INCREMENTAL INTERPRETATION: APPLICATIONS, THEORY, AND RELATIONSHIP TO DYNAMIC SEMANTICS*

David Milward & Robin Cooper

Centre for Cognitive Science, University of Edinburgh
2, Buccleuch Place, Edinburgh, EH8 9LW, Scotland, davidm@cogsci.ed.ac.uk

ABSTRACT

Why should computers interpret language incrementally? In recent years psycholinguistic evidence for incremental interpretation has become more and more compelling, suggesting that humans perform semantic interpretation before constituent boundaries, possibly word by word. However, possible computational applications have received less attention. In this paper we consider various potential applications, in particular graphical interaction and dialogue. We then review the theoretical and computational tools available for mapping from fragments of sentences to fully scoped semantic representations. Finally, we tease apart the relationship between dynamic semantics and incremental interpretation.

APPLICATIONS

Following the work of, for example, Marslen-Wilson (1973), Just and Carpenter (1980) and Altmann and Steedman (1988), it has become widely accepted that semantic interpretation in human sentence processing can occur before sentence boundaries and even before clausal boundaries. It is less widely accepted that there is a need for incremental interpretation in computational applications.

In the 1970s and early 1980s several computational implementations motivated the use of incremental interpretation as a way of dealing with structural and lexical ambiguity (a survey is given in Haddock 1989). A sentence such as the following has 4862 different syntactic parses due solely to attachment ambiguity (Stabler 1991).

1) I put the bouquet of flowers that you gave me for Mothers' Day in the vase that you gave me for my birthday on the chest of drawers that you gave me for Armistice Day.

Although some of the parses can be ruled out using structural preferences during parsing (such as Late Closure or Minimal Attachment (Frazier 1979)), extraction of the correct set of plausible readings requires use of real world knowledge. Incremental interpretation allows on-line semantic filtering, i.e. parses of initial fragments which have an implausible or anomalous interpretation are rejected, thereby preventing ambiguities from multiplying as the parse proceeds.

However, on-line semantic filtering for sentence processing does have drawbacks. Firstly, for sentence processing using a serial architecture (rather than one in which syntactic and semantic processing is performed in parallel), the savings in computation obtained from on-line filtering have to be balanced against the additional costs of performing semantic computations for parses of fragments which would eventually be ruled out anyway from purely syntactic considerations. Moreover, there are now relatively sophisticated ways of packing ambiguities during parsing (e.g. by the use of graph-structured stacks and packed parse forests (Tomita 1985)). Secondly, the task of judging plausibility or anomaly according to context and real world knowledge is a difficult problem, except in some very limited domains. In contrast, statistical techniques using lexeme co-occurrence provide a relatively simple mechanism which can imitate semantic filtering in many cases. For example, instead of judging *bank* as a financial institution as more plausible than *bank* as a riverbank in the noun phrase *the rich bank*, we can compare the number of co-occurrences of the lexemes **rich** and $\mathbf{bank}_1$ (= riverbank) versus **rich** and $\mathbf{bank}_2$ (= financial institution) in a semantically analysed corpus. Cases where statistical techniques seem less appropriate are where plausibility is affected by local context. For example, consider the ambiguous sentence, *The decorators painted a wall with cracks* in the two contexts *The room was supposed to look rundown* vs. *The clients couldn't afford wallpaper.* Such cases involve reasoning with an interpretation in its immediate context, as opposed to purely judging the likelihood of a particular linguistic expression in a given application domain (see e.g. Cooper 1993 for discussion).

Although the usefulness of on-line semantic filtering during the processing of complete sentences is debatable, filtering has a more plausible role to play in interactive, real-time environments, such as interactive spell checkers (see e.g. Wirén (1990) for arguments for incremental parsing in such environments). Here the choice is between whether or not to have semantic filtering at all, rather than whether to do it on-line, or at the end of the sentence.

The concentration in early literature on using incremental interpretation for semantic filtering has perhaps distracted from some other applications which provide less controversial applications. We will

*This research was supported by the UK Science and Engineering Research Council, Research Grant RR30718.

consider two in detail here: graphical interfaces, and dialogue.

