

Ambiguity Resolution in the DMTRANS PLUS

Hiroaki Kitano*, Hideto Tomabechi, and Lori Levin

Center for Machine Translation

Carnegie Mellon University

Pittsburgh, PA 15213 U.S.A.

Abstract

We present a cost-based (or energy-based) model of disambiguation. When a sentence is ambiguous, a parse with the least cost is chosen from among multiple hypotheses. Each hypothesis is assigned a cost which is added when: (1) a new instance is created to satisfy reference success, (2) links between instances are created or removed to satisfy constraints on concept sequences, and (3) a concept node with insufficient priming is used for further processing. This method of ambiguity resolution is implemented in DMTRANS PLUS, which is a second generation bi-directional English/Japanese machine translation system based on a massively parallel spreading activation paradigm developed at the Center for Machine Translation at Carnegie Mellon University.

1 Introduction

One of the central issues in natural language understanding research is ambiguity resolution. Since many sentences are ambiguous out of context, techniques for ambiguity resolution have been an important topic in natural language understanding. In this paper, we describe a model of ambiguity resolution implemented in DMTRANS PLUS, which is a next generation machine translation system based on a massively parallel computational paradigm. In our model, ambiguities are resolved by evaluating the cost of each hypothesis; the hypothesis with the least cost will be selected. Costs are assigned when (1) a new instance is created to satisfy reference success, (2) links between instances are created or removed to satisfy constraints on concept sequences, and (3) a concept node with insufficient priming is used for further processing.

The underlying philosophy of the model is to view parsing as a dynamic physical process in which one trajectory is taken from among many other possible paths. Thus our notion of *the cost of the hypothesis* is a representation of *the workload required to take the path representing the hypothesis*. One other important idea is that our model employs the direct memory

*E-mail address is hiroaki@al.nl.cs.cmu.edu. Also with NEC Corporation.

access (DMA) paradigm of natural language processing. Under the DMA paradigm, the mental state of the hearer is modelled by a massively parallel network representing memory. Parsing is performed by passing markers in the memory network. In our model, the meaning of a sentence is viewed as modifications made to the memory network. The meaning of a sentence in our model is definable as the difference in the memory network before and after understanding the sentence.

2 Limitations of Current Methods of Ambiguity Resolution

Traditional syntactic parsers have been using attachment preferences and local syntactic and semantic constraints for resolving lexical and structural ambiguities. ([17], [28], [2], [7], [26], [11], [5]) However, these methods cannot select one interpretation from several plausible interpretations because they do not incorporate the discourse context of the sentences being parsed ([8], [4]).

Connectionist-type approaches as seen in [18], [25], and [8] essentially stick to semantic restrictions and associations. However, [18], [25], [24] only provide local interactions, omitting interaction with context. Moreover, difficulties regarding variable-binding and embedded sentences should be noticed.

In [8], world knowledge is used through testing referential success and other sequential tests. However, this method does not provide a uniform model of parsing: lexical ambiguities are resolved by marker passing and structural disambiguations are resolved by applying separate sequential tests.

An approach by [15] is similar to our model in that both perceive parsing as a physical process. However, their model, along with most other models, fails to capture discourse context.

[12] uses marker passing as a method of contextual inference after a parse; however, no contextual information is feed-backed during the sentential parsing (marker-passing is performed after a separate parsing

process providing multiple hypotheses of the parse). [20] is closer to our model in that marker-passing based contextual inference is used during a sentential parse (i.e., an integrated processing of syntax, semantics and pragmatics at real-time); however the parsing (LFG, and case-frame based) and contextual inferences (marker-passing) are not under an uniform architecture.

Past generations of DMTRANS ([19], [23]) have not incorporated cost-based structural ambiguity resolution schemes.

3 Overview of DMTRANS PLUS

3.1 Memory Access Parsing

DMTRANS PLUS is a second generation DMA system based upon DMTRANS ([19]) with new methods of ambiguity resolution based on costs.

