COMPUTERS AND ARTIFICIAL INTELLIGENCE

ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ И ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

POČÍTAČE A UMELÁ INTELIGENCIA

f-p1

reprint

SLOVENSKÁ AKADÉMIA VIED

ÚSTAV TECHNICKEJ KYBERNETIKY

MAIN JOURNAL TOPICS

KNOWLEDGE PROCESSING	Cognitive science, Learning, Knowledge representation, Knowledge bases, Expert system tools, Natural language processing
PERCEPTION	Image processing, Pattern recognition, Vision systems, Tactile systems, Speech recognition
BASIC THEORIES	Problem solving, Theorem proving, Computational models, Computational Complexity, Parallel computation, Language semantics
PROGRAMMING	Functional programming, Logic programming, Object-oriented programming, Intelligent programming environment
COMPUTER ARCHITECTURE	Parallel machines, Data-flow machines, Reduction machines, Inference machines, Knowledge-based machines, High-level-language machines
TECHNICAL APPLICATIONS	Intelligent robot systems, Intelligent CAD/CAM systems, Computer aided engineering, Computer graphics, VLSI design technology
OTHER APPLICATIONS	Education, Diagnostics, Planning, Machine translation systems, Dialogue systems

Apart from the main topics referred to, the Editorial Board will give preference to publishing papers from the field of Computer Sciences related to Artificial Intelligence, as well as from Artificial Intelligence related to computers and advanced robots.

Research papers will be published in English or Russian only, abstracts of papers in English and Russian.

THE DIRECT MEMORY ACCESS PARADIGM AND ITS APPLICATIONS TO NATURAL LANGUAGE PROCESSING

Hideto TOMABECHI, Masaru TOMITA

Center for Machine Translation, Carnegie Mellon University, Pittsburgh, USA

Abstract. This paper describes three types of applications of the Direct Memory Access (DMA) paradigm to natural language understanding, namely 1) machine translation; 2) speech understanding; 3) natural language interfaces. The results of research utilizing the DMA approach at Center for Machine Translation (CMT) at Carnegie Mellon University in these three areas are reported.

Парадигма прямого доступа к памяти и ее применения в обработке естественного языка
Хидето Томабети и Масару Томита

Резюме. В статье описаны три типа применений парадигмы прямого доступа к памяти (DMA) к пониманию естественного языка, а именно 1) машинный перевод; 2) понимание речи; 3) интерфейсы естественного языка. Приводятся результаты исследований, использующие подход DMA в этих трех областях в Центре машинного перевода (CMT) в Университете Карнеги Меллон.

Keywords: direct memory access, machine translation, speech understanding, natural language interface, parsing.

CR classification descriptors: I.2.7. [Natural Language Processing], Language parsing and understanding, Speech recognition and understanding, I.2.1. [Applications and Expert Systems], Natural language interfaces.

PART I

OVERVIEW OF DMA PARADIGM

1. SOME BACKGROUND AND HISTORY

The Direct Memory Access (DMA) method of parsing originated in QUILLIAN's notion of semantic memory (see [31]), which was used in his TLC (see [32]) and led to further research in semantic network-based processing[1]. TLC used breadth-first spreading marker-passing as an intersection search of two lexically pointed nodes in semantic memory, leaving interpretation of text as an intersection of the paths. Thus, interpretation of input text was directly performed on semantic memory. Although TLC was the first DMA system, DMA had not been explored as a model of parsing until the DMAP0 system of [36], except as a scheme for disambiguations. DMAP0 used a guided marker-passing algorithm to avoid the problem of an explosion of search paths, from which a dumb[2] (not guided) marker passing mechanism inherently suffers. DMAP0 used P-markers (Prediction markers) and A-markers (Activation markers) as markers passed around in memory, adopting the notion of concept sequence to represent linear ordering of concepts as linguistic knowledge, which guides linear predictions of concepts sending P-markers in memory. Concept sequences, which encompass phrasal patterns, are attached to notes in memory that represent some specific experiential memory structure. In DMAP0, A-markers are sent above in the abstraction hierarchy from the lexically activated node in memory, and P-markers are sent to the next element of the concept sequence only after the A-marker from below hits a node that is already P-marked. Concept refinement is performed using concept refinement links (Cref-links) when a whole concept sequence is activated. Concept refinement locates the most specific node in memory, below the activated root node, which represents the specific instance of the input text. DMTRANS (see [42]) evolved DMA into a theory of cross-linguistic translations and added mechanisms of explanatory generation, C-Marker passing (for further contextual disambiguations), and a revised scheme of concept refinement while performing English/Japanese translations.

In the DMA model, natural language understanding is viewed as a memory activity which identifies input with what is already stored in memory as episodic (experiential) and thematic knowledge. This is contrasted with the traditional model of parsing, which we call the "build-and-store" paradigm, in which a syntactic parser (with the help of semantics) builds up a tree-style representation of an input sentence, and processing is done sentence by sentence with little (if any) interaction between parses. In other words, the DMA paradigm models the human mind in the sense that both linguistic and non-linguistic experiences are being remembered during the course of understanding the input, and each sentence that is recognized records a context that influences the processing of successive inputs. On the other hand, in traditional (non-DMA) systems, each input sentence is parsed into syntactic trees, and semantics are used primarily as a tool for guaranteeing the right configuration of syntactic trees; normally, no long-term memory (such as experiential memory) is involved during the parse. Also, in these systems, the result of a parse is usually lost after the processing of each sentence.

---

[1] Such as [10], [17], [4], [15], [18], [6], [28] and connectionist and distributed parallel models including [41], [11], [53], [2], and [3].

[2] We call it 'dumb' when markers are passed everywhere (through all links) from a node. In a 'guided' scheme, markers are passed through specific links only.

2. WHAT IS REPRESENTED IN DMA

Our DMA view of natural language understanding differs from the traditional "build-and-store" model in that natural language input is not represented as an independent syntactic and semantic representational structure (such as the case-frame representation in our own large-scale machine translation system at CMT).

For example, below is the parse output of our build-and-store parser (Universal Parser, [47]) using case-frame representation.

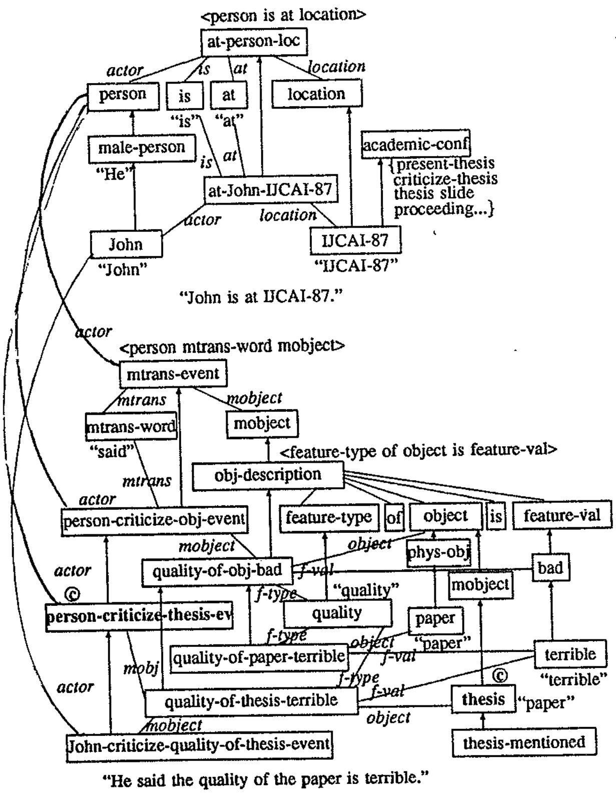


Fig. 1 Concept refinement and context marking

While the parser succeeded in building a representational structure that contains enough information for the generator to produce the output: *Ich habe beim Sprechen meiner Schmerzen im Rachen*, the build-and-store parser has no way of relating the sentence to other utterances the speakers have made unless it triggers some other module (such as inferences). For any external modules to handle any extra-sentential phenomenon (such as contextual disambiguation), it has to access some representation of such extra-sentential information.

In the real-world natural language understanding situation, sentences are always uttered in context. It seems it is only in our linguistics text books that sentences are quoted and treated totally out of context.

However, traditional parsers (at least until now) have neglected this fact (that real sentences are always in context), and always build and store representations for one sentence at a time. This inevitably means that any extra-sentential phenomena (such as disambiguation) are out of the scope of these traditional parsers. There have been some efforts in the area developing inferential mechanisms (such as [28]) that received the representations of the build-and-store sentence-based parsers and somehow connected the representations for each sentence in some contextual manner; however, since parsers acted independently of these modules in the first place, no such contextual information was utilized during parsing[3].

In DMA parsing, a sentence is not a unit of representation. In other words, a whole input session (such as a discourse) is a basic unit of representation because the memory activity of parsing each sentence is recorded. In our model what has been input for the life of the network is represented, the most recently recognized sentence as just another new instance of input to the network. Since a sentence is not a representational unit, handling extra-sentential phenomena is not especially different from handling intra-sentential phenomena. For example, a scheme for pronoun reference resolution (such as overt or empty subject pronouns in Japanese) within a sentence in DMA parsing can apply even if an antecedent NP was in some other preceding sentences.

As a brief example, in Government and Binding Theory, Condition B of Binding Theory tells us that *John_i saw him_i* is ungrammatical, and a build-and-store GB parser (or any parser that uses this principle) knows that *him* must not be *John*. However, 1) there is no way that this build-and-store parser knows who *him* is actually coindexed with, because that person is probably mentioned in some previous sentences; 2) and depending upon syntactic context, Condition B may not always apply (such as in *John_i placed the telephone near him_i*[4]); 3) and more interestingly, natural language input may not be grammatical anyway, (i.e., parsers are not grammaticality judgers but are natural language recognizers) especially in speech understanding attempts. In parsing *John saw him*, a DMA parser recognizes the input as some specific instance of a general event that is known in memory as an event of someone *mtrans*ing some image of someone to himself along with many past instances of similar cases and generalizations, expectations, explanations, etc., that accompany the event that are part of the whole memory network. By the participation of such a knowledge and the syntactic knowledge supplied to the network, the DMA parser is capable of attaining the effects of Condition B as well as other recognitions that purely syntactic principles may not be able to perform. Moreover, while recognizing the input *John saw him*, DMA records in memory this input event as a specific instance of the *mtrans*ing action. In recording this instance, a DMA parser will create a specific link from this newly recorded instance to some specific individual (instance) that was recorded in memory by previous sentences which fits the requirements

[3] Except perhaps for [42], in which DMA-type contextual information was constantly supplied even during the traditional pseudo-unification parsing.

that are stipulated by current context as well as already known semantics (world knowledge). This activity is independent of sentence boundary and indexicality (intra- or extra-sentential), and is handled naturally as a part of the understanding activity in DMA parsers.