The Foundations for Intelligent Graphics Project (FIG)[1] considered various ways in which natural language input could be used within computer aided design systems (the particular application studied was computer aided kitchen design, where users would not necessarily be professional designers). Incremental interpretation was considered to be useful in enabling immediate visual feedback. Visual feedback could be used to provide confirmation (for example, by highlighting an object referred to by a successful definite description), or it could be used to give the user an improved chance of achieving successful reference. For example, if sets of possible referents for a definite noun phrase are highlighted during word by word processing then the user knows how much or how little information is required for successful reference.[2]

Human dialogue, in particular, task oriented dialogue is characterised by a large numbers of self-repairs (Levelt 1983, Carletta et al. 1993), such as hesitations, insertions, and replacements. It is also common to find interruptions requesting extra clarification, or disagreements before the end of a sentence. It is even possible for sentences started by one dialogue participant to be finished by another. Applications involving the understanding of dialogues include information extraction from conversational databases, or computer monitoring of conversations. It also may be useful to include some features of human dialogue in man-machine dialogue. For example, interruptions can be used for early signalling of errors and ambiguities.

Let us first consider some examples of self-repair. Insertions add extra information, usually modifiers e.g.

2) We start in the middle with .... in the middle of the paper with a blue disc (Levelt 1983:ex.3)

Replacements correct pieces of information e.g.

3) Go from left again to uh ..., from pink again to blue (Levelt 1983:ex.2)

In some cases information from the corrected material is incorporated into the final message. For example, consider[3] :

4) a The three main sources of data come, uh ..., they can be found in the references

b John noticed that the old man and his wife, uh ..., that the man got into the car and the wife was with him when they left the house

c Every boy took, uh ..., he should have taken a water bottle with him

In (a), the corrected material *the three main sources of data come* provides the antecedent for the pronoun *they*. In (b) the corrected material tells us that the man is both old and has a wife. In (c), the pronoun *he* is bound by the quantifier *every boy*.

For a system to understand dialogues involving self-repairs such as those in (4) would seem to require either an ability to interpret incrementally, or the use of a grammar which includes self repair as a syntactic construction akin to non-constituent coordination (the relationship between coordination and self-correction is noted by Levelt (1983)). For a system to generate self repairs might also require incremental interpretation, assuming a process where the system performs on-line monitoring of its output (akin to Levelt's model of the human self-repair mechanism). It has been suggested that generation of self repairs is useful in cases where there are severe time constraints, or where there is rapidly changing background information (Carletta, p.c.).

A more compelling argument for incremental interpretation is provided by considering dialogues involving interruptions. Consider the following dialogue from the TRAINS corpus (Gross et al., 1993):

5) **A**: so we should move the engine at Avon, engine E, to ...
B: engine E1
A: E1
B: okay
A: engine E1, to Bath ...

This requires interpretation by speaker B before the end of A's sentence to allow objection to the apposition, *the engine at Avon, engine E*. An example of the potential use of interruptions in human computer interaction is the following:

6) **User**: Put the punch onto ...
Computer: The punch can't be moved. It's bolted to the floor.

In this example, interpretation must not only be before the end of the sentence, but before a constituent boundary (the verb phrase in the user's command has not yet been completed).

CURRENT TOOLS

1. Syntax to Semantic Representation

In this section we shall briefly review work on providing semantic representations (e.g. lambda expressions) word by word. Traditional layered models of sentence processing first build a full syntax tree for a sentence, and then extract a semantic representation from this. To adapt this to an incremental perspective, we need to be able to provide syntactic structures

---

[1] Joint Councils Initiative in Cognitive Science/HCI, Grant 8826213, EdCAAD and Centre for Cognitive Science, University of Edinburgh.

[2] This example was inspired by the work of Haddock (1987) on incremental interpretation of definite noun phrases. Haddock used an incremental constraint based approach following Mellish (1985) to provide an explanation of why it is possible to use the noun phrase *the rabbit in the hat* even when there are two hats, but only one hat with a rabbit in it.

[3] Example (a) is reconstructed from an actual utterance. Examples (b) and (c) were constructed.