Unlike most natural language systems, which are based on the “Build-and-Store” model, our system employs a “Recognize-and-Record” model ([14],[19], [21]). Understanding of an input sentence (or speech input in Φ DMTRANS PLUS) is defined as changes made in a memory network. Parsing and natural language understanding in these systems are considered to be memory-access processes, identifying existent knowledge in memory with the current input. Sentences are always parsed in context, i.e., through utilizing the existing and (currently acquired) knowledge about the world. In other words, during parsing, relevant discourse entities in memory are constantly being remembered.

The model behind DMTRANS PLUS is a simulation of such a process. The memory network incorporates knowledge from morphophonetics to discourse. Each node represents a concept (Concept Class node; CC) or a sequence of concepts (Concept Sequence Class node; CSC).

CCs represent such knowledge as phones (i.e. [k]), phonemes (i.e. /k/), concepts (i.e. *Hand-Gun, *Event, *Mtrans-Action), and plans (i.e. *Pick-Up-Gun). A hierarchy of Concept Class (CC) entities stores knowledge both declaratively and procedurally as described in [19] and [21]. Lexical entries are represented as lexical nodes which are a kind of CC. Phoneme sequences are used only for Φ DMTRANS PLUS, the speech-input version of DMTRANS PLUS.

CSCs represent sequences of concepts such as phoneme sequences (i.e. <k//a//i//g//i/>), concept sequences (i.e. <*Conference *Goal-Role *Attend *Want>), and plan sequences (i.e. <*Declare-Want-Attend *Listen-Instruction>). The linguistic knowledge represented as CSCs can be low-level surface specific patterns such as phrasal lexicon entries [1]

or material at higher levels of abstraction such as in MOP’s [16]. However, CSCs should not be confused with ‘discourse segments’ [6]. In our model, information represented in discourse segments are distributively incorporated in the memory network.

During sentence processing we create concept instances (CI) corresponding to CCs and concept sequence instances (CSI) corresponding to CSCs. This is a substantial improvement over past DMA research. Lack of instance creation and reference in past research was a major obstacle to seriously modelling discourse phenomena.

CIs and CSIs are connected through several types of links. A guided marker passing scheme is employed for inference on the memory network following methods adopted in past DMA models.

DMTRANS PLUS uses three markers for parsing:

- An Activation Marker (A-Marker) is created when a concept is initially activated by a lexical item or as a result of *concept refinement*. It indicates which instance of a concept is the source of activation and contains relevant cost information. A-Markers are passed upward along is-a links in the abstraction hierarchy.
- A Prediction marker (P-Marker) is passed along a concept sequence to identify the linear order of concepts in the sequence. When an A-Marker reaches a node that has a P-Marker, the P-Marker is sent to the next element of the concept sequence, thus predicting which node is to be activated next.
- A Context marker (C-Marker) is placed on a node which has contextual priming.

Information about which instances originated activations is carried by A-Markers. The binding list of instances and their roles are held in P-Markers¹.

The following is the algorithm used in DMTRANS PLUS parsing:

Let *Lex*, *Con*, *Elem*, and *Seq* be a set of lexical nodes, conceptual nodes, elements of concept sequences, and concept sequences, respectively.

Parse(S)

For each word *w* in *S*, do:

Activate(w),

For all *i* and *j*:

if $Active(N_i) \wedge N_i \in Con$

¹Marker parsing spreading activation is our choice over connectionist network precisely because of this reason. Variable binding (which cannot be easily handled in connectionist network) can be trivially attained through structure (information) passing of A-Markers and P-Markers.

then do concurrently:
Activate(isa(N_i))
 if *Active(e_j.N_i) ∧ Predicted(e_j.N_i) ∧ ¬Last(e_j.N_i)*
 then *Predict(e_{j+1}.N_i)*
 if *Active(e_j.N_i) ∧ Predicted(e_j.N_i) ∧ Last(e_j.N_i)*
 then *Accept(N_i), Activate(isa(N_i))*

Predict(N)
 for all *N_i ∈ N* do:
 if *N_i ∈ Con*,
 then *Pmark(N_i), Predict(isainv(N_i))*
 if *N_i ∈ Elem*,
 then *Pmark(N_i), Predict(isainv(N_i))*
 if *N_i ∈ Seq*,
 then *Pmark(e₀.N_i), Predict(isainv(e₀.N_i))*
 if *N_i = NIL*,
 then *Stop*.