Thus, in the DMA model, natural language understanding (parsing) is a process of dynamically changing the mental state (memory) by receiving new information as a linguistic input. It is assumed that the meaning of a sentence cannot be statically represented as a stand-alone representation, isolated from the rest of the sentences at the time of utterance and the dynamic state of the memory of the language understander. Thus, in the DMA model, there is no separate meaning representation for a specific sentence, but a whole memory network (i.e., the current state of activity of the memory network at the time of the utterance of some specific sentence) is the representation of what is being recognized.

3. A CLOSE LOOK AT DMA PARSING

We will take a close look at a typical DMA parsing method currently employed by parsers under this paradigm. Linguistic knowledge is represented in Concept Sequence Classes (CSC) which are sequences of concepts at different levels of abstraction[5] and are attached to some Concept Class node which subcategorizes for other Concept Class nodes. For example, <PERSON ACT OBJECT> is a concept sequence attached to the CC node PERSON-ACT-EVENT which could be instantiated by the sentence *John kicked him.*

World knowledge is formed as a hierarchy of Concept Class (CC) entities storing knowledge both declaratively and procedurally. The specific instances of concepts input are recorded in memory as Concept Instances that are tokens of the Concept Classes.

DMA parsing typically employs a guided spreading-activation marker passing algorithm and three markers are passed around in memory for parsing:

---

[4] One might account for this as follows:
John_i placed the telephone_j [PRO_j [near him_i]]
with PRO acting as a SUBJECT and, so, defining a governing category for *him*. Note also examples like:
John keeps a gun with him at all times
?John keeps a gun with himself at all times
One might argue that PPs in these examples are optional and attached as adjunct small clauses with a PRO in subject position. Our point here is not the discussion of adequacy/inadequacy of principles of Binding Theory on all occasions. Instead, as claimed in 1), the mechanism taken in the traditional syntactic theories (such as GB) for handling anaphoric expressions is handling them through principles/constraints to coindex some anaphoric word to some antecedent phrase (as in our example), whereas in DMA parsing, this is handled as a memory process. In DMA, interpreting anaphoric expressions is a search for a specific entity (instance) in memory that was identified with the antecedent phrase and which can also be identified with the anaphoric expression in question.

[5] These can be at low surface specific patterns such as in phrasal lexicon (see e.g. [1]) or at higher levels of abstraction such as in MOP'S in [38] (and mixtures of constituents from different levels of abstraction).

- Activation markers (A-Marker) are passed upward in the hierarchy of Concept Classes.
- Prediction markers (P-Marker) are placed on CSCs to identify the linguistic construction of the input sentence.
- Context markers (C-Marker) are placed on nodes which have contextual relevancy to prior sentences.

The following is the basic algorithm used in DMA parsing:

1. Initially the first elements of all concept sequences (CSC — Concept Sequence Class) are predicted by putting P-Markers on them.
2. A lexical node is activated by the input word. For example, the input "John" activates the lexical node JOHN.
3. An A-Marker is created and sent to the corresponding CC (Concept Class) nodes. The concept class corresponding to the lexical node JOHN is the concept class node *John*.
4. A CI (Concept Instance) under the CC is searched for.
   If the CI exists, an A-Marker is propagated to higher CC nodes. For example, if "John" has been mentioned in a previous sentence, there will be a CI node such as JOHN007 and an A-marker will be propagated upward in the CC hierarchy, activating nodes such as *male-person*.
   Else, a CI node is created under the CC, and A-Marker is propagated to higher CC nodes.
5. An A-Marker is propagated upward in the abstraction hierarchy.
6. 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. When an A-Marker reaches any contextual Root node[6], C-Markers are put on the contextual children nodes designated by the root node.
8. When the last element of a concept sequence receives an A-Marker,
   - Constraints (world and discourse knowledge) are checked for[7] to evaluate the "goodness" of the interpretation[8].
   - CSI (Concept Sequence Instance) is created under CSC with packaging links to each CI[9].
   - The memory network is modified by performing inferences stored in the root CSC which had the accepted CSC attached to it.

---

[6] A node that influences context. We will be discussing this issue in the next section.

[7] If the constraint is violated, large cost is added, if no evidence for a constraint violation is found, add small cost, if the constraint is satisfied add no cost. The cost for the interpretation is propagated by storing the information in the A-Marker. It is not in the scope of this paper to discuss constraint cost analysis in DMA parsing. [22] discusses the analyses in detail for DmTrans Plus cost-based ambiguity resolution.

[8] Analogous to the way Hobbs uses the term "goodness" based on cost in his abduction-based interpretation scheme (see [19]). Again, consult DmTrans Plus literature for the specific scheme that is adopted in the system.

[9] I.e., *concept refinement*. We will be discussing this in the later sections of this paper.

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

Readers may find our scheme of spreading activations similar to those researched by connectionists. However, we have not adopted connectionist associative architecture[10] and backpropagation in our thematic conceptual clusters. Our spreading activations are guided and we do not spread everywhere.

3.1. An Example

We will give an example of a typical DMA parsing session[11].

The input utterance is:

A fat man with a S & W in his right hand kicked open the door. Mary picked up an Uzzi. She shot the man with the hand-gun. She continued until she ran out of shells.

We will take a close look at parsing of the third sentence *She shot the man with the hand-gun* which may be ambiguous[12] without the preceding sentences. We are assuming that the hearer knows that an Uzzi is a machine gun, i.e., Uzzi was known in the memory network linked to a concept representing machine gun by an abstraction link. Also, we assume that the hearer knows that a S & W is a hand-gun. Preceded by the first and second sentences, the third sentence is unambiguous. However, if we process the third sentence alone, PP-attachment is ambiguous. Since in a DMA model (as in the actual language use situation), sentences are always in context, the ambiguity resolution of the third sentence is possible, because memory activity for processing first and second sentences are recorded in the memory.

During the processing of the first sentence, the DMA parser (DMTRANS PLUS) creates instances of 'man' and 'S & W' and records them as active instances in memory (i.e., MALE-PERSON001 and S & W001 are created). In addition, a link between MALE-PERSON001 and S & WI001 is created with POSSESS relation label. This link creation is invoked by triggering side-effects (i.e., inferences) stored in the CSC representing description of a person and instantiation of the specific instance as "a fat man with a S & W" (i.e., instantiation of PERSON-PP-DESC001 in our memory net). While during the processing of the second sentence, instances for 'Mary' and 'Uzzi' (i.e. MARY001, UZZI001) are created along with the POSSESSES link created as a side-effect of instantiation of CSC representing the action of a person picking up something (i.e., instantiation of PERSON-PTRANS-OBJ-EVENT001).

Now the first word *She* of the third sentence is input. It activates the lexical node SHE which activates the CC node FEMALE-PERSON. A specific instance of female person

[10] The connectionist associative model still lacks abilities to express complex relations between concepts and to perform variable binding (marker passing algorithm with structured markers can handle this) which are essential to handle linguistic phenomena such as metonymy as explained in TOURETZKY [1988].

[11] Taken from a sample run of DMTRANS PLUS (see [22]) which is currently developed at the Center for Machine Translation as the second generation DMA system.

[12] Globally ambiguous, as opposed to garden path (as discussed in [30]).

known in memory that is currently recorded as active is searched for and is found to be MARY001. If there is more than one such instance active, we carry multiple hypotheses here (and the parser will attempt to choose one by application of constraints in later processing). Since we only have MARY001 recorded in memory as currently active, we enhance the activation of this CI node and that triggers the activation upward in the abstraction hierarchy. The activation spreads upward and hits the CC node Person.

With the second sentence, we have three relevant concept sequences (CSC's[13]):

CSC1: <Person Shoot Physical-Object>

CSC2: <Person Shoot Physical-Object With Instrument>

CSC3: <Person With Instrument>

All first elements of concept sequences are initially predicted. When Person receives an activation from below, since it was predicted (P-Marked) by the three relevant concept sequences (along with numerous other CSC's that have Person as the first element of their sequence[14]), P-Markers are sent to the next elements of the three sequences (i.e., Shoot, Shoot[15], and With). While these markers are sent to the next elements in the concept sequences, the activation goes upward in the abstraction hierarchy concurrently (which produces no interesting effect for our look at the processing of the second sentence).

When the second word *shot* comes in, the lexical node SHOT gets activated, and activates the CC node Shoot, which was P-Marked by receiving P-Markers from Person by CSC1 and CSC2 activity. Now, P-Markers are sent to Physical-Object in turn. The third and fourth words, *the* and *man* are identified with a CSC representing a determiner noun sequence (which we omit here for the sake of simplicity) and the CC node Male-Person receives an activation. The currently active instance for Male-Person is searched for, and MALE-PERSON001 is found. It is reactivated and sends activation upward. This activation hits the CC node Physical-Object (because a human is also a physical-object in memory), which was predicted by receiving P-Markers from Shoot using CSC1 and CSC2. As for CSC1, Physical-object is the last element of the sequence, and we create a CSI as an instance of this known sequence of concepts. The CSI for CSC1 is instantiated with the specific CI nodes that originated the upward activations. A CI node is created under the root CC node with the accepted CSC attached to it. The links

[13] As 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.

[14] This is where parallelism comes in the fact and that our algorithm is massively parallel in nature, because numerous sequences are active at distributed locations of memory concurrently. An ideal hardware for DMA model is a massively parallel hardware. See [46] for discussions in this direction. Also, there has been a report of implementing DMA-type algorithm for text-retrieval in a VLSI chip (see [21]). Encoding DMA algorithm in a micro-code in a VLSI is also appealing way of implementing a very fast DMA parser.

[15] Although we have the same concept Shoot in the sequences, since they are different sequences, different P-Markes are sent for each sequence to the same CC node Shoot.

from the root CC node (i.e., PERSON-SHOOT-EVENT1, which has CSC1 attached to it) are inherited by the newly created instance node. This means that thematic roles (i.e., represented as links) are inherited from the root node with specific instances filling the slots. This action is called *concept refinement* in DMA parsing. As for CSC2, the P-Marker is now sent to WITH.