(of some sort) for fragments of sentences, and be able to extract semantic representations from these.

One possibility, which has been explored mainly within the Categorial Grammar tradition (e.g. Steedman 1988) is to provide a grammar which can treat most if not all initial fragments as constituents. They then have full syntax trees from which the semantics can be calculated.

However, an alternative possibility is to directly link the partial syntax trees which can be formed for non-constituents with functional semantic representations. For example, a fragment missing a noun phrase such as *John likes* can be associated with a semantics which is a function from entities to truth values. Hence, the partial syntax tree given in Fig. 1[4],

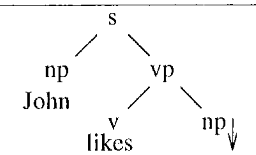

Fig. 1

can be associated with a semantic representation, **λx. likes(john,x)**.

Both Categorial approaches to incremental interpretation and approaches which use partial syntax trees get into difficulty in cases of left recursion. Consider the sentence fragment, *Mary thinks John*. A possible partial syntax tree is provided by Fig. 2.

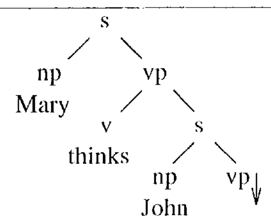

Fig. 2

However, this is not the only possible partial tree. In fact there are infinitely many different trees possible. The completed sentence may have an arbitrarily large number of intermediate nodes between the lower s node and the lower **np**. For example, **John** could be embedded within a gerund e.g. *Mary thinks John leaving here was a mistake*, and this in turn could be embedded e.g. *Mary thinks John leaving here being a mistake is surprising*. **John** could also be embedded within a sentence which has a sentence modifier requiring its own s node e.g. *Mary thinks John will go home probably*[5], and this can be further embedded e.g. *Mary thinks John will go home probably because he is tired*.

The problem of there being an arbitrary number of different partial trees for a particular fragment is reflected in most current approaches to incremental interpretation being either incomplete, or not fully word by word. For example, incomplete parsers have been proposed by Stabler (1991) and Moortgat (1988). Stabler's system is a simple top-down parser which does not deal with left recursive grammars. Moortgat's M-System is based on the Lambek Calculus: the problem of an infinite number of possible tree fragments is replaced by a corresponding problem of initial fragments having an infinite number of possible types. A complete incremental parser, which is not fully word by word, was proposed by Pulman (1986). This is based on arc-eager left-corner parsing (see e.g. Resnik 1992).

To enable complete, fully word by word parsing requires a way of encoding an infinite number of partial trees. There are several possibilities. The first is to use a language describing trees where we can express the fact that *John* is dominated by the s node, but do not have to specify what it is immediately dominated by (e.g. D-Theory, Marcus et al. 1983). Semantic representations could be formed word by word by extracting 'default' syntax trees (by strengthening dominance links into immediated dominance links wherever possible).

A second possibility is to factor out recursive structures from a grammar. Thompson et al. (1991) show how this can be done for a phrase structure grammar (creating an equivalent Tree Adjoining Grammar (Joshi 1987)). The parser for the resulting grammar allows linear parsing for an (infinitely) parallel system, with the absorption of each word performed in constant time. At each choice point, there are only a finite number of possible new partial TAG trees (the TAG trees represents the possibly infinite number of trees which can be formed using adjunction). It should again be possible to extract 'default' semantic values, by taking the semantics from the TAG tree (i.e. by assuming that there are to be no adjunctions). A somewhat similar system has recently been proposed by Shieber and Johnson (1993).

The third possibility is suggested by considering the semantic representations which are appropriate during a word by word parse. Although there are any number of different partial trees for the fragment *Mary thinks John*, the semantics of the fragment can be represented using just two lambda expressions[6]:

λP. thinks(mary,P(john))
λP. λQ. Q(thinks(mary,P(john)))

Consider the first. The lambda abstraction (over a

[4] The downarrow notation for missing constituents is adopted from Synchronous Tree Adjoining Grammar (Shieber & Schabes 1990).

[5] The treatment of **probably** as a modifier of a sentence is perhaps controversial. However, treatment of it as a verb phrase modifier would merely shift the potential left recursion to the verb phrase node.