Activate
I ← instanceof(c)
 if *i = φ* then
createinst(c), Addcost, activate(c)
 else
 for each *i ∈ I*
 do concurrently:
activate(c)

Accept
 if *Constraints ≠ T*
Assume(Constraints), Addcost
activate(isa(c))

where *N_i* and *e_j.N_i* denote a node in the memory network indexed by *i* and a *j*-th element of a node *N_i*, respectively.

Active(N) is true iff a node or an element of a node gets an A-Marker.

Activate(N) sends A-Markers to nodes and elements given in the argument.

Predict(N) moves a P-Marker to the next element of the CSC.

Predicted(N) is true iff a node or an element of a node gets a P-Marker.

Pmark(N) puts a P-Marker on a node or an element given in the argument.

Last(N) is true iff an element is the last element of the concept sequence.

Accept(N) creates an instance under *N* with links which connect the instance to other instances.

isa(N) returns a list of nodes and elements which are connected to the node in the argument by abstraction links.

isainv(N) returns a list of nodes and elements which are daughters of a node *N*.

Some explanation would help understanding this algorithm:

1. Prediction.

Initially all the first elements of concept sequences (CSC - Concept Sequence Class) are predicted by putting P-Markers on them.

2. Lexical Access.

A lexical node is activated by the input word.

3. Concept Activation.

An A-Marker is created and sent to the corresponding CC (Concept Class) nodes. A cost is added to the A-Marker if the CC is not C-Marked (i.e. A C-Marker is not placed on it).

4. Discourse Entity Identification

A CI (Concept Instance) under the CC is searched for.

If the CI exists, an A-Marker is propagated to higher CC nodes.

Else, a CI node is created under the CC, and an A-Marker is propagated to higher CC nodes.

5. Activation Propagation.

An A-Marker is propagated upward in the abstraction hierarchy.

6. Sequential Prediction.

When an A-Marker reaches any P-Marked node (i.e. part of CSC), the P-Marker on the node is sent to the next element of the concept sequence.

7. Contextual Priming

When an A-Marker reaches any Contextual Root node. C-Markers are put on the contextual children nodes designated by the root node.

8. Conceptual Relation Instantiation.

When the last element of a concept sequence receives an A-Marker, Constraints (world and discourse knowledge) are checked for.

A CSI is created under the CSC with packaging links to each CI. This process is called concept refinement. See [19].

The memory network is modified by performing inferences stored in the root CSC which had the accepted CSC attached to it.

9. Activation Propagation

A-Marker is propagated from the CSC to higher nodes.

3.2 Memory Network Modification

Several different incidents trigger the modification of the memory network during parsing:

- An individual concept is instantiated (i.e. an instance is created) under a CC when the CC receives an A-Marker and a CI (an instance that

was created by preceding utterances) is not existent. This instantiation is a creation of a specific discourse entity which may be used as an existent instance in the subsequent recognitions.

- A concept sequence instance is created under the accepted CSC. In other words, if a whole concept sequence is accepted, we create an instance of the sequence instantiating it with the specific CIs that were created by (or identified with) the specific lexical inputs. This newly created instance is linked to the accepted CSC with a instance relation link and to the instances of the elements of the concept sequences by links labelled with their roles given in the CSC.
- Links are created or removed in the CSI creation phase as a result of invoking inferences based on the knowledge attached to CSCs. For example, when the parser accepts the sentence *I went to the UMIST*, an instance of *I* is created under the CC representing *I*. Next, a CSI is created under PTRANS. Since PTRANS entails that the agent is at the location, a location link must be created between the discourse entities *I* and *UMIST*. Such revision of the memory network is conducted by invoking knowledge attached to each CSC.

