessary. The only known property of X_2 is that it has a roof. The Lexicon is probed to see what has a roof, the fact

(5) stored with 'building' that buildings have roofs is found, and the text is checked for an occurrence of 'building'. Assume 'building(X_1)' has not yet been inferred from 'bank(X_1)', so that 'building' is not found. The Lexicon is then searched for something which is a building, the fact (4) associated with 'bank' that a bank is a building is found, and a bank is mentioned explicitly in the text. The required chain of inference

$$\text{bank}(X_1) \supset \text{building}(X_1) \supset \text{roof}(X_3, X_1)$$

is found. Hence the antecedent of 'it' is located.

The difficulty with this search is that it is very expensive. It requires exponential time and the branching factor of the search can be very large. For example, there could be a great many axioms in the Lexicon which say that something is a building; gas stations, post offices, dime stores, etc. are all buildings. Therefore in order to cut down on the size of the search, and at the same time to take advantage of the effectiveness of the naive algorithm of Part 2, a bidirectional search (Pohl 1971) is used which starts not only at the pronoun but also at the entity the naive algorithm would choose as antecedent, in hopes that the two searches will meet somewhere in the middle. Thus, in about 90% of the cases, the search will go quite fast.

Once a plausible antecedent is found, a shallow check is made to insure that the properties of the two entities to be merged are not obviously inconsistent.

More than one plausible antecedent may be found by the search. If so, we choose the candidate that maximizes the redundancy in the simplest possible way, by inferring freely from the properties of the candidates and of the pronoun and picking the candidate that has the most properties of high salience in common with the pronoun.

4. Examples of the semantic approach to pronoun resolution

4.1.

The first example is from the archaeology text (Watson 1966: 21) and illustrates knitting working in conjunction with predicate interpretation:

The plain was reduced by erosion to *its* present level.

or, in the logical notation,

$$\text{reduce}(\text{erode}(X_1), X_2 \mid \text{plain}(X_2), X_3 \mid \text{present}(\text{level}(X_3, X_4)))$$

i.e. something, X_1, eroding reduces X_2 which is a plain to X_3 where X_3's being the level of X_4 is true at present. We must identify the antecedent not only if 'it' (X_4) but also of the implicit entity which is eroding (X_1).

Note that syntactic criteria do not solve the problem, for in the sentence

Walter was introduced by John to his present wife

'his' could refer to Walter, John, or someone else. Selectional criteria will not work either, for 'erosion' can have a present level, as in

Contour farming has reduced erosion to its present level.

Consider now what happens in the course of semantic processing. We apply predicate interpretation to 'reduce'. This predicate demands of its second argument that it be capable of movement along some vertical axis (real or metaphorical), leading us to infer that a plain, being a land form, is characterized by an altitude, i.e. a position on the real vertical axis going from the center of the earth outward. 'Reduce' then adds the information that a change downward to the third argument X_3 has occurred on this axis:

(11) $\text{become}(\text{at}(X_2, X_5), \text{at}(X_2, X_3)) \mid \text{exceed}(X_5, X_3, X_7 \mid \text{Altitude-axis}(X_7)), \text{vertical}(X_7)$

i.e. the plain X_2 at X_5 becomes X_2 at X_3 where X_5 exceeds X_3 on the vertical Altitude axis X_7.

Next the arguments of 'reduce' are processed in turn. From the argument X_1 of 'erode' we must also infer that it is capable of movement along a real or metaphorical vertical axis. 'Erode' also says this movement is in a downward direction. X_1 has no explicit properties, so we cannot infer a vertical axis. Hence we simply assume one to exist.

(12) $\text{become}(\text{at}(X_1, X_8), \text{at}(X_1, X_9)) \mid \text{exceed}(X_8, X_9, X_{10} \mid \text{vertical}(X_{10}))$

Since (11) and (12) are identical except for temporary entities, and since no contradiction could be derived if we identified the temporary entities, the knitting operation applies, and the implicit entity which is eroding, X_1, is identified with the plain, X_2, X_8 is identified with X_5, X_9 with X_3, and the vertical axis X_{10} with the Altitude axis X_7.