When the fifth word *with* is input, it activates WITH, since WITH was P-Marked by 1) CSC2 with P-marker sent from PHYSICAL-OBJECT (in order to attach PP *with the hand-gun* directly to VP) and 2) CSC3 with P-Marker sent from PERSON which was activated by *She* (in order to incorporate PP into larger NP). Both sequences send P-markers to INSTRUMENT. When the last two words, *the* and *hand-gun*, are input, the CC node HAND-GUN is activated, and since S&W001 is the currently active CI node under HAND-GUN, it sends activation upward in the class hierarchy. This activation hits INSTRUMENT, and both CSC2 and CSC3 sequences get accepted; concept refinement is conducted, creating specific CI nodes and CSI nodes respectively. With the recognition of CSC3, the newly created root CI node packaging the recognized input instances now sends activation upward, and hits PHYSICAL-OBJECT (again, person is also physical-object); thus whole CSC1 gets accepted.

When the concept sequences are satisfied (i.e., all elements are activated), their stored constraints are tested. The constraint for CSC2 is (POSSESSES PERSON INSTRUMENT) and for CSC3 (and CSC1 that uses CSC3) is (POSSESSES PERSON INSTRUMENT). Note that CSC2 represents the PP-attachment case when PP is attached directly to VP and CSC3 (CSC1) represents attachment to the second NP in *She shot the man with the hand-gun*.

In order for CSC2 to be consistent with what is already known in memory, (POSSESSES MARY001 HAND-GUN001) must be satisfied. Since such knowledge (link) is not recorded in memory (and nothing contradicts it either[16]) this assertion is temporarily forced and the interpretation using CSC2 is attached with certain cost[17]. On the other hand, the interpretation using CSC3 (i.e., CSC1) is attached with no cost because (POSSESSES MALE-PERSON HAND-GUN is recorded in memory. As a result, an A-Marker from CSC3 is propagated upward to CSC1 with no cost. This way, CSC1 interpretation is costly[18] and CSC2 is not. Thus, the interpretation using PP-attachment to the second NP is chosen in DMA parsing.

One other thing that is triggered during the parsing of the input sentences is C-Marker propagation. Some CC nodes that are thematically influential in the determination of contexts are designated as Contextual Root CC nodes and they send C-Marker activations[19] to nodes that are linked by thematic relation links. These nodes may or may not have lexical nodes attached to them, therefore specific vocabulary may not trigger

[16] Note that the fact that Mary possesses an Uzzi does not rule out the possibility of her possessing hand-gun as well.

[17] The reason that DMTRANS PLUS does not reject the interpretation using CSC2 is that the partial information, about MARY001's possession does not contradict the CSC2 interpretation so it simply forces the interpretation with the cost for the assumption without known record backing it up.

[18] For details of cost-based ambiguity resolution strategy, see [22].

[19] Which weakens with passage of time which is true with lexical (A-Marker) activation also.

contextual activation; however, the recognition of a whole concept sequence may trigger such an activation[20]. HAND-GUN and MACHINE-GUN are examples of such contextual root nodes, and when they received activations from below, they sent C-markers to their thematically packaged nodes. Among them was the concept BULLET, which received C-Markers twice from HAND-GUN and MACHINE-GUN.

While the fourth sentence *She continued until she ran out of shells* is processed, the fact that BULLET is C-Marked is used in order to solve the lexical ambiguity of *shells*. Among the many interpretations of *shells* only the interpretation of BULLET has received a C-Marker, and the interpretation with the highest C-Activation is automatically chosen[21] (although other hypotheses are also carried). Also, in processing the fourth sentence, we have the phenomenon of VP ellipsis. This is handled by simply picking up the most active (recently activated) root CI node (i.e, MARY-SHOOT-UZZI-EVENT001, recorded as an instance of 'person shooting' at acceptance of CSC2), which is a default action for VP (ACT) ellipsis, and checking consistency with the rest of the role fillers (including the newly hypothesized BULLET using C-Marker activation).

We described in some detail the mechanism of DMA parsing, using the example of DMTRANS PLUS parsing. In the following sections, we will look at DMA used in three specific application areas, namely, 1) machine translation, 2) speech-understanding, and 3) natural language interfaces.

PART II

MACHINE TRANSLATION UNDER DMA

As our first application of DMA paradigm in natural language processing, we developed a machine translation system utilizing the model. The Direct Memory Access Translation (DM-TRANS) is a DMA based MT system developed at CMT (see [42] and [43]). Parser part of DMTRANS utilizes the guided spreading activation algorithm described in the previous section. The cost-base ambiguity resolution was added while upgrading the system to the second generation DMTRANS PLUS system and this section does not include the discussion of the scheme.

We prefer DMA as paradigm for machine translation, because translation is performed directly through the network of memory, which makes dynamic interaction with other memory-related process possible, and because all previously created memory structures can potentially participate in translation. DMTRANS extends and integrates theories of direct memory access understanding into translation with consideration of cross-cultural questions that accompany the attempt. We view translation as locating existing memory structures under the source language that the text is referring to and generating text that refers to these memory structures in the target language.

Often, a single memory structure is not shared by different languages and in that case, use of similar existing memory structures and explanation by surrounding memory

---

[20] This is analogous to the way that MOP-based systems (such as FRUMP, see [9]) activated relevant MOP's without simple keyword matching.

[21] Just as people are typically unaware of the alternatives, see [30].

structures replace direct generation from identified memory structures. Currently, the system is developed to translate between English and Japanese and is capable of understanding and generating fairly complex sentences between the two languages.

4. WHERE MOST MT SYSTEMS FAIL

4.1. Syntactic and Semantic Ambiguity

Because most MT systems do not understand what they are translating, they are incapable of making decisions based on the content of the material they are translating. For example, sentences such as *I saw a man with a telescope* may be handled by current systems by representing multiple interpretations of the input; however, this does not mean these systems are capable of handling ambiguous sentences, since none of these systems are capable of choosing the most correct interpretation over the others. This makes an autonomous translation extremely unlikely, because very often sentences can have multiple interpretations (most of which, humans are unaware of); without human assistance, such systems are incapable of selecting one interpretation over others[22]. Thus, being able to generate all possible interpretations of an input sentence does not automatically mean the system is capable of handling syntactically ambiguous sentences. We claim that the system should be able to select the correct interpretation (what the speaker intended) in order to claim that it "handles" such a sentence. Unfortunately, most current MT systems fail in this task.

By the same token, most MT systems fail in handling semantically (lexically) ambiguous sentences. Consider the examples: *The quality of this paper is terrible* and *John gave Mary a punch*. In the former example, the interpretation of *paper* should be different (for example, Japanese for 'thesis' and 'a sheet of paper' is different) according to what has been said before (or perhaps, visual perception of the situation may supply help). In the latter sentence, interpretation should be different again due to the context (Japanese for punch as 'hitting' and punch as 'a drink' is different). Again, being able to generate multiple interpretations of sentences does not mean the system is capable of handling semantically ambiguous sentences. The system instead should be able to choose appropriate interpretations.

4.2. Ellipsis and Anaphora

In most MT systems, ellipsis in a sentence results in either no parse at all or output with missing slots. For example, in translating *kouryo suru to ittaga, totemo shinjigatai* ([he] said, [he] will consider [it], but [I] can hardly believe [it]) which is a typical Japanese sentence with missing subjects, most MT systems simply fail in filling in missing informa-

[22]This problem is conspicuous when a sentence has a fairly complex structure including conjuncts. Consider "Show me the picture of lung with small cell carcinoma with magnification of ten and the brain with squamous cell carcinoma with magnification of five".

tion[23]. Another example is *How often does squamous cell carcinoma metastasize to the brain? Lung? Large cell carcinoma?*. Unless MT systems perform some strong inference at run-time, it is beyond their capacity to handle this phenomenon. Since few conventional MT systems are performing any kind of contextual inferences at runtime and normally the representation structures that are built during the translation of one sentence are either lost or not used in any meaningful way during the translation of other sentences, ellipses are hard problems for these systems. Actually, since filling in missing information requires the understanding of the text and the contextual knowledge, any inference that hopes to solve this problem needs to be memory based.

Anaphoric expressions are another kind of phenomenon that most MT systems fail to handle. Consider the example of *Leia threw a long sword at the giant rat. It ate it.* Current MT systems are satisfied with translating 'it' as 'it'[24]; however, this often creates problems: for example, Japanese does not prefer *sore* (it) for animate objects whereas English refers to both animate and inaminate objects with it. In some languages, the morphology of 'it' changes according to what it is referring to. Even if the MT systems decide to output 'it' as 'it' unless they do so with knowing what 'it' is referring to, there is a danger of causing errors in translations without noticing that they mis-translated the input.

5. CONTEXTUAL RECOGNITION OF CONCEPTS

DMTRANS outperforms most systems in choosing an appropriate interpretation of sentences over others in accordance with contexts. This is possible because sentences are always recognized in context in DMTRANS, by performing strong predictions based on what has been recognized previously.

In DMTRANS, the contextual recognition of concepts is performed through 1) the use of CSC recognitions that are organized in thematic role relationships, 2) use of C-Marker propagations. Since memory activity records the specific instances of input (through lexically triggered CI creations or concept refinements), concept sequences are always instantiated in context. In other words, in DMA parsing, CSC recognition guarantees that recognitions are always performed based upon preceding phrases and sentences.

Also, the C-Marker propagation mechanism helps to resolve ambiguities in texts, especially when an input word has multiple meanings and also when the multiple interpretations of an input text may be solvable through the context that was established relatively recently[25]. When activation is spread upward in the abstraction hierarchy and

---

[23] Simple heuristics such as "assume the missing subject to be the subject of the former clause' does not work here.

[24] As long as 'it' is translated as 'it' (perhaps 'sore' in Japanese), the translation is treated as accurate in most systems.

[25] Which is often the case with the ambiguities that most MT systems are currently avoiding to handle. When the context was not established relatively recently, i. e., if the context is the result of larger conceptual framework, then the C-Marking may not always help. In such a case, the top-down predictions through the higher level MOP structures are more effective than the use of Context Marker passing.

if more than one route exists (such as two meanings for a word), then the route through the C-Marked concepts is chosen unless the route hits a higher level concept that indicates a contrary preference.