Since modification of any part of the memory network requires some *workload*, certain costs are added to analyses which require such modifications.

4 Cost-based Approach to the Ambiguity Resolution

Ambiguity resolution in DMTRANS PLUS is based on the calculation of the cost of each parse. Costs are attached to each parse during the parse process.

Costs are attached when:

1. A CC with insufficient priming is activated,
2. A CI is created under CC, and
3. Constraints imposed on CSC are not satisfied initially and links are created or removed to satisfy the constraint.

Costs are attached to A-Markers when these operations are taken because these operations modify the memory network and, hence, workloads are required. Cost information is then carried upward by A-Markers. The parse with the least cost will be chosen.

The cost of each hypothesis are calculated by:

$$C_i = \sum_{j=0}^n c_{ij} + \sum_{k=0}^m \text{constraint}_{ik} + \text{bias}_i$$

where C_i is a cost of the i -th hypothesis, c_{ij} is a cost carried by an A-Marker activating the j -th element of the CSC for the i -th hypothesis, constraint_{ik} is a cost of assuming k -th constraint of the i -th hypothesis, and bias_i represents lexical preference of the CSC for the i -th hypothesis. This cost is assigned to each CSC and the value of C_i is passed up by A-Markers if higher-level processing is performed. At higher levels, each c_{ij} may be a result of the sum of costs at lower-levels.

It should be noted that this equation is very similar to the activation function of most neural networks except for the fact our equation is a simple linear equation which does not have threshold value. In fact, if we only assume the addition of cost by priming at the lexical-level, our mechanism of ambiguity resolution would behave much like connectionist models without inhibition among syntactic nodes and excitation links from syntax to lexicon². However, the major difference between our approach and the connectionist approach is the addition of costs for instance creation and constraint satisfaction. We will show that these factors are especially important in resolving structural ambiguities.

The following subsections describe three mechanisms that play a role in ambiguity resolution. However, we do not claim that these are the only mechanisms involved in the examples which follow³.

4.1 Contextual Priming

In our system, some CC nodes designated as Contextual Root Nodes have a list of thematically relevant nodes. C-Markers are sent to these nodes as soon as a Contextual Root Node is activated. Thus each sentence and/or each word might influence the interpretation of following sentences or words. When a node with C-Marker is activated by receiving an A-Marker, the activation will be propagated with no cost. Thus, a parse using such nodes would have no cost. However, when a node without a C-Marker is activated, a small cost is attached to the interpretation using that node.

In [19] the discussion of C-Marker propagation concentrated on the resolution of word-level ambiguities. However, C-Markers are also propagated to conceptual

²We have not incorporated these factors primarily because structured P-Markers can play the role of top-down priming; however, we may be incorporating these factors in the future.

³For example, in one implementation of DMTRANS, we are using time-delayed decaying activations which resolve ambiguity even when two CI nodes are concurrently active.

class nodes, which can represent word-level, phrasal, or sentential knowledge. Therefore, C-Markers can be used for resolving phrasal-level and sentential-level ambiguities such as structural ambiguities. For example, *atama ga itai* literally means, '(my) head hurts.' This normally is identified with the concept sequences associated with the *have-a-symptom concept class node, but if the preceding sentence is *asita yakuinkai da* ('There is a board of directors meeting tomorrow'), the *have-a-problem concept class node must be activated instead. Contextual priming attained by C-Markers can also help resolve structural ambiguity in sentences like *did you read about the problem with the students?* The cost of each parse will be determined by whether *reading with students* or *problems with students* is contextually activated. (Of course, many other factors are involved in resolving this type of ambiguity.)

Our model can incorporate either C-Markers or a connectionist-type competitive activation and inhibition scheme for priming. In the current implementation, we use C-Markers for priming simply because C-Marker propagation is computationally less-expensive than connectionist-type competitive activation and inhibition schemes⁴. Although connectionist approaches can resolve certain types of lexical ambiguity, they are computationally expensive unless we have massively parallel computers. C-Markers are a reasonable compromise because they are sent to semantically relevant concept nodes to attain contextual priming without computationally expensive competitive activation and inhibition methods.