[6] Two representations are appropriate if there are no VP-modifiers as in dependency grammar. If VP-modification is allowed, two more expressions are required:
λP. λR. (R(λx.thinks(mary,x)))(P(john)) and
λP. λR. λQ Q((R(λx.thinks(mary,x)))(P(john))).

functional item of type e→t) can be thought of as a way of encoding an infinite set of partial semantic (tree) structures. For example, the eventual semantic structure may embed **john** at any depth e.g.

thinks(mary,sleeps(john))
thinks(mary,possibly(sleeps(john)))
etc.

The second expression (a functional item over type e→t and t→t), allows for eventual structures where the main sentence is embedded e.g.

possibly(thinks(mary,sleeps(john)))

This third possibility is therefore to provide a syntactic correlate of lambda expressions. In practice, however, provided we are only interested in mapping from a string of words to a semantic representation, and don't need explicit syntax trees to be constructed, we can merely use the types of the 'syntactic lambda expressions', rather than the expressions themselves. This is essentially the approach taken in Milward (1992) in order to provide complete, word by word, incremental interpretation using simple lexicalised grammars, such as a lexicalised version of formal dependency grammar and simple categorial grammar[7].

2. Logical Forms to Semantic Filtering

In processing the sentence *Mary introduced John to Susan*, a word-by-word approach such as Milward (1992) provides the following logical forms after the corresponding sentence fragments are absorbed:

Mary	λP.P(mary)
Mary introduced	λx.λy.intr(mary,x,y)
Mary introduced John	λy.intr(mary,john,y)
Mary introduced John to	λy.intr(mary,john,y)
Mary introduced John to Sue	intr(mary,john,sue)

Each input level representation is appropriate for the meaning of an incomplete sentence, being either a proposition or a function into a proposition.

In Chater et al. (1994) it is argued that the incrementally derived meanings are not judged for plausibility directly, but instead are first turned into existentially quantified propositions. For example, instead of judging the plausibility of **λx.λy.intr(mary,x,y)**, we judge the plausibility of **∃(x,T,∃(y,T,intr(mary,x,y)))**[8]. This is just the proposition *Mary introduced something to something* using a generalized quantifier notation of the form **Quantifier(Variable,Restrictor,Body)**.

Although the lambda expressions are built up monotonically, word by word, the propositions formed from them may need to be retracted, along with all the resulting inferences. For example, *Mary introduced something to something* is inappropriate if the final sentence is *Mary introduced noone to anybody*. A rough algorithm is as follows:

1. Parse a new word, Word_i
2. Form a new lambda expression by combining the lambda expression formed after parsing Word_{i-1} with the lexical semantics for Word_i
3. Form a proposition, P_i, by existentially quantifying over the lambda abstracted variables.
4. Assert P_i. If P_i does not entail P_{i-1} retract P_{i-1} and all conclusions made from it[9].
5. Judge the plausibility of P_i. If implausible block this derivation.

It is worth noting that the need for retraction is not due to a failure to extract the correct 'least commitment' proposition from the semantic content of the fragment *Mary introduced*. This is due to the fact that it is possible to find pairs of possible continuations which are the negation of each other (e.g. *Mary introduced noone to anybody* and *Mary introduced someone to somebody*). The only proposition compatible with both a proposition, **p**, and its negation, ¬**p** is the trivial proposition, **T** (see Chater et al. for further discussion).

3. Incremental Quantifier Scoping

So far we have only considered semantic representations which do not involve quantifiers (except for the existential quantifier introduced by the mechanism above).

In sentences with two or more quantifiers, there is generally an ambiguity concerning which quantifier has wider scope. For example, in sentence (a) below the preferred reading is for the same kid to have climbed every tree (i.e. the universal quantifier is within the scope of the existential) whereas in sentence (b) the preferred reading is where the universal quantifier has scope over the existential.

7) a A tireless kid climbed every tree.
   b There was a fish on every plate.

Scope preferences sometimes seem to be established before the end of a sentence. For example, in sentence (a) below, there seems a preference for an outer scope reading for the first quantifier as soon as we interpret *child*. In (b) the preference, by the time we get to e.g. *grammar*, is for an inner scope reading for the first quantifier.