5.0.1. Examining Our Sample Translation

In order to demonstrate this mechanism, let us examine a short translation of a semantically (word-sense) ambiguous sentence: *John is at IJCAI-87. He said the quality of the paper is terrible* (Figure 1). Initially, all the first elements of concept sequences (indicated by <...>) are predicted. The first word *John* comes in and activates the concept JOHN (put A-Marker on it) then the A-Marker is sent upward until it hits the concept PERSON which is predicted by AT-PERSON-LOC as the first element of the sequence. Then the prediction is sent to IS which gets activated by receiving A-Marker from the next input word *is*. Then AT is predicted as the third element of the sequence which meets activation from the input *at*. Then the prediction for LOCATION is made. When the word *IJCAI-87* comes in, and activates IJCAI-87 and then LOCATION (IJCAI-87 has two immediate ancestors: ACADEMIC-CONFERENCE and LOCATION) which was predicted as the last element of the concept sequence: <PERSON IS AT LOCATION>, this concept sequence is accepted and the root-concept AT-PERSON-LOC gets activated. Then the search is performed to find a specific concept under the root concept that indicates the input[26], and a concept refinement is conducted to get to AT-JOHN-IJCAI-87. If this is not found, DMTRANS creates this concept as a specific episode of AT-PERSON-LOC. At the same time, since ACADEMIC-CONFERENCE (activated by IJCAI-87) is a contextual-root concept it sends C-Markers to PERSON-PRESENT-THESIS, PERSON-CRITICIZE-THESIS, THESIS, PROCEEDINGS, etc.. When the next word *He* comes in, it sends activation upward and finds that the only male person activated in memory is JOHN, and activates JOHN again; PERSON gets re-activated, which is predicted as the first element of MTRANS-EVENT, then *said* comes in and fits as the second element of the concept sequence attached to MTRANS-EVENT. Likewise, *The quality of the paper is terrible* is accepted, being identified with the sequence <FEATURE-TYPE OF OBJECT IS FEATURE-VALUE> attached to Object-Description.

5.0.2. Contextual Choices

One thing that happens is that when *paper* which is attached both to PAPER and THESIS comes in, only THESIS sends activation upward because THESIS was C-Marked by ACADEMIC-CONFERENCE and PAPER was not marked. This choice is not challenged when MTRANS-EVENT is accepted and is concept-refined to PERSON-CRITICIZE-THESIS-EVENT,

[26] Concept refinement in DMTRANS is performed as a search for a node that packages the input recognized concept with links parallel to the links from the accepted root node to the elements of the accepted concept sequence.

since this concept also supports the contextual interpretation of *paper*[27]. This way, understanding is left as activated memory structures representing AT-JOHN-IJCAI-87 and JOHN-CRITICIZE-QUALITY-OF-THESIS-EVENT that are instances of the refined concepts under accepted root concepts. Also, if two conflicting choices of a concept are marked by two C-Markers, the C-Marker put by the concept activated more recently gets preference. For example, in *John was writing a letter on a plane to IJCAI-87. The ink smeared. He said the quality of this paper is terrible* and in *John was printing a paper for IJCAI-87. The printer jammed. He said the quality of this paper is terrible*, both PAPER and THESIS are C-Marked by IJCAI-87[28] and INK, IJCAI-87 and PRINTER, respectively[29]. However, since, *ink* and *printer* both come after IJCAI-87 in both cases, PAPER is preferred over THESIS in both cases, and it gets activated. Unless these activations meet contradicting hypotheses elsewhere, PAPER becomes the contextual interpretation of *paper*.

5.1. Explanatory Generation

DMTRANS is capable of generating output through the mechanism of explanatory generation which can handle translation of culturally sensitive sentences and the concepts that do not have counterpart lexical entries in the target languages.

5.1.1. Multiple Concept Sequences

We have two different concept sequences stored in each root concept, one for English and one for Japanese[30]. Especially because they represent texts from different language families, the sequences are rarely the same; however, the roles are shared, it is because memory structures are independent of languages and the types of roles are inherent in the root concepts, not in the languages. Similar approaches are taken in MOPTRANS [24] and CMU's current generation MT system [49]. Both systems take advantage of shared memory structures for translation, the former using MOPs as the shared structure and the latter using case frames as the shared structure.

5.1.2. Generation Mechanism

Generation begins with the result of memory activation parsing from input in one language. For each concept refined nodes left in memory, we do the following. 1) Check at the lexical node for the refined concept in the target language and if a lexical entry is found, generate in accordance with templates stored with the concept and we are done. 2) If not, which is often the case[31], we generate according to the stored concept sequence for the target language. That is to generate from the first element of the concept sequences (go back to 1 with the first element of the concept sequence). 3) Since not all concepts have a sequence attached to it, search the abstraction hierarchy upward for

[27] C-Marked by the same contextual root concept as THESIS

[28] Actually, C-Marked by ACADEMIC-CONFERENCE which was activated by IJCAI-87.

[29] These three concepts trigger (activate) contextual-root concepts.

abstraction of the refined concept which has concept sequences attached to it. 4) Get the sequence from this abstraction and then instantiate with the roles in the refined-concepts. Then from the first element of the instantiated concepts sequence, repeat from looking up lexical node again. If not found, repeat from the 2 again to explain this concept.

5.1.3. Examining Our Sample Translation

One sample short translation is translating the Japanese sentence: *Gionshoja no kane no koe, shogyomujo no hibiki ari*[32] which is translated to be: "The sound of bell at Gionshoja has the tone of *shogyomujo* (impermanence of all phenomena in world)". The result of understanding by DmTrans leaves the two concepts (instances) in memory that are: SOUND-OF-BELL-AT-GIONSHOJA and EXISTS-TONE-OF-SHOGYO-MUJO. In order to generate the first concept in English, it looks for the conceptual root concept above SOUND-OF-BELL-AT-GIONSHOJA and finds SOUND-OF-INSTRUMENT which has the sequence <SOUND OF MUSICAL-INSTRUMENT[33]> attached to it. We instantiate this sequence by the concepts packaged in SOUND-OF-BELL-AT-GIONSHOJA and get <SOUND OF BELL-AT-GIONSHOJA>. By the same token, generate BELL-AT-GIONSHOJA by explaining it through the packaged concepts (that are neighbours in the linking relations) found in the concept sequences attached to the ancestor concepts and get <BELL AT GIONSHOJA>. For the second concept left as the result of understanding: EXISTS-TONE-OF-SHOGYO-MUJO, we apply the same generation mechanism. First search the concept sequence attached to the ancestor of the EXIST-TONE-OF-SHOGYO-MUJO which is EXISTS-FEATURE-TYPE-OF-SOUND and return <HAS THE FEATURE-TYPE-OF-SOUND> and instantiate it to be: <HAS THE TONE-OF-SHOGYO-MUJO>. Then generate TONE-OF-SHOGYO-MUJO explaining <TONE OF SHOGYO-MUJO>. Here SHOGYO-MUJO is a concept peculiar to the Japanese culture (no corresponding English terms); however, since it is integrated into our memory network, it can be explained using the same generation mechanism. We get to its ancestor IMPERMANENCE-OF-ALL-PHENOMENA and return <IMPERMANENCE OF ALL PHENOMENA> and generate this in English.

Note that DmTrans outputs *shogyomujo* as *shogyomujo*, and adds the explanation of

---

[30] Actually, we may have multiple concept sequences attached to a concept within a language instead of one for each language.

[31] This is the inherent uniqueness of the DmTrans system, that the system does not halt even if the lexical entry is not found in the target language; instead DmTrans tries to explain the concept through surrounding concepts in the memory network that have lexical entries in the target language.

[32] From *Heikemonogatari* written around 1210.

[33] Never mind even if the categorization of the 'bell at a Buddhist temple' to be a musical instrument sounds controversial. This is how we categorize in our memory network and the parser recognized accordingly. In other words, we could categorize the BELL to be something else and the same generation mechanism can handle the explanatory generation using the different definition of the concept.

the word in parentheses. This is because an English lexical entry for the concept representing SHOGYO-MUJO was not found in memory and we know that the phrase in parentheses is the close meaning of the word "shogyomujo". This mechanism is much more desirable than the behaviours of many current MT systems in which they either halt execution with input words without corresponding target language vocabulary or simply output the original source words (without any attempts to explain). Since a concept may not be shared across languages, this type of translation happens often, especially in the cross-cultural context[34].

The strength of the DmTrans generation mechanism is that since generation is performed directly from the state of the memory network left as the recognition of the source text, i.e., the understanding of the input text, it can generate the output in the target language using the concepts that are available in the target language and therefore, existence, or lack thereof, of the counterpart vocabularies for the input words does not change the performance of the translation.

5.2. Dynamic Interactions with the Rest of Cognition

Since translation is performed by directly accessing the memory network, other faculties of cognition can dynamically participate in translation. With *Leia threw an apple at the giant rat. It ate it.* Whenever a pronoun comes in as an input, DmTrans tries to identify the object that is referred to[35]. In this example, the concept ANIMAL-INGEST-OBJECT-EVENT gets activated by the input *it ate it.* ANIMAL-INGEST-OBJECT-EVENT is a memory structure[36] (CC node) which is a kind of INGEST-EVENT. It has two roles to be filled: Actor and Object. In order to determine the Actor, the inference mechanism is activated and it looks for activated concepts in memory that can be an Actor and finds GIANT-RAT to be a candidate given restriction set forth by the memory structure[37]. Then a search is made for concepts previously activated in memory that fit the requirements for Objects and APPLE is selected to be an acceptable object of INGEST-EVENT. This example only requires a minimum amount of work for deciding objects; however, this architecture allows for deeper inferences if necessary, such as utilizing causal relations[38] stored in the CC nodes and constraint cost analysis (as performed in DmTrans Plus).

---

[34] The described explanatory generation mechanism works effectively in translation between English and Japanese, where a one to one match of concepts is often difficult to find due to the difference in the cultural contexts. Even words such as *river* and *kawa* (Japanese for river) which are normally substituted for one another without any further consideration, reveal the difference in concepts attached to them, i.e., the Japanese word *kawa* is normally associated with images of clear rapid streams. What about *kou* in Chinese?

[35] This is independent of the question whether to translate IT as IT. Even if we do, it is better to know what is referred by it with the reasons indicated before.

[36] MOP-based memory structure as indicated previously.

[37] If *John* is known to be a name of a dog, we need more inference. Such as check the previously activated memory structure (propel-event) and infer where the apple is now, etc..

[38] And also, such knowledge as Explanation Patterns (XPs) associated with higher level structures, see [39]. Actually, the parser part of DmTrans was originally designed as an integrated part of a case-based reasoning system to allow direct inference on input sentences.