When the third argument of 'reduce' is processed, we first apply predicate interpretation to 'present'. 'Present' carries with it the implication that what it describes – X_3 being the level of X_4 – resulted by a 'becoming' from some previous state. 'Level' demands that its first argument be a point on a vertical scale, and that its second argument be at that point. Thus, we infer

$$\text{become}(X_{11}, \text{at}(X_4, X_3 \mid \text{on}(X_3, X_{12} \mid \text{vertical}(X_{12}))))$$

i.e. the state X_{11} changes into the state in which X_4 is located at X_3 which is a point on a vertical scale X_{12}. Knitting identifies this with (11), thereby identifying X_{11} with '$\text{at}(X_2, X_5)$', X_4 ('it') with X_2 ('the plain'), and vertical axis X_{12} with the Altitude axis X_7.

When at last we invoke operation 4, we find that all entities have been identified except the anaphoric definite noun phrase 'the plain'.

4.2.

The second example comes from *Newsweek* and illustrates how the intersentence operation aids in pronoun resolution.

> The FBI said they had tentative identifications on the fugitives, but didn't know where *they* were.

We wish to find the antecedent of the 'they' in the 'but' clause.

The naive algorithm does not work on this example. The first entity it would propose is the omitted subject of 'didn't know', i.e. the FBI, as if the FBI were lost in a forest and didn't know where they were. Next it would light upon 'tentative identifications', as if the FBI had identifications but a clerk had misfiled them. Only at last would the correct antecedent, 'fugitives', be reached.

In examining the semantic approach, assume we have the correct parse, conjoining the 'but' clause with 'they had tentative identifications ...'. Assume also that the 'they' of the first clause and the omitted subject of the second have been recognized as coreferential with 'the FBI'. The conjunction 'but' makes the Contrast pattern strongly preferred. From 'the FBI had tentative identifications on the fugitives' we can infer 'the FBI had tentatively identified the fugitives'. From this can be inferred 'the FBI (tentatively) knows the names of the fugitives'. This is compared with the assertion of the 'but' clause, which paraphrased is 'the FBI does not know the location of X'. We find that the predicates are contradictory, and the first arguments are the same. Therefore the Contrast pattern will be matched if the second arguments – 'the names of the fugitives' and 'the location of X' – can be shown to be similar. This can be accomplished by assuming 'they' and 'the fugitives' to be the same.

The search step, operation 4, would locate the antecedent, even if the first three operations failed. The following chain of inference would be discovered:

(13) From '$\text{fugitive}(X_1)$' infer '$\text{hide-from}(X_1, X_2 \mid \text{police}(X_2))$'
(14) From (13) infer '$\text{cause}(X_1, \text{not}(\text{know}\ (X_2, \text{location}(X_1))))$'.
From (14) infer '$\text{not}(\text{know}(X_2, \text{location}(X_1)))$'.

(If something is caused, it holds.) But this is just the property we know about 'they'.

4.3.

Next is the classic example from Winograd (1972: 33). Consider

(15) They_1 prohibited them_2 from demonstrating because they_3 feared violence.
(16) They_1 prohibited them_2 from demonstrating because they_3 advocated violence.

'They_3' is coreferential with 'they_1' in (15), but with 'them_2' in (16).

In (15), the intersentence operation will seek to link the two clauses, and on account of the conjunction 'because' a match with a Cause pattern will be the most sought.

For the sake of this discussion we will coin a word 'diswant', analogous to 'dislike'. To diswant S is to want not-S. From 'x prohibits y' we can infer 'x diswants y' and from 'x fears z' we can infer 'x diswants z'. We will use (7) and try to establish a causal link from 'they_3 diswant violence' to 'they_1 diswant (they_2 demonstrate)'.

A prominent fact about demonstrations, well known to anyone aware in the 'sixties, is that frequently

(17) (w demonstrate) cause violence.