8) a A teacher gave every child a great deal of homework on grammar.

[7] The version of categorial grammar used is AB Categorial Grammar with Associativity.

[8] The proposition **T** is always true. See Chater et al. (1994) for discussion of whether it is more appropriate to use a non-trivial restrictor.

[9] Retraction can be performed by using a tagged database, where each proposition is paired with a set of sources e.g. given **(P→Q,{u4})**, and **(P,{u5})** then**(Q,{u4,u5})** can be deduced.

b Every girl in the class showed a rather strict new teacher the results of her attempt to get the grammar exercises correct.

This intuitive evidence can be backed up by considering garden path effects with quantifier scope ambiguities (called *jungle paths* by Barwise 1987). The original examples, such as the following,

9) Statistics show that every 11 seconds a man is mugged here in New York city. We are here today to interview him

showed that preferences for a particular scope are established and are overturned. To show that preferences are sometimes established before the end of a sentence, and before a potential sentence end, we need to show garden path effects in examples such as the following:

10) Mary put the information that statistics show that every 11 seconds a man is mugged here in New York city and that she was to interview him in her diary

Most psycholinguistic experimentation has been concerned with which scope preferences are made, rather than the point at which the preferences are established (see e.g. Kurtzman and MacDonald, 1993). Given the intuitive evidence, our hypothesis is that scope preferences can sometimes be established early, before the end of a sentence. This leaves open the possibility that in other cases, where the scoping information is not particularly of interest to the hearer, preferences are determined late, if at all.

3.1 Incremental Quantifier Scoping: Implementation

Dealing with quantifiers incrementally is a rather similar problem to dealing with fragments of trees incrementally. Just as it is impossible to predict the level of embedding of a noun phrase such as *John* from the fragment *Mary thinks John*, it is also impossible to predict the scope of a quantifier in a fragment with respect to the arbitrarily large number of quantifiers which might appear later in the sentence. Again the problem can be avoided by a form of packing. A particularly simple way of doing this is to use unscoped logical forms where quantifiers are left in situ (similar to the representations used by Hobbs and Shieber (1987), or to Quasi Logical Form (Alshawi 1990)). For example, the fragment *Every man gives a book* can be given the following representation:

11) $\lambda z.\mathrm{gives}(<\forall,x,\mathrm{man}(x)>,<\exists,y,\mathrm{book}(y)>,z)$

Each quantified term consists of a quantifier, a variable and a restrictor, but no body. To convert lambda expressions to unscoped propositions, we replace an occurrence of each argument with an empty existential quantifier term. In this case we obtain:

12) $\mathrm{gives}(<\forall,x,\mathrm{man}(x)>,<\exists,y,\mathrm{book}(y)>,<\exists,z,T>)$

Scoped propositions can then be obtained by using an outside-in quantifier scoping algorithm (Lewin, 1990), or an inside-out algorithm with a free variable constraint (Hobbs and Shieber, 1987). The propositions formed can then be judged for plausibility.

To imitate jungle path phenomena, these plausibility judgements need to feed back into the scoping procedure for the next fragment. For example, if *every man* is taken to be scoped outside *a book* after processing the fragment *Every man gave a book*, then this preference should be preserved when determining the scope for the full sentence *Every man gave a book to a child.* Thus instead of doing all quantifier scoping at the end of the sentence, each new quantifier is scoped relative to the existing quantifiers (and operators such as negation, intensional verbs etc.). A preliminary implementation achieves this by annotating the semantic representations with node names, and recording which quantifiers are 'discharged' at which nodes, and in which order.

DYNAMIC SEMANTICS

Dynamic semantics adopts the view that "the meaning of a sentence does not lie in its truth conditions, but rather in the way in which it changes (the representation of) the information of the interpreter" (Groenendijk and Stokhof, 1991). At first glance such a view seems ideally suited to incremental interpretation. Indeed, Groenendijk and Stokhof claim that the compositional nature of Dynamic Predicate Logic enables one to "interpret a text in an on-line manner, i.e., incrementally, processing and interpreting each basic unit as it comes along, in the context created by the interpretation of the text so far".