5.3. Translation system that acquires vocabulary

DmTrans is capable of creating new concepts while translating, and is capable of learning new vocabulary for newly created concepts in a multi-lingual context. When a concept refinement is performed, if a specific concept representing the input sentence is not found underneath the accepted root concept, a new specialization is created. Also, the user of the system is asked to input the English and Japanese names (words) for the concept (or input phrase can simply be stored as a phrasal lexicon). By the same token, we can simply assert facts to be translated by DmTrans and the system stores the assertion as well as it translates it as long as it is not incompatible with what it already knows. At the same time, the acquired concept is accessible from different contexts because of the hierarchical organization of memory. This way DmTrans implements dynamic memory as its memory network and is capable of learning while translating.

PART III

SPEECH UNDERSTANDING SYSTEM UNDER DMA

Now we will take a look at the speech understanding system ΦDmTrans (Tomabechi, et al. [46]) developed at CMT utilizing DMA paradigm.

Recently a few efforts have been made in the area of processing speech input to a natural language understanding system. These include [13], [48], [37], [44], and [13].

Among them, Tomabechi & Tomita and Hayes, et al. use contextual information for disambiguation of speech inputs and therefore, since extra-sentential information is important in the speech input system, ΦDmTrans shares this feature of the two systems. The uniqueness of ΦDmTrans, however, is that:

- it uses a parallel spreading activation network from the phonetical level,
- morphophonetic and phonological knowledge is dynamically utilized during memory activity,
- the morphophonemic, episodic/thematic and pragmatic levels of processing are fully integrated.

6. PROBLEMS IN SPEECH INPUT

6.1. Phonetics, Phonology and Morphology

The difficulty of parsing speech input is that unlike written text input, a parser receives multiple hypotheses as input for a particular voice input. This is partly due to current limitations on speech recognition systems, which are incapable of determining specific phonemes for each input and generally produce several possible segmentations of the hypothesized phonetic stream. It is not rare that a speech parser outputs 30 to 50 well-formed, semantically acceptable parse results for each independent sentence of a speech recognition device output.

For example, when testing the CMU-CMT speech parser (a phoneme-based

Generalized-LR parser (ΦGLR, see [37]), the Japanese input "atamagaitai" ("I have a headache") was spoken into a speech recognition system[39] (under ordinary office environment) and accepted by the integrated[40] parser with 57 ambiguous interpretations. Each of the ambiguous interpretations is semantically legitimate, meeting the local restrictions set forth by case-frame instantiation restrictions. Below are some of the highly scored interpretations:

These are just some of the 57 disambiguations that were produced as acceptable readings by the speech understanding system given the input "atamagaitai". One problem that is typified here by the ΦGLR speech parser, and commonly shared by most existing speech understanding systems, is that these systems do not sufficiently utilize morphophonetic and phonological knowledge during recognition and understanding. We will be discussing such knowledge in Section 4, but to be precise, it is the kind of knowledge that, for example, dictates what type of phonetic and phonological variation are possible for each type of phonetic features specific to Japanese. Humans apparently utilize such knowledge in processing a sequence of phones, and we would like to model such processing, since speech input is not a sequence of independently-determined phones but a connected string of successive phones.

6.2. Need for Contextual Knowledge in Speech Understanding

As we have seen in the preceding subsection, even with the semantic restrictions set forth by a syntax/semantics parser, we suffer from the problem of ambiguities that do not arise when the complete text is considered (i.e., 57 interpretations of "atamagaitai" in the preceding subsection were all acceptable syntactically and semantically only when not considering the context). This problem increases when the vocabulary of the speech understanding system enlarges and the variety of sentences that are accepted by the system expands. Although possible morphophonemic analyses of the speech input may be narrowed with the use of phonetic and phonological knowledge during speech understanding, we will still have a large number of ambiguities for a specific phonetic stream.

In other words, local semantic restriction checks and phonetic/phonological narrowings are not sufficient for disambiguating continuous speech input, since an interpretation can be totally legitimate phonologically, syntactically, and semantically, but can mean something drastically different from what has been input into the speech recognition system (as well as being contextually inappropriate). The speech understanding system needs extra-sentential knowledge to choose an appropriate hypothesis for grouping phonetic segments and for selecting the appropriate word-sense of lexical entries. That is to say that the need for contextual knowledge in speech understanding systems is even more urgent than in text input understanding systems; in a speech understanding system, the input can be interpreted in a way that is not possible in text input systems,

[39] Matsushita Research Institute's speech recognition hardware. The speech recognition system and the speech input enhanced LR parser are described in detail in [37].

[40] By 'integrated', we mean concurrent processing of syntax and semantics during parsing as opposed to some parsing methods where syntax and semantics are separately processed.

and the input can still be acceptable to the local semantic restriction checks that integrated parsers perform within a sentence (such as slot-filler restriction checks of case-frame parsers).

7. PHONOLOGICAL KNOWLEDGE IN ΦDmTrans

Phonological knowledge is represented in ΦDmTrans as weighted links connecting phonetic and phonemic nodes and functions stored in phonetic nodes capturing the physical and acoustic properties of sounds in a language (distinctive features) as well as environments that dynamically affect phonetic alterations. Phonological knowledge is used for providing the information to identify physical properties of articulated sounds instead of mental representations of each segment of words.

We represent distinctive features of Japanese sound using a distinctive feature system using SPE (Chomsky & Halle) and encoding them into a network using weighted links storing the phonemic distances based on the feature matrix (Appendix 2). We assume that lower phonemic distance results in higher confusion probabilities (i.e., higher weights). One thing to be noted here is that we use existing speech recognition hardware[41] which outputs sequence of phonemes as hypothesis for input speech. Thus input to ΦDmTrans is a sequence of phonemes (may be noisy, with added, dropped, or altered phonemes) instead of raw speech input data. Thus, ΦDmTrans should not be confused with neural-net based speech recognition systems although interface to such a system is currently being considered in [52].

The utilization of distinctive feature matrices, however, is a static knowledge that is encoded initially to the network (before parsing). We also need a scheme to dynamically assess the confusion of phones depending upon the phonetic environments that appear in the input speech. In Japanese, some speakers produce a glottal stop in a word initially before a vowel. In some speech recognition systems, the glottal stop may be interpreted as some voiceless stops, most likely /k/ because it is closer than others. Also, high vowels becoming voiceless between voiceless consonants or after a voiceless consonant in the word final position is another well known example in Japanese.

The method of capturing these types of phonological rules in our system is that we initially provide phonological environments and rules in a declarative form and the system precompiles the knowledge into functions stored in the phonetic nodes locally that are assessed every time the node is activated[42] so that the phonemic activations are

---

[41] Built by Matsushita Research Institute (see [26]). Since we use Matsushita's recognition hardware, we adopt the phonemic system that the hardware recognizes. However, we have to note that some segments are not phonemes but are allophonic variants.

[42] The functions are stored as daemons in the nodes that are implemented via 'FrameKit' representations. For example, with the voiceless vowel between voiceless consonants example, the rule is originally supplied declaratively and then the declarative rule is precompiled as functions to be evaluated and stored locally in the phonetic node representing the voiceless vowel. At parsing time, when the voiceless vowel is hypothesized by the speech recognition hardware, i.e., receives the activation (A-Marker), then the functions stored in the node as the daemons are triggered and

dynamically modified depending upon the phonetic environments on the speech input independent of the confusion matrices described above. This kind of phonological knowledge is thus encoded in the network for the dynamic phonetic activation changes, as well as the static confusion matrices that are pre-supplied and encoded as weighted links of the network along with the phonemic distances.

8. UNDERSTANDING IN ΦDMTRANS

8.1. Phone Level Activity

ΦDMTRANS is the first DMA parser that works at the phonetic level. We will discuss the scheme of phonetic and phonological recognition in this subsection. First, ΦDMTRANS has as its nodes in the memory network nodes for phones and phonemes in each language. A phoneme may be realized as different phones in different phonetic environments. Several different phones may represent the same phoneme, for example phone [e] after dental and alveolar stops and affricates may represent phoneme /a/, in addition to phone [a] representing the phoneme /a/ in ordinary environments. In our memory network, each phone is connected to phonemes they represent via abstraction links. Also, each phoneme is connected by weighted phonological relation links to other phonemes. The weights of the links are determined by the strength of phonemic closeness based upon phonological distinctive feature thresholds as described in Section 4.

Above the phonemic nodes in the abstraction hierarchy are the lexical nodes, representing words. We have each lexical node in the memory network containing the phonemic sequence realizing the lexical entry in the given language. For example, in Japanese the lexical node "atama" (head) has the list <a t a m a> attached to it. So the structure linking phonetic node to lexical node is like this:

```
>when i talk my throat hurts

(((:CFNAME *HAVE-A-SYMPTOM) (:MOOD DEC)
  (:ASSOCIATED-ACTION
   ((:CFNAME *TALK) (:MOOD DEC)
    (:AGENT
     ((:CFNAME *PATIENT) (:HUMAN +) (:PRO +) (:NUMBER SG)
      (:PERSON 1)))
    (:TIME PRESENT)))
  (:SYMPTOM
   ((:CFNAME *PAIN)
    (:LOCATION
     ((:CFNAME *BODY-PART)
      (:POSSESSIVE
       ((:CFNAME *HUMAN) (:NUMBER SG) (:PERSON 1) (:PRO +)))
      (:NAME *THROAT)))))
  (:TIME PRESENT)))
```

checks the environment (a lazy evaluation is used to attain the evaluation for both preceding and following nodes) and if the environment matches the precompiled knowledge for the voiceless vowel between voiceless consonants, then the voiced vowel phonetic nodes (i.e., [i] and [u] for Japanese) get activated and send activation to their phonemic nodes instead of activating the phonemic node for voiceless vowel.

We use A-Markers and P-Markers passed around in memory. P-Markers are passed along the phonemic sequences and A-Markers are passed above in the abstraction hierarchy (i.e., from phone to phoneme). The basic algorithm of marker passing is as described in the previous sections for text-based DMA parsers. The algorithm for phonetic recognition is as follows. At the beginning of recognition, all the first elements of the phonemic sequences (such as /a/) are P-Marked by lexical nodes.

1. When the first input phone comes in (with this example, [a]) we put an A-Marker on (A-Mark) the phone node representing the phone (the node [a]).
2. When a node receives an A-Marker (i.e., if A-Marked) it sends an activation to (A-Marks) the node in its abstraction (i.e., phoneme /a/).
3. When an A-Marker and P-Marker meet, send a P-Marker to the next element of the sequence (i.e., since /a/ was P-Marked by the lexical node "atama", it sends a P-Marker in turn to /t/).
4. When the whole sequence is activated, then activate the root of the sequence (i.e., by repeating from 1. for [t], [a], [m], [a], the phonemic sequence <a t a m a> gets accepted and then we activate the lexical node "atama").