4.2 Reference to the Discourse Entity

When a lexical node activates any CC node, a CI node under the CC node is searched for ([19], [21]). This activity models reference to an already established discourse entity [27] in the hearer's mind. If such a CI node exists, the reference succeeds and this parse will be attached with no cost. However, if no such instance is found, reference failure results. If this happens, an instantiation activity is performed creating a new instance with certain costs. As a result, a parse using newly created instance node will be attached with some cost.

For example, if a preceding discourse contained a reference to a thesis, a CI node such as THESIS005 would have been created. Now if a new input sentence contains the word *paper*, CC nodes for THE-

⁴This does not mean that our model can not incorporate a connectionist model. The choice of C-Markers over the connectionist approach is mostly due to computational cost. As we will describe later, our model is capable of incorporating a connectionist approach.

SIS and SHEET-OF-PAPER are activated. This causes search for CI nodes under both CC nodes. Since the CI node THESIS005 will be found, the reading where *paper* means thesis will not acquire a cost. However, assuming that there is not a CI node corresponding to a sheet of paper, we will need to create a new one for this reading, thus incurring a cost.

We can also use reference to discourse entities to resolve structural ambiguities. In the sentence *We sent her papers*, if the preceding discourse mentioned Yoshiko's papers, a specific CI node such as YOSHIKO PAPER003 representing Yoshiko's papers would have been created. Therefore, during the processing of *We sent her papers*, the reading which means we sent papers to her needs to create a CI node representing papers that we sent, incurring some cost for creating the instance node. On the other hand, the reading which means we sent Yoshiko's papers does not need to create an instance (because it was already created) so it is costless. Also, the reading that uses *paper* as a sheet of paper is costly as we have demonstrated above.

4.3 Constraints

Constraints are attached to each CSC. These constraints play important roles during disambiguation. Constraints define relations between instances whose sentences or sentence fragments are accepted. When a constraint is satisfied, the parse is regarded as plausible. On the other hand, the parse is less plausible when the constraint is unsatisfied. Whereas traditional parsers simply reject a parse which does not satisfy given constraint, DMTRANS PLUS, builds or removes links between nodes forcing them to satisfy constraint. A parse with such forced constraints will record an increased cost and will be less preferred than parse without attached costs.

The following example illustrates how this scheme resolves an ambiguity. As an initial setting we assume that the memory network has instances of 'man' (MAN1) and 'hand-gun' (HAND-GUN1) connected with a POSSES relation (i.e. link). The input utterance is: "Mary picked up an Uzzi. Mary shot the man with the hand-gun." The second sentence is ambiguous in isolation and it is also ambiguous if it is not known that an Uzzi is a machine gun. However, when it is preceded by the first sentence and if the hearer knows that Uzzi is a machine gun, the ambiguity is drastically reduced. DMTRANS PLUS hypothesizes and models this disambiguation activity utilizing knowledge about the world through the cost recording mechanism described above.

During the processing of the first sentence, DMTRANS PLUS creates instances of 'Mary' and 'Uzzi

and records them as active instances in memory (i.e., MARY1 and UZZI1 are created). In addition, a link between MARY1 and UZZI1 is created with the POSSES relation label. This link creation is invoked by triggering side-effects (i.e., inferences) stored in the CSC representing the action of 'MARY1 picking up the UZZI1'. We omit the details of marker passing (for A-, P-, and C-Markers) since it is described detail elsewhere (particular in [19]).