Putting these two quotes together is, however, misleading, since it suggests a more direct mapping between incremental semantics and dynamic semantics than is actually possible. In an incremental semantics, we would expect the information state of an interpreter to be updated word by word. In contrast, in dynamic semantics, the order in which states are updated is determined by semantic structure, not by left-to-right order (see e.g. Lewin, 1992 for discussion). For example, in Dynamic Predicate Logic (Groenendijk & Stokhof, 1991), states are threaded from the antecedent of a conditional into the consequent, and from a restrictor of a quantifier into the body. Thus, in interpreting,

13) John will buy it right away, if a car impresses him

the input state for evaluation of *John will buy it right away* is the output state from the antecedent *a car impresses him*. In this case the threading through semantic structure is in the opposite order to the order in which the two clauses appear in the sentence.

Some intuitive justification for the direction of threading in dynamic semantics is provided by considering appropriate orders for evaluation of propositions against a database: the natural order in which to evaluate a conditional is first to add the antecedent, and then see if the consequent can be proven. It is only at the sentence level in simple narrative texts that the presentation order and the natural order of evaluation necessarily coincide.

The ordering of anaphors and their antecedents is often used informally to justify left-to-right threading or threading through semantic structure. However, threading from left-to-right disallows examples of optional cataphora, as in example (13), and examples of compulsory cataphora as in:

14) Beside her, every girl could see a large crack

Similarly, threading from the antecedents of conditionals into the consequent fails for examples such as:

15) Every boy will be able to see out of a window if he wants to

It is also possible to get sentences with 'donkey' readings, but where the indefinite is in the consequent:

16) A student will attend the conference if we can get together enough money for her air fare

This sentence seems to get a reading where we are not talking about a particular student (an outer existential), or about a typical student (a generic reading). Moreover, as noted by Zeevat (1990), the use of any kind of *ordered* threading will tend to fail for Bach-Peters sentences, such as:

17) Every man who loves her appreciates a woman who lives with him

For this kind of example, it is still possible to use a standard dynamic semantics, but only if there is some prior level of reference resolution which reorders the antecedents and anaphors appropriately. For example, if (17) is converted into the 'donkey' sentence:

18) Every man who loves a woman who lives with him appreciates her

When we consider threading of possible worlds, as in Update Semantics (Veltman 1990), the need to distinguish between the order of evaluation and the order of presentation becomes more clear cut. Consider trying to perform threading in left-to-right order during interpretation of the sentence, *John left if Mary left.* After processing the proposition *John left* the set of worlds is refined down to those worlds in which John left. Now consider processing *if Mary left.* Here we want to reintroduce some worlds, those in which neither Mary or John left. However, this is not allowed by Update Semantics which is *eliminative*: each new piece of information can only further refine the set of worlds.

It is worth noting that the difficulties in trying to combine eliminative semantics with left-to-right threading apply to constraint-based semantics as well as to Update Semantics. Haddock (1987) uses incremental refinement of sets of possible referents. For example, the effect of processing *the rabbit* in the noun phrase *the rabbit in the hat* is to provide a set of all rabbits. The processing of *in* refines this set to rabbits which are in something. Finally, processing of *the hat* refines the set to rabbits which are in a hat. However, now consider processing *the rabbit in none of the boxes.* By the time *the rabbit in* has been processed, the only rabbits remaining in consideration are rabbits which are in something. This incorrectly rules out the possibility of the noun phrase referring to a rabbit which is in nothing at all. The case is actually a parallel to the earlier example of *Mary introduced someone to something* being inappropriate if the final sentence is *Mary introduced noone to anybody.*

Although this discussion has argued that it is not possible to thread the states which are used by a dynamic or eliminative semantics from left to right, word by word, this should not be taken as an argument against the use of such a semantics in incremental interpretation. What is required is a slightly more indirect approach. In the present implementation, semantic structures (akin to logical forms) are built word by word, and each structure is then evaluated independently using a dynamic semantics (with threading performed according to the structure of the logical form).