When a certain phone (such as [t]) is activated, it not only activates its abstraction (such as the phoneme /t/) but also activates other phonemes that are related by the weighted links exceeding the given threshold. The weight of the phonological relation link is based upon distinctive feature study of each phone in the given language. For example, in Japanese the phoneme /t/ has the distinctive features 'alveolar' and 'stop' shared with the phoneme /d/, and link weight of 8 between them. So, if the threshold is given to be 5, when phone [t] is activated, both phonemes /t/ and /d/ are activated. This way, the phonological knowledge is encoded in the memory network as weighted links and is utilized during the spreading activation. Also, if the activated node contains the phonological rule application functions (i.e., stored as daemons, see footnote 40), and if the evaluation applies the rule and performs the dynamic alteration of the currently active phonetic node, then the phonemic nodes of the altered phone are activated capturing the phonetic changes in different environments which are not expressed in the static weighted links. Of course, because we have many lexical entries that share similarity in attached phonemic sequences, and also because of activation of allophones (i.e., as we have seen both [a], and [e] may be under /a/), we have quite a significant number of simultaneously active phonemic sequences for a given stream of phones. This is where the strength of the parallel nature of our spreading activation mechanism is demonstrated, as we are assuming a massively parallel network for effective implementation of the phoneme-based DMA parser.

8.2. Word Level and Sentential Level Activity

After a lexical node is activated through the acceptance of a whole phonemic sequence attached to a lexical node, we have similar spreading activations at the word level. We will omit the details of this processing in this section because ΦDMTRANS shares the same DMA algorithm at word and sentential level recognitions with other DMA parsers described in this paper.

A brief example here is the processing of the sentence "atamagaitai", which we saw before as a problematic input to other speech understanding systems. We have nodes such as *HAVE-A-PAIN (representing the concept having a pain) and concept sequence including <*BODY-LOCATION *PP[GA] *PAIN-SPEC> attached to the node[43].

Whenever a phonemic sequence such as <a t a m a> is accepted it activates lexical nodes (such as *atama*) and the DMA parsing continues as we have already seen in the previous sections. When the whole <*BODY-LOCATION *PP[GA] *PAIN-SPEC> CSC sequence is accepted, concept refinement is performed and the instance is created under *HAVE-A-PAIN which will specifically package the instances of the lexically activated CC nodes. After the parse, the generator generates the output from the current state of the memory (as we have seen in the DmTrans section[44]).

9. OTHER COMPONENTS OF ΦDmTrans

We have focused our discussion in this paper on the method our system uses to handle the phonetic input stream as part of an understanding system. ΦDmTrans is a machine translation system that works on speech inputs and we will briefly describe other parts of the system. In essence, our system consists of three parts:

- Speech recognition hardware and control programs
- An understanding module utilizing the spreading activation mechanism
- A generation module that utilizes explanatory generation.

The Speech recognition hardware is supplied through the courtesy of Matsushita Research Institute, and provides high-speed speaker-independent speech recognition. The details of this hardware are described in [26] and [14]. The understanding module that we have described in this paper receives a hypothetical stream of phones and performs the spreading activation marker passing memory activity as an understanding of the input. The result of the understanding is what is left in memory after the activation of memory stabilizes. Generation is performed directly from the state of the memory after the understanding. The generation mechanism that is used for DmTrans is used in ΦDmTrans.

10. FUTURE POSSIBILITIES WITH ΦDmTrans

We have seen the parser part of ΦDmTrans in detail which essentially is a DMA parser that performs spreading activation guided marker passing from the phonetic level.

---

[43] *PP[GA] is a syntactic category representing the post-position "ga". This way, we can integrate syntactic knowledge as in subcategorization lists in syntactic theories as well. '*' preceding a concept name indicates that it is a CC node, in order to distinguish it from phonetic/phonemic nodes.

[44] Since ΦDmTrans is based on DmTrans Plus which is not covered in this paper, there are some other activities that are performed during and after the parse; however, discussions of such activities are not the topic of this paper.

Combined with the DmTrans generator, ΦDmTrans is a translation system and with the appropriate speech synthesis hardware added (we utilize DECtalk[45] at CMT), the system is a speech to speech translation system with strong contextual understanding capability. Machine translation, however, is not the sole viable area of adopting ΦDmTrans architecture for speech understanding. For example, CMT has developed a natural language interface system based on DMA architecture (DM-Command, described in next section), the parser of which ΦDmTrans can replace to make it a speech command and query system. With the fast processing through the spreading activation algorithm and the strong contextual understanding capability, the system is a viable alternative to existing speech understanding systems particularly under noisy environment and for pragmatically difficult inputs.

PART IV

NATURAL LANGUAGE INTERFACE UNDER DMA

As one practical application of DMA, we have developed a natural language interface for our large-scale knowledge-based machine translation system[46] called DM-Command. This application of DMA also demonstrates the power of this model, since direct access to memory during parsing allows dynamic evaluation of input commands and question answering without running separate inferential processes, while dynamically utilizing the MT system's already existing domain knowledge sources. The implementation of the DMA natural language system has been completed and is used for development of actual grammars, domain knowledge-bases, and syntax/semantics mapping rules by the researchers at CMT. This system has been demonstrated to be effective as an MT developmental support system, since researchers who develop these individual knowledge sources are otherwise unknowledgeable about the internal implementation of the MT system. The DMA natural language interface can provide access to the system's internal functions through natural language command and query inputs. This use of the DMA model for natural language interfaces demonstrates that it is an effective alternative to other natural language interface schemes.

11. DM-COMMAND

The DM-Command system which we describe in this section is a natural language interface developed for grammar, knowledge-base, and syntax/semantics mapping rule writers at CMT, which enables these researchers to access the MT system's internal functions for their development and debugging purposes. The DM-Command parser borrows the basic algorithm from the DmTrans machine translation system, which

[45] DECtalk Model DTC01-AA by Digital Equipment Corporation.

[46] The CMU-MT system which is the target system for the DM-Command system described in this paper is described in detail in [49] and [25].

performs recognition of input via the guided spreading activation marker-passing of A-markers, P-markers and C-markers in memory.

As a brief example, let us consider processing the input command *show me* *HAVE-A-PAIN, where *HAVE-A-PAIN is an actual name of a concept definition in our frame system (FRAMEKIT VERS. 2, see [29]). Independent of the semantic network of domain knowledge used by the MT system, the DM-COMMAND has separate memory network representing concepts involved in performing various actions in the MT system. Among such concepts is the concept 'show-frame', which represents the action of pretty-printing FRAMEKIT definitions stored as domain knowledge. This concept has the concept sequence <mtrans-word person *CONCEPT> attached to it. This concept sequence predicts that the first input word may point to an instance of 'mtrans-word' (such as 'show'), followed by an instance of person followed by some concept in the form of a FRAMEKIT name. When the first input word *Show* comes in, it activates (puts an A-marker on) the lexical node 'show', which in turn sends activation (A-marker) above in the abstraction hierarchy and hits 'mtrans-word'.

With a P-A-collision, a P-marker is sent to the next element of concept sequence, which is *Person*. Then the next word, *me*, activates the lexical node '1st-person' and then activates *Person* (an A-marker is sent above in the abstraction hierarchy). Since *Person* was P-marked at a previous marker collision at 'mtrans-word', another collision occurs here. Therefore, a P-marker is again sent to the next element of the concept sequence, which is '*CONCEPT'. Finally, *HAVE-A-PAIN comes in. Now, the spreading activation occurs not in the command memory network, but in the domain knowledge network (doctor/patient dialog domain) activating '*HAVE-A-PAIN' initially[47] and then activating the concepts above it (e.g., '*HAVE-A-SYMPTOM') until the activation hits the concept '*CONCEPT' which was P-marked at the previous collision. Since it is the final element of the concept sequence <mtrans-word person *CONCEPT>, this concept sequence is accepted when this collision of A-marker and P-marker happens. When a whole concept sequence is accepted, we activate the root node for the sequence, which in this case is the concept 'show-frame'. Also, in addition to activating this concept, we perform *concept refinement*, which searches for a specific node in the command network that represents our input sentence. Since it does not exist in this first parse, DM-COMMAND creates that concept[48]. This newly created concept is an instance of 'mtrans-

[47] One thing to note here is that the concept '*HAVE-A-PAIN' that is activated by input *HAVE-A-PAIN is not part of the memory network for the DM-COMMAND's MT system commanding concepts, instead it is a memory unit that is a part of the MT systems domain knowledge, in other words '*HAVE-A-PAIN' belongs to a different memory network from 'show-frame', 'mtrans-word', and *Person*. This does not cause a problem to the DM-COMMAND, and actually, it can utilize any number of independent semantic networks simultaneously, as long as concept sequences guide passing of P-marker from one network to another. For example, the '*PERSON' in the domain knowledge semantic network represents some generic person, whereas *Person* in DM-COMMAND command knowledge network represents persons involved in the use of the DM-COMMAND system.

[48] In DMTRANS, when such creation of concepts occurred the user was asked to provide the vocabulary, and thus served as a model for vocabulary acquisition as well as concept creation. In DM-COMMAND, we randomly generate names for such newly created instances and user does not supply names for the newly created concepts.

frame', and its object slot is now filled not by generic '*CONCEPT' but instead by '*HAVE-A-PAIN', specific to our input sentence. This final concept-refined concept is the result of the parse.

For the actual evaluation of an action, DM-COMMAND triggers functions that are stored in the concept which is located or created after the parse. The specific functions for triggering the commands are stored in root concepts, such as 'mtrans-frame'. In the case of 'mtrans-frame', the function *pretty-frame* (FRAMEKIT's function for pretty-printing a frame object) is stored. The newly created frame inherits this function from 'mtrans-frame' and the object of pretty-printing is instantiated to be *HAVE-A-PAIN which is a subclass of *CONCEPT and is the object of printing in our example input.

With the DMA model, natural language understanding is performed as a memory search in the network of concepts, by first identifying the input with the specific concept sequence that represents a root concept, and then performing concept refinement. Since the actual interface to the MT system can be stored in the root node, we will only need to evaluate[49] the result of the parse, and thus as soon as the parse finishes, the command action is directly performed. Likewise, the natural language interface for triggering system functions is integrated into the memory search activity under the DMA paradigm, and this way, inference is integrated into natural language understanding.