When the second sentence comes in, an instance MARY1 already exists and, therefore, no cost is charged for parsing 'Mary'⁵. However, we now have three relevant concept sequences (CSC's⁶):

CSC1: (<agent> <shoot> <object>)

CSC2: (<agent> <shoot> <object> <with> <instrument>)

CSC3: (<person> <with> <instrument>)

These sequences are activated when concepts in the sequences are activated in order from below in the abstraction hierarchy. When the "man" comes in, recognition of CSC3:(<person> <with> <instrument>) starts. When the whole sentence is received, we have two top-level CSCs (i.e., CSC1 and CSC2) accepted (all elements of the sequences recognized). The acceptance of CSC1 is performed through first accepting CSC3 and then substituting CSC3 for <object>.

When the concept sequences are satisfied, their constraints are tested. A constraint for CSC2 is (POSSES <agent> <instrument>) and a constraint for CSC3 (and CSC1, which uses CSC3) is (POSSES <person> <instrument>). Since 'MARY1 POSSESS HAND-GUN1' now has to be satisfied and there is no instance of this in memory, we must create a POSSESS link between MARY1 and HAND-GUN1. A certain cost, say 10, is associated with the creation of this link. On the other hand, MAN1 POSSESS HAND-GUN1 is known in memory because of an earlier sentence. As a result, CSC3 is instantiated with no cost and an A-Marker from CSC3 is propagated upward to CSC1 with no cost. Thus, the cost of instantiating CSC1 is 0 and the cost of instantiating CSC2 is 10. This way, the interpretation with CSC1 is favored by our system.

⁵Of course, 'Mary' can be 'She'. The method for handling this type of pronoun reference was already reported in [19] and we do not discuss it here.

⁶As we can see from this example of CSC's, a concept sequence can be normally regarded as a subcategorization list of a VP head. However, concept sequences are not restricted to such lists and are actually often at higher levels of abstraction representing MOP-like sequences.

5 Discussion:

5.1 Global Minima

The correct hypothesis in our model is the hypothesis with the least cost. This corresponds to the notion of global minima in most connectionist literature. On other hand, the hypothesis which has the least cost within a local scope but does not have the least cost when it is combined with global context is a local minimum. The goal of our model is to find a global minimum hypothesis in a given context. This idea is advantageous for discourse processing because a parse which may not be preferred in a local context may yield a least cost hypothesis in the global context. Similarly, the least costing parse may turn out to be costly at the end of processing due to some contextual inference triggered by some higher context.

One advantage of our system is that it is possible to define global and local minima using massively parallel marking passing, which is computationally efficient and is more powerful in high-level processing involving variable-binding, structure building, and constraint propagations⁷ than neural network models. In addition, our model is suitable for massively parallel architectures which are now being researched by hardware designers as next generation machines⁸.

5.2 Psycholinguistic Relevance of the Model

The phenomenon of lexical ambiguity has been studied by many psycholinguistic researchers including [13], [3], and [17]. These studies have identified contextual priming as an important factor in ambiguity resolution.

One psycholinguistic study that is particularly relevant to DMTRANS PLUS is Crain and Steedman [4], which argues for the principle of referential success. Their experiments demonstrate that people prefer the interpretation which is most plausible and accesses previously defined discourse entities. This psycholinguistic claim and experimental result was incorporated in our model by adding costs for instance creation and constraint satisfaction.

Another study relevant to our model is the lexical preference theory by Ford, Bresnan and Kaplan [5]. Lexical preference theory assumes a preference order among lexical entries of verbs which differ in subcategorization for prepositional phrases. This type of preference was incorporated as the *bias* term in our cost equation.

⁷Refer to [22] for details in this direction.

⁸See [23] and [9] for discussion.

Although we have presented a basic mechanism to incorporate these psycholinguistic theories, well controlled psycholinguistic experiments will be necessary to set values of each constant and to validate our model psycholinguistically.

5.3 Reverse Cost

In our example in the previous section, if the first sentence was *Mary picked an S&W* where the hearer knows that an S&W is a hand-gun, then an instance of 'MARY POSSES HAND-GUN1' is asserted as true in the first sentence and no cost is incurred in the interpretation of the second sentence using CSC2. This means that the cost for both PP-attachments in *Mary shot the man with the handgun* are the same (no cost in either cases) and the sentence remains ambiguous. This seems contrary to the fact that in *Mary picked a S&W. She shot the man with the hand-gun*, that natural interpretation (given that the hearer knows S&W is a hand-gun) seems to be that it was Mary that had the hand-gun not *the man*. Since our costs are only negatively charged, the fact that 'MARY1 POSSES S&W' is recorded in previous sentence does not help the disambiguation of the second sentence.