IMPLEMENTATION

At present there is a limited implementation, which performs a mapping from sentence fragments to fully scoped logical representations. To illustrate its operation, consider the following discourse:

19) London has a tower. Every parent shows it ...

We assume that the first sentence has been processed, and concentrate on processing the fragment. The implementation consists of five modules:

1. A word-by-word incremental parser for a lexicalised version of dependency grammar (Milward, 1992). This takes fragments of sentences and maps them to unscoped logical forms.

INPUT: **Every parent shows it**
OUTPUT:
λz.show(< ∀,x,parent(x)>,<pronoun,y>,z)

2. A module which replaces lambda abstracted variables with existential quantifiers in situ.

INPUT: Output from 1.
OUTPUT: **show(< ∀,x,parent(x)>,<pronoun,y>,< ∃,z,T>)**

3. A pronoun coindexing procedure which replaces pronoun variables with a variable from the same sentence, or from the preceding context.

INPUT: Output(s) from 2 and a list of variables available from the context.
OUTPUT: **show(< ∀,x,parent(x)>,w,< ∃,z,T>)**

4. An outside-in quantifier scoping algorithm based on Lewin (1990).

INPUT: Output from 3.
OUTPUT1: ∀(x,parent(x),∃(z,T,show(x,w,z)))
OUTPUT2: ∃(z,T,∀(x,parent(x),show(x,w,z)))

5. An 'evaluation' procedure based on Lewin (1992), which takes a logical form containing free variables (such as the *w* in the LF above), and evaluates it using a dynamic semantics in the context given by the preceding sentences. The output is a new logical form representing the context as a whole, with all variables correctly bound.

INPUT: Output(s) from 4, and the context, ∃(w,T,tower(w) & has(london,w))
OUTPUT1: ∃(w,T,tower(w) & has(london,w) & ∀(x,parent(x),∃(z,T,show(x,w,z))))
OUTPUT2: ∃(w,T,∃(z,T,tower(w) & has(london,w) & ∀(x,parent(x),show(x,w,z))))

At present, the coverage of module 5 is limited, and module 3 is a naive coindexing procedure which allows a pronoun to be coindexed with any quantified variable or proper noun in the context or the current sentence.

CONCLUSIONS

The paper described some potential applications of incremental interpretation. It then described the series of steps required in mapping from initial fragments of sentences to propositions which can be judged for plausibility. Finally, it argued that the apparently close relationship between the states used in incremental semantics and dynamic semantics fails to hold below the sentence level, and briefly presented a more indirect way of using dynamic semantics in incremental interpretation.

REFERENCES

Alshawi, H. (1990). Resolving Quasi Logical Forms. *Computational Linguistics, 16*, p.133-144.

Altmann, G.T.M. and M.J. Steedman (1988). Interaction with Context during Human Speech Comprehension. *Cognition, 30*, p.191-238.

Barwise, J. (1987). Noun Phrases, Generalized Quantifiers and Anaphors. In P. Gardenfors, Ed., *Generalized Quantifiers*, p.1-29, Dordrecht: Reidel.

Carletta, J., R. Caley and S. Isard (1993). A Collection of Self-repairs from the Map Task Corpus. Research Report, HCRC/TR-47, University of Edinburgh.

Chater, N., M.J. Pickering and D.R. Milward (1994). What is Incremental Interpretation? ms. To appear in Edinburgh Working Papers in Cognitive Science.

Cooper, R. (1993). A Note on the Relationship between Linguistic Theory and Linguistic Engineering. Research Report, HCRC/RP-42, University of Edinburgh.

Frazier, L. (1979). *On Comprehending Sentences: Syntactic Parsing Strategies.* Ph.D. Thesis, University of Connecticut. Published by Indiana University Linguistics Club.

Groenendijk, J. and M. Stokhof (1991). Dynamic Predicate Logic. *Linguistics and Philosophy, 14*, p.39-100.

Gross, D., J. Allen and D. Traum (1993). The TRAINS 91 Dialogues. TRAINS Technical Note 92-1, Computer Science Dept., University of Rochester.

Haddock, N.J. (1987). Incremental semantic interpretation and incremental syntactic analysis. Ph.D. Thesis, University of Edinburgh.