12. CHARACTERISTICS OF DMA NL-INTERFACE

12.1 Context Again

In order for a natural language interface to the internals of the machine translation system to work, the interface must be able to recognize the input based on what it already knows as domain specific knowledge in the area of translation and the system's own implementation. When some action is requested the interface must understand the request and respond according to what is requested, and therefore it is necessary to recognize the input within the context of the domain knowledge and current discourse, and to trigger the system's internal functions appropriately. For example, if a knowledge-base developer inputs *Show me all the mapping rules on* *FLIP-DOWN-LEVER in order to debug some conceptual bug in the knowledge-base, the natural language interface needs to recognize what *mapping rule* means in the context of knowledge-base machine translation development as well as recognizing that *FLIP-DOWN-LEVER is a FRAMEKIT definition, in order to show the syntax and semantics mapping rules that are associated with the concept in that domain (such as computer operation). Also when the next input is *And* *SWITCH-ON, if the result of a parse is lost at each sentence, understanding of this sentence is impossible. Other example is when input is *Send the output to Mr. Takeda* where contextual word-sense disambiguation must be performed to recognize 1) that *send* means to send via Unix mail utility; and 2) *output* means the output of the parser, function, etc. according to the current context. These require the

---

[49] Evaluation is implemented in FRAMEKIT system as the triggering of daemons, which is comparable to message passing in object-oriented systems.

natural language module to access the knowledge source of the MT system during parsing and also to recognize the input in the context of the domain knowledge; knowledge about system's internal implementations, and current discourse. DM-Command handles these because parsing is performed as recognition of current input with what it already knows as domain knowledge and as knowledge about the system (in which it is used). Also, the result of each parse is not lost but accumulated in the active memory network.

12.2. Integration of Inference in NL-Interface

The parser for a natural language interface to an MT system needs to recognize the input according to what the MT system already knows as the knowledge source and according to its own internal implementations. A traditional integrated parser[50] will require an external inferential process that will perform the tasks of contextual disambiguation and inferencing in searching for the appropriate action determined by the system's particular internal architecture. Ideally, the inference module and the parser must interact during parsing, due to the constraints put on the understanding of the system within the context established by the knowledge domain and the system's implementation. However, unless memory and inference are integrated, such an interaction is difficult to perform[51], and without such interactions, parsing can be either very slow or fail in contextually difficult sentences because of the interdependencies of concept meanings expressed in the input language.

In the DM-Command system, memory is organized so that the concept which represents the request for action is directly connected to the concept that represents the action that is requested. Likewise, the direct memory access recognition of a question means that the concept which is identified by the input is directly connected to the concept that represents the answer, as long as the system knows (or potentially knows) the answer. In other words, in the DMA model, recognition of a request for action is a triggering of the action requested and recognition of a question is knowing the answer (i.e., as soon as we understand the question, either we know the answer, or we know the inferences to be performed (or functions to be evaluated) to get the answer) as long as memory contains the action and the answer. To reiterate our claim in the previous sections, in this model, memory is organized in the hierarchical network of concepts which are related by links that define the concepts. Thus, as soon as we identify the input with a certain concept in the memory, we can trigger the action (if this is a concept that represents some action (or request for action)), or answer the question (if the concept represents some knowledge (or request for some knowledge)). Thus, parsing and inference are integrated in the memory search process. It should be understood, however,

[50] By integrated parser, we mean a parser that performs both syntactic and semantic analyses in some integrated manner.

[51] For example, DESI (see [8]) used a request-based conceptual analyzer (see [34]) for parsing input to the natural language interface which supplied meaning representations to the separate inference module. The separation of the two modules was inevitable in such a system, because conceptual analyzers were without long-term memory.

that it is not our claim that we can eliminate inference altogether. Our claim is that 1) the memory search through concept refinement itself is an inference which is normally performed by separate inference modules (such as contextual inference and discourse analyses modules) in other parsing paradigm; and 2) whenever further inference is necessary, such inference can be directly triggered after concept refinement from the result of the parse (for example, as a daemon stored in the abstraction of the refined concept) and therefore, the inference is integrated in the memory activity.

12.3 Multiple Semantic Network and Portability

DM-COMMAND utilizes two types of semantic networks. One is the semantic network that is developed under the MT system as domain knowledge that DM-COMMAND utilizes. The other is the network of memory which is unique to DM-COMMAND. This memory represents a hierarchy of concepts involved in commanding and question-answering necessary for the development of machine translation systems. This memory network is written with generic concepts for development of MT systems, so that this memory we have developed at CMT should be portable to other systems[52].

The control mechanism (i.e., spreading activation guided marker-passing algorithm) and the actual functions for performing actions are separate (actual functions are integrated into the DM-COMMAND memory network). This separation makes the system highly portable, first because virtually no change is necessary in the control mechanism for transporting to other systems, and second because the size of the whole system can be trimmed or expanded according to the machine's available virtual memory space simply by changing the size of the DM-COMMAND memory network[53].

Thus, under DMA, a natural language interface can 1) directly spread markers on the target system's already existing semantic network[54], utilizing the existing knowledge for understanding input texts; 2) utilize a command and query conceptual network developed elsewhere (such as DM-COMMAND), with minimum modifications in the functions stored in the root nodes that trigger the actions; 3) be ported to different systems with virtually no change in the control mechanism since it is a guided spreading activation marker-passing mechanism and no system specific functions are included (those functions are included in the command/query semantic net).

---

[52] Of course, we will need to change the specific functions that are stored in some of the nodes and perhaps some of the specific (lower in the hierarchy) concepts need to be modified for each specific system.

[53] If only a basic command natural language interface is required, then we can trim the parts of memory used for advanced interface and question-answering. On the other hand, if machine's memory is of no concern, we can write memory-net and concept-sequences for all the system functions of the target MT system. Also, note that due to the spreading activation guided marker-passing algorithm of the DM-COMMAND recognizer, the speed of the system is minimally affected by an increase in the size of the memory for commanding and question-answering. It is because spreading activation is local to each concept and its packaged nodes under guided marker-passing that even if the size of the whole memory network increased, the amount of computation for each concept should not increase accordingly.

[54] As long as semantic nets are implemented in a general frame language or object oriented systems.

PART V

DISCUSSION

13. HANDLING ELLIPSIS AND ANAPHORA IN DMA

In a practical natural language systems, the capacity to handle elliptic and anaphoric expressions is important. As we have seen in our applications of DMA, DMA systems introduced in this paper are capable of handling these phenomena, because under the DMA paradigm the result of each parse is not lost after each sentence, but instead remains as part of the contextual knowledge in the memory network. Specifically, 1) ellipses are handled since ellipses are characterized as the lack of elements in a concept sequence, and these are recoverable as long as the elements or their descendants had been activated in previous parses. For example, in DM-COMMAND, with the input *jgr92.gra uchidase. sem.tst mo.* (Print jgr92.gra. Sem.tst also). Second sentence has the object dropped; however, this can be supplied since the memory activity after the first sentence is not lost and the memory can supply the missing object; 2) anaphoric and pronoun references are resolved by utilization of both semantic knowledge (represented as constraints on possible types of resolutions) and also by the activations left from the previous parses in memory similar to the way that the elliptic expressions are handled. Finding a contextually salient NP corresponding to some NP means, in DMA, searching for a concept in memory which is previously activated and can be contextually substituted for currently active concept sequence. The process of resolving anaphoric expressions is the process of finding the entity (CI node) that previous NP was identified with and the current anaphoric phrase (word) is referring to. In other words, it is the process that uses the already existing specifications (i.e., CI node activation/creation through lexical input and concept refinement) of a phrase to find the specification of the anaphoric expression[55]. For example in *Pretty-print dm.lisp. Send it to mt@nl, it* can be identified with the concept in memory that represents dm.lisp recorded as specific CI in memory during the recognition of the first sentence.

14. DMA AND SYNTAX

One noticeable characteristic of the current implementations of the DMA paradigm is that the concept sequence is the sole syntactic knowledge for constituent order in parsing[56]. Therefore, a DMA system needs deliberate preparation of concept sequences

[55] Sidner's view of anaphora interpretation (see [40]) is exactly the way that it is expressed here, although Sidner did not take memory-based approach to the solution.

[56] Although generation is normally helped by external syntactic knowledge such as in the case of DMTRANS PLUS. Also, note that here syntactic knowledge simply refers to constituent order rules and principles only.

to handle syntactically complex sentences (such as deeply embedded clauses, small clauses, many types of sentential adjuncts, etc.). This does not mean that it is incapable of handling syntactically complex sentences[57]; instead it means that concept sequences at some level of abstraction (at syntactic template level down to phrasal lexicon level) must be prepared for each type of complex sentence. In other words, although such sentences can be handled by the combination of concept sequences, designing such sequences can be complex and less general than using external syntactic knowledge. Thus, current adoption of linear sequence of concepts demonstrates interesting trade-off between strong contextual recognition capacity (and handling ill-formed input which is very valuable for speech inputs and NL-Interface) and necessity for deliberate (and often complex) CSC preparation for handling syntactically complex sentences.

Of course, there is nothing to prevent DMA paradigm to integrate constituent order syntactic knowledge other than a linear sequence of concepts. Actually, we have already implemented two alternative schemes for integrating phrase-structure rules into DMA. One method we used was having syntactic nodes as part of the memory and writing phrase-structure rules as concept sequences[58]. Another method was to integrate the DMA memory activity into an augmented context-free grammar unification in a generalized LR parsing. The second method used in a continuous speech understanding is described in [44] in which DMA based memory activity was integrated with unification-based syntax/semantics processing to attain strong syntactic screening on candidate hypothesis for a stream of phonemes[59].

While handling syntactically complex sentences is rather expensive for DMA systems, since it relies solely on linear concept sequences, the capacity to handle phenomena such as ellipsis, anaphora, pronoun resolution, and contextual disambiguation is often more valuable than handling syntactically complex sentences. Our experience shows that an increase in the size and complexity of the system in order to integrate full syntactic processing, enhancing the DMA's capacity to handle syntactically complex sentences, has so far outweighed the need for such capacity. Also, as a run-time system, since massively parallel network is assumed, concurrent activation of large number of sequences poses no problem[60].

---

[57] Actually, the descriptive power of the concept sequences is (at least) equivalent to context free grammar. We could even write context free grammar as a concept sequences in extreme cases.

[58] Due to recursive nature of phrase-structure rules, we did not find this method appealing, unless we obtain a truly parallel machine.