In order to resolve ambiguities such as this one which remain after our cost-assignment procedure has applied, we are currently working on a reverse cost charge scheme. This scheme will retroactively increase or decrease the cost of parses based on other evidence from the discourse context. For example, the discourse context might contain information that would make it more plausible or less plausible for Mary to use a handgun. We also plan to implement time-sensitive diminishing levels of charges to prefer facts recognized in later utterances.

5.4 Incorporation of Connectionist Model

As already mentioned, our model can incorporate connectionist models of ambiguity resolution. In a connectionist network activation of one node triggers interactive excitation and inhibition among nodes. Nodes which get more activated will be primed more than others. When a parse uses these more active nodes, no cost will be added to the hypothesis. On the other hand, hypotheses using less activated nodes should be assigned higher costs. There is nothing to prevent our model from integrating this idea, especially for lexical ambiguity resolution. The only reason that we do not implement a connectionist approach at present is that the computational cost will be enormous on current computers. Readers should also be aware that DMA is a guided marker passing al-

gorithm in which markers are passed only along certain links whereas connectionist models allow spreading of activation and inhibition virtually to any connected nodes. We hope to integrate DMA and connectionist models on a real massively parallel computer and wish to demonstrate real-time translation. One other possibility is to integrate with a connectionist network for speech recognition⁹. We expect, by integrating with connectionist networks, to develop a uniform model of cost-based processing.

6 Conclusion

We have described the ambiguity resolution scheme in DMTRANS PLUS. Perhaps the central contribution of this paper to the field is that we have shown a method of ambiguity resolution in a massively parallel marker passing paradigm. Cost evaluation for each parse through (1) reference and instance creation, (2) constraint satisfaction and (3) C-Markers are combined into the marker passing model. We have also discussed on the possibility to merge our model with connectionist models where they are applicable. The guiding principle of our model, that parsing is a physical process of memory modification, was useful in deriving mechanisms described in this paper. We expect further investigation along these lines to provide us insights in many aspects of natural language processing.

Acknowledgements

The authors would like to thank members of the Center for Machine Translation for fruitful discussions. We would especially like to thank Masaru Tomita, Hitoshi Iida, Jaime Carbonell, and Jay McClelland for their encouragement.

Appendix: Implementation

DMTRANS PLUS is implemented on IBM-RT's using both CMU-COMMONLISP and MULTILISP running on the Mach distributed operating system at CMU. Algorithms for structural disambiguation using cost attachment were added along with some other house-keeping functions to the original DMTRANS to implement DMTRANS PLUS. All capacities reported in this paper have been implemented except the schemes mentioned in the sections 5.3 and 5.4 (i.e., negative costs, integration of connectionist models).

⁹Augmentation of the cost-based model to the phonological level has already been implemented in [10].