Haddock, N.J. (1989). Computational Models of Incremental Semantic Interpretation. *Language and Cognitive Processes, 4, (3/4)*, Special Issue, p.337-368.

Hobbs, J.R. and S.M. Shieber (1987). An Algorithm for Generating Quantifier Scoping. *Computational Linguistics, 3*, p47-63.

Joshi, A.K. (1987). An Introduction to Tree Adjoining Grammars. In Manaster-Ramer, Ed., *Mathematics of Language*, Amsterdam: John Benjamins.

Just, M. and P. Carpenter (1980). A Theory of Reading from Eye Fixations to Comprehension. *Psychological Review, 87*, p.329-354.

Kurtzman, H.S. and M.C. MacDonald (1993). Resolution of Quantifier Scope Ambiguities. *Cognition, 48(3)*, p.243-279.

Lewin, I. (1990). A Quantifier Scoping Algorithm without a Free Variable Constraint. In *Proceedings of COLING 90*, Helsinki, vol 3, p.190-194.

Lewin, I. (1992). Dynamic Quantification in Logic and Computational Semantics. Research report, Centre for Cognitive Science, University of Edinburgh.

Levelt, W.J.M. (1983). Modelling and Self-Repair in Speech. *Cognition, 14*, p.41-104.

Marcus, M., D. Hindle, and M. Fleck (1983). D-Theory: Talking about Talking about Trees. In *Proceedings of the 21st ACL*, Cambridge, Mass. p.129-136.

Marslen-Wilson, W. (1973). Linguistic Structure and Speech Shadowing at Very Short Latencies. *Nature, 244*, p.522-523.

Mellish, C.S. (1985). *Computer Interpretation of Natural Language Descriptions.* Chichester: Ellis Horwood.

Milward, D.R. (1991). Axiomatic Grammar, Non-Constituent Coordination, and Incremental Interpretation. Ph.D. Thesis, University of Cambridge.

Milward, D.R. (1992). Dynamics, Dependency Grammar and Incremental Interpretation. In *Proceedings of COLING 92*, Nantes, vol 4, p.1095-1099.

Moortgat, M. (1988). *Categorial Investigations: Logical and Linguistic Aspects of the Lambek Calculus*, Dordrecht: Foris.

Pulman, S.G. (1986). Grammars, Parsers, and Memory Limitations. *Language and Cognitive Processes, 1(3)*, p.197-225.

Resnik, P. (1992). Left-corner Parsing and Psychological Plausibility. In *Proceedings of COLING 92*, Nantes, vol 1, p.191-197.

Shieber, S.M. and M. Johnson (1993). Variations on Incremental Interpretation. *Journal of Psycholinguistic Research, 22(2)*, p.287-318.

Shieber, S.M. and Y. Schabes (1990). Synchronous Tree-Adjoining Grammars. In *Proceedings of COLING 90*, Helsinki, vol 3, p.253-258.

Stabler, E.P. (1991). Avoid the pedestrian's paradox. In R. Berwick, S. Abney, and C. Tenny, Eds., *Principle-Based Parsing: Computation and Psycholinguistics.* Kluwer.

Steedman, M. (1988). Combinators and Grammars. In R. Oehrle et al., Eds., *Categorial Grammars and Natural Language Structures*, p.417-442.

Thompson, H., M. Dixon, and J. Lamping (1991). Compose-Reduce Parsing. In *Proceedings of the 29th ACL*, p.87-97.

Tomita, M. (1985). *Efficient Parsing for Natural Language.* Kluwer.

Veltman F. (1990). Defaults in Update Semantics. In H. Kamp, Ed., *Conditionals, Defaults and Belief Revision*, DYANA Report 2.5.A, Centre for Cognitive Science, University of Edinburgh.

Wirén, M. (1990). Incremental Parsing and Reason Maintenance. In *Proceedings of COLING 90*, Helsinki, vol 3, p.287-292.

Zeevat, H. (1990). Static Semantics. In J. van Benthem, Ed., *Partial and Dynamic Semantics I*, DYANA Report 2.1.A, Centre for Cognitive Science, University of Edinburgh.