[59] Another attempt that is currently being made at CMT is to provide a better environment for encoding concept sequences and eliminating the need for external syntactic help altogether. Currently a graphic oriented CSC editing system is being developed at CMT that takes advantage of an existing graphic frame-based network editor at CMT (see [27]).

[60] Rather, we find it is the advantage of our model from both viewpoints of efficiency and cognitive plausibility to have large number of concurrently active sequences in memory while parsing.

PART VI

CONCLUSION

We have seen three types of applications of Direct Memory Access paradigm of natural language understanding. From a practical point of view, DMA may be interesting because a lexically guided spreading activation mechanism is parallel in nature, and recent availability of massively parallel machines[61] makes it an appealing theory for machine translation, utilizing such parallel architectures. However, the impact of this theory is that natural language processing is performed as an integrated part of cognition, cooperating with other faculties through memory. Most NL systems have failed in tackling contextually ambiguous sentences; however, in DMA models, with use of episodic and thematic memory, cost constraint propagation, and the C-Maker propagations, performance with ambiguous, elliptic, and anaphoric sentences is significantly improved.

Also in terms of translation, explanatory generation handles culturally sensitive translations more effectively, especially when lexical entries in the target language are not available. Also, we have experienced some favorable results with integration of DMA type recognition with unification-based parsing and such a scheme is available if strict syntactic processing is necessary for a specific natural language application area.

Spreading activation marker passing algorithm is an established method of knowledge-base search in Artificial Intelligence research and the Direct Memory Access paradigm brings the approach to the level of simulating human cognitive processing. As a model of natural language processing, we have seen that DMA is not only a strong theory of human information processing as claimed in the past literature in the field, but also a viable scheme for building practical natural language systems.

Acknowledgments. The authors would like to thank members of the Center for Machine Translation for fruitful discussions. Lori Levin, Eric Nyberg and Teruko Mitamura were especially helpful in preparing the final version of this paper.

APPENDIX

APPENDIX 1: IMPLEMENTATIONS

A. DmTrans Implementation

The DmTrans was originally developed as a natural language front-end to a case-base reasoning system developed at Yale AI Project under Project IVY with Lawrence Hunter, Alain Rostain from Yale University and Dr. Jerry Silbert of West Haven Veterans Administration Hospital. The parser part of the code was originally written in T programming language as a part of DMAP Project at Yale AI Project and was converted

[61] Such as 'The Connection Machine' (see [16]).

to CMU-COMMONLISP at Center for Machine Translation (CMT) at Carnegie Mellon University. Other parts of the DMTRANS system were developed at CMT. FrameKit+ is used as the knowledge representation tool. The system runs on an IBM-RT[62] running Mach (see [33]) and parallelism is simulated using lazy evaluations. Latest version of the system DMTRANS PLUS is implemented on CMU-COMMONLISP using FRAMEKIT VERS. 2.0 (see [29]). Virtually all code is rewritten for DMTRANS PLUS.

B. ΦDMTRANS Implementation and Recognition Hardware

Speech recognition hardware was developed by Matsushita Research Institute and is used in our system by the courtesy of the Institute. In addition to the firm-ware written control codes, low-level control program is written in 'C' for the device hardware. Current implementation of ΦDMTRANS runs real-time[63] on HP9000 AI Workstations and is written in the HP CommonLisp. Object-code of speech recognition control programs is directly called from inside the CommonLisp code. Also, non-real-time[64] versions are implemented on IBM-RTs using CMU-COMMONLISP and MULTILISP. The parallelism of spreading activation is simulated using lazy evaluations in CommonLisp versions. Parallelism in the MULTILISP[65] version is supported at the operating system level on Mach.

C. DM-COMMAND Implementation

The DM-COMMAND system has been implemented on the IBM-RT and HP9000 AI workstations, both running CommonLisp. The system directly utilizes the FRAMEKIT-represented domain knowledge (currently in the area of computer manuals and doctor/patient conversations) of the CMU-MT knowledge-based large-scale machine translation system. It handles inputs in both English and Japanese. The current size of the DM-COMMAND system is roughly 5,000 lines of Lisp code (this does not include the MT system functions and the FRAMEKIT VERS. 2.0 frame system, parts of which must also be

---

[62] Due to space limitations the actual sample outputs of the systems described in this paper are not included. The technical reports from CMU-CMT contain sample outputs of all three systems.

[63] By 'real-time' we mean that what is spoken into the microphone is translated into sentences in the target language with a negligible delay.

[64] Non-real-time on IBM-RTs simply because hardware connections between RTs and the speech recognition hardware are not currently supported and therefore, processing is done via a network.

[65] MULTILISP is described in Halstead [1985], which is a parallel lisp developed at MIT for Concert multi-processors and is now implemented on a distributed operating system Mach at CMU. Because MULTILISP is a true parallel Lisp, the MULTILISP version of ΦDMTRANS runs on any parallel hardware that supports MULTILISP. MULTILISP has already been implemented on several types of parallel computers including Concert, Multi-vaxens and Encores.

loaded into memory) and is not expected to increase, since the future variety in types of commands and questions that the system will handle will be integrated into the network of memory that represents concepts for commanding and question/answering and not into the system code itself[66]. Compiled code on IBM-RTs and HP9000s is fast enough that parsing and performing commanded action happens virtually in real-time. We are expecting to increase the variety in types of system functions and grammar/rule development functions; however, as noted above, since such increases will occur in the memory network, as a system implementation, DM-COMMAND is a completed system.

APPENDIX 2: DISTINCTIVE FEATURE MATRIX USING SPE[67]

```
atamagaitai   (I have a headache.)
kazokuwaitai  ((The) families want to stay.)
kazokuheitai  ((My) family is soldier(s).)
kazokudeitai  (I want to stay as (a) family.)
asabanaisou   (Love (make love) (every) morning and night.)
asakaraikou   (Go (come) (from) tomorrow morning.)
kazokuwaikou  ((The) families go.)
asamadeikou   (Go before morning, Come until morning.)
okosanaika    (Shall we wake (one) up?)
okosumaika    (Shall we not wake (one) up?)
kazokuheikou  ((The) family is disappointed.)
kazokudeikou  (Go with the family.)
gohunaisou    (Love (make love) for five minutes.)
ugokumaika    (Shall I not move?)
atukunaika    (Is it not hot?)
dokoeikou     (Where shall we go?)
dokodeikou    (Where shall we come?)
koupumadeikou (go to (the) cup.)
```

```
  "atama" < lexical node
  <a t a m a>  < phonemic sequence
  /              attached to "atama"
 /
 |   -5--/u/  < phonological rel link with
 |  /           distinctive feature weight
 | /
/a/   < phoneme node
 |
[a]   < phone node
```

[66] One advantage of DM-COMMAND is that the whole system is only 5,000 lines long and we need not load the whole MT system (which is quite large) for developing grammar and concept entity definitions and writing syntax/semantics mapping rules.

[67] Due to Teruko Mitamura.

Below is the distinctive feature matrix used in the Phoneme-based DMTRANS for Japanese as a basis of encoding weights of links.

	p	t	(c)	k	b	d	g	(*)	s	z	r	m	n	=	w	j	h	i	e	a	o	u
cons	+	+	+	+	+	+	+	+	+	+	+	+	+	+	−	−	−	−	−	−	−	−
syll	−	−	−	−	−	−	−	−	−	−	−	−	−	+	−	−	−	+	+	+	+	+
son	−	−	−	−	−	−	−	+	−	−	+	+	+	+	+	+	−	+	+	+	+	+
high	−	−	+	+	−	−	+	+	−	−	−	−	−	+	+	+	−	+	−	−	−	+
back	−	−	−	+	−	−	+	+	−	−	−	−	−	+	+	−	−	−	−	+	+	+
low	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	+	−	−
cor	−	+	+	−	−	+	−	−	+	+	+	−	+	−	−	−	−	−	−	−	−	−
voice	−	−	−	−	+	+	+	+	−	+	+	+	+	+	+	+	−	+	+	+	+	+
cont	−	−	−	−	−	−	−	−	+	+	−	−	−	−	+	+	+	+	+	+	+	+
nasal	−	−	−	−	−	−	−	+	−	−	−	+	+	+	−	−	−	−	−	−	−	−

REFERENCES

[1] Becker, J. D. (1975): The phrasal lexicon. In: Theoretical issues in natural language processing. Proceedings of the workshop of the ACL. (Eds.) Schank, R. C. and Nash-Webber, B. N.

[2] Berg, G. (1987): A Parallel Natural Language Processing Architecture with Distributed Control. In: Proceedings of the CogSci-87.

[3] Bookman, L. A. (1987): A Microfeature Based Scheme for Modelling Semantics. In: Proceedings of the IJCAI-87.

[4] Charniak, E. (1983): Passing Markers: A theory of Contextual Influence in Language Comprehension. Cognitive Science 7.

[5] Charniak, E. and Santos, E. (1987): A Connectionist Context-Free Parser Which is Not Context-Free, But Then It is Not Really Connectionist Either. in: Proceedings of the CogSci-87.

[6] Charniak, E. (1986): A neat theory of marker passing. In: Proceedings of the AAAI-86.

[7] Chomsky, N., and Halle, M. (1986): The Sound Pattern of English. New York, Harper and Row.

[8] Cullingford, R. E. and Booth, S. L. (1985): How to make a natural-language interface robust. GIT-ICS-85/27, Georgia Institute of Technology.

[9] DeJong, F. (1979): Skimming stories in real time: An experiment in integrated understanding Report 116, Dept. of Computer Science, Yale University.

[10] Fahlman, S. E. (1983): NETL: A system for representing and using real-world knowledge. The MIT Press.

[11] Granger, R. H., Eiselt, K. P. (1984): The parallel organization of lexical, syntactic, and pragmatic inference processes. In: Proceedings of the First Annual Workshop on Theoretical Issues in Conceptual Information Processing.

[12] Halstead, R. (1985): Multilisp: A language for Concurrent Symbolic Computation. In: ACM Trans. on Prog. Languages and Systems.

[13] Hayes, P., Hauptmann, A., Carbonell, J. and Tomita, M. (1986); Parsing Spoken Language: A Semantic Caseframe Approach. In: Proceedings of Coling-86.

[14] HIRAOKA, S., MORII, S., HOSHIMI, M. and NIYADA, K. (1986): Compact Isolated Word Recognition System for Large Vocabulary. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP86).

[15] HAHN, U. and REIMER, U. (1983): World expert parsing: An approach to text parsing with a distributed lexical grammar. Technical Report, Universität Konstanz, West Germany.

[16] HILLIS, Daniel