A fundamental fact relating the real world with mental and emotional worlds is

(18) (x cause y & z diswant y) cause (z diswant x).

This has the interesting effect of transforming a causal link between two events in the real world into the reversed causal link between mental states in which these events are apprehended. Because of (17) and (18) '$they_3$ diswant violence' causes '$they_3$ diswant (w demonstrate)', so that the link is established if we identify '$they_3$' with '$they_1$' and w with '$they_2$'.

To match the Cause pattern in (16) we seek a causal chain from the second clause to the statement inferable from the first clause that '$they_1$ diswant ($they_2$ demonstrate)'. The chain is as follows: Someone advocating something often causes that something to occur. In particular, $they_3$ advocating violence may cause violence to occur. Normally, someone, in particular $they_1$, will diswant violence. Therefore, by (18), $they_1$ will diswant $they_3$ advocating violence. Since '$they_3$ demonstrate' causes '$they_3$ advocate x', again by (18) we infer '$they_1$ diswant ($they_3$ demonstrate)'. The pattern is then matched by identifying '$they_3$' and '$they_2$'.

The pronoun problem of (15) and (16) could also be solved by proposing both '$they_1$' and '$them_2$' as plausible antecedents and holding a competition to see which choice maximized redundancy. In (15), the redundancy between '$they_1$ diswant ...' and '$they_3$ diswant ...' would result in '$they_1$' being chosen. In (16), the prominent fact about demonstrating that someone who demonstrates advocates something would make '$them_2$' the choice that maximized redundancy.

4.4.

Last is the classic example from Charniak (1972, 1974), in the style of a children's story:

Jack invited Janet to his birthday party.
Janet wondered if Jack would like a kite.
But Bill said Jack already had a kite.
Jack would make her take it back.

The question is, how do we know 'it' refers to the kite she is thinking about buying and not the more recently mentioned kite Jack already has.

This is a difficult text, but it is important to be clear about where the difficulty lies. For it does not lie in the pronoun resolution problem in the last sentence. In part, it lies in determining the relation between the first two sentences. And this difficulty is reflected in a slight but perceptible discontinuity the reader senses at that point.

The relation between the two sentences is causal. Establishing a causal chain from the first to the second might go as follows: Jack inviting Janet to the birthday party causes her to want to come to it. (19), a 'want' version of (18), may be proposed as a general rule and is applied here several times.

(19) If y enables or is required for z, then x wanting z causes x to want y.

A guest at a birthday party is required to give the host a present. To be a present something should be new and it should be liked by the recipient. For it to be new one has to buy it. Now if Jack would like a kite and the present were a kite, then Jack would like the present. Therefore her wanting Jack to like the present causes her to want to know (i.e. wonder) if Jack would like her to buy a kite.

Regardless of the particular facts used and the particular causal chain found, we must somehow flesh out the second sentence to something like

Janet wondered if Jack would like the kite if she bought one for him for a birthday present.

Once this is accomplished, pronoun resolution becomes a straightforward matter. We simply take the word 'back' seriously.[3] For there to be motion *back* there must have been motion *to*. Therefore, in performing predicate interpretation on 'back', we look for the most salient motion in the previous text, preferably involving the same agent and object as the motion modified by 'back'. We find the motion of Janet's buying the kite, and therefore identify 'it' with the kite she considers buying.

The intersentence operation is also relevant. The Violated Expectation

[3] Charniak (1974) has also pointed this out.

pattern between the second and third sentences, signalled by 'but', is verified by noting that the expectation that Jack would like the kite is violated. From this the unacceptability of the kite as a gift can be inferred, and this is the cause whose effect is the return of Janet's kite in the fourth sentence.

A minor aspect that contributes to Janet's kite being chosen as antecedent is the fact that the kite exists in the hypothetical world of Janet's wondering. Jack's kite exists in the real world. 'It' occurs in the hypothetical world dominated by 'would'. Thus, maximum redundancy is served if we assume 'it' to be the hypothetical kite.