References

- [1] Becker, J.D. *The phrasal lexicon*. In 'Theoretical Issues in Natural Language Processing', 1975.
- [2] Boguraev, B. K., et. al., *Three Papers on Parsing*, Technical Report 17, Computer Laboratory, University of Cambridge, 1982.
- [3] Cottrell, G., *A Model of Lexical Access of Ambiguous Words*, in 'Lexical Ambiguity Resolution', S. Small, et. al. (eds), Morgan Kaufmann Publishers, 1988.
- [4] Crain, S. and Steedman, M., *On not being led up with garden path: the use of context by the psychological syntax processor*, in 'Natural Language Parsing', 1985.
- [5] Ford, M., Bresnan, J. and Kaplan, R., *A Competence-Based Theory of Syntactic Closure*, in 'The Mental Representation of Grammatical Relations', 1981.
- [6] Grosz, B. and Sidner, C. L., *The Structure of Discourse Structure*, CSLI Report No. CSLI-85-39, 1985.
- [7] Hays, P. J., *On semantic nets, frames and associations*, in 'Proceedings of IJCAI-77', 1977.
- [8] Hirst, G., *Semantic Interpretation and the Resolution of Ambiguity*, Cambridge University Press, 1987.
- [9] Kitano, H., *Multilingual Information Retrieval Mechanism using VLSI*, in 'Proceedings of RIAO-88', 1988.
- [10] Kitano, H., et. al., Manuscript *An Integrated Discourse Understanding Model for an Interpreting Telephony under the Direct Memory Access Paradigm*, Carnegie Mellon University, 1989.
- [11] Marcus, M. P., *A theory of syntactic recognition for natural language*, MIT Press, 1980.
- [12] Norvig, P., *Unified Theory of Inference for Text Understanding*, Ph.D. Dissertation, University of California, Berkeley, 1987.
- [13] Prather, P. and Swinney, D., *Lexical Processing and Ambiguity Resolution: An Autonomous Processing in an Interactive Box*, in 'Lexical Ambiguity Resolution', S. Small, et. al. (Eds), Morgan Kaufmann Publishers, 1988.
- [14] Riesbeck, C. and Martin, C., *Direct Memory Access Parsing*, YALEU/DCS/RR 354, 1985.
- [15] Selman, B. and Hirst, G., *Parsing as an Energy Minimization Problem*, in *Genetic Algorithms and Simulated Annealing*, Davis, L. (Ed.), Morgan Kaufmann Publishers, CA, 1987.
- [16] Schank, R., *Dynamic Memory: A theory of learning in computers and people*. Cambridge University Press. 1982
- [17] Small, S., et. al. (Eds.) *Lexical Ambiguity Resolution*, Morgan Kaufmann Publishers, Inc., CA, 1988.
- [18] Small, S., et. al. *Toward Connectionist Parsing*, in Proceedings of AAAI-82, 1982.
- [19] Tomabechi, H., *Direct Memory Access Translation*, in 'Proceedings of the IJCAI-88', 1987.
- [20] Tomabechi, H. and Tomita, M., *The Integration of Unification-based Syntax/Semantics and Memory-based Pragmatics for Real-Time Understanding of Noisy Continuous Speech Input*, in 'Proceedings of the AAAI-88', 1988.
- [21] Tomabechi, H. and Tomita, M., *Application of the Direct Memory Access paradigm to natural language interfaces to knowledge-based systems*, in 'Proceedings of the COLING-88', 1988.
- [22] Tomabechi, H. and Tomita, M., Manuscript. *MASSIVELY PARALLEL CONSTRAINT PROPAGATION: Parsing with Unification-based Grammar without Unification*. Carnegie Mellon University.
- [23] Tomabechi, H., Mitamura, T., and Tomita, M., *DIRECT MEMORY ACCESS TRANSLATION FOR SPEECH INPUT: A Massively Parallel Network of Episodic/Thematic and Phonological Memory*, in 'Proceedings of the International Conference on Fifth Generation Computer Systems 1988' (FGCS'88), 1988.
- [24] Touretzky, D. S., *Connectionism and PP Attachment*, in 'Proceedings of the 1988 Connectionist Models Summer School, 1988.
- [25] Waltz, D. L. and Pollack, J. B., *Massively Parallel Parsing: A Strongly Interactive Model of Natural Language Interpretation*. *Cognitive Science* 9(1): 51-74, 1985.
- [26] Wanner, E., *The ATN and the Sausage Machine: Which one is baloney?* *Cognition*, 8(2), June, 1980.
- [27] Webber, B. L., *So what can we talk about now?*, in 'Computational Models of Discourse', (Eds. M. Brady and R.C. Berwick), MIT Press, 1983.
- [28] Wilks, Y. A., Huang, X. and Fass, D., *Syntax, preference and right attachment*, in 'Proceedings of the IJCAI-85, 1985.