5. Summary

We have proposed three steps for pronoun resolution.

(1) The intersentence relation operation together with knitting.

(2) Predicate interpretation together with knitting.

(3) The bidirectional search through the Lexicon, using the naive algorithm.

To these two more steps can be added as a fail-safe mechanism:

(4) For all the entities of the correct gender in the current sentence and the previous sentence, hold a competition to maximize the redundancy as in Operation 4 of Section 3.2. At least we know the antecedent is there in 98% of the cases. Heretofore, this has been the best solution offered for the pronoun resolution problem.

(5) Apply the naive algorithm. This has the advantage that it always gives an answer.

Normally, pronoun resolution will be accomplished by the first two steps. On reflection, this should not be surprising. For the reason a speaker uses a pronoun is precisely because the identity of the entity is obvious without description to anyone who is understanding what is being said, and the first two semantic operations capture, in large measure, what is meant by understanding.

References

Charniak, E., 1972. Toward a model of children's story comprehension. AI TR-266, Massachusetts Institute of Technology Artificial Intelligence Laboratory.

Charniak, E., 1974. 'He will make you take it back': A study in the pragmatics of language. Working Paper 5, Institute for Semantic and Cognitive Studies.

Chomsky, N., 1970. Remarks on nominalization. In: R. Jacobs and P. Rosenbaum (eds.), Readings in transformational grammar, 184–221. Waltham, Mass.: Blaisdell.

Culicover, P., 1976. A constraint on coreferentiality. Foundations of Language 14, 109–118.

Grinder, J., and P. Postal, 1971. Missing antecedents. Linguistic Inquiry 2, 269–312.

Hobbs, J., 1975. A general system for semantic analysis of English and its use in drawing maps from directions. American Journal of Computational Linguistics, microfiche 32.

Hobbs, J., 1976a. Pronoun resolution. Res. Rep. 76-1, Dept. Comp. Sci., City College, CUNY, July 1976.

Hobbs, J., 1976b. A computational approach to discourse analysis. Res. Rep. 76-2, Dept. Comp. Sci., City College, CUNY, December 1976.

Jackendoff, R., 1972. Semantic interpretation in generative grammar. Cambridge, Mass.: MIT Press.

Jespersen, O., 1954. A modern English grammar on historical principles, part VII: Syntax. London: Allen and Unwin.

Klapholtz, D., and A. Lockman, 1975. Contextual reference resolution. American Journal of Computational Linguistics, microfiche 36.

Lakoff, G., 1968. Pronouns and reference. Unpublished paper.

Langacker, R., 1969. On pronominalization and the chain of command. In: D. Reibel and S. Schane (eds.), Modern studies in English, 160–186. Englewood Cliffs: Prentice-Hall.

Lees, R., and E. Klima, 1963. Rules for English pronominalization. Language 39, 17–28.

Olson, D., 1970. Language and thought: Aspects of a cognitive theory of semantics. Psychological Review 77, 257–273.

Pohl, I., 1971. Bi-directional search. In: B. Meltzer and D. Michie (eds.), Machine intelligence, vol. 6, 127–140. New York: American Elsevier.

Rieger, C., 1974. Conceptual memory: A theory and computer program for processing the meaning content of natural language utterances. Stanford AIM-233.

Ross, J., 1967. On the cyclic nature of English pronominalization. In: To honor Roman Jakobson, 1669–1682. The Hague: Mouton.

Watson, W., 1966. Early civilization in China. New York: McGraw-Hill.

Wilks, Y., 1975. A preferential, pattern-seeking, semantics for natural language inference. Artificial Intelligence 6, 53–74.

Winograd, T., 1972. Understanding natural language. New York: Academic Press.

Readings In Natural Language Processing

edited by

Barbara J. Grosz
(SRI International)

Karen Sparck Jones
(University of Cambridge, Computer Laboratory)

Bonnie Lynn Webber
(University of Pennsylvania)

Morgan Kaufmann Publishers, Inc.
95 First Street, Los Altos, California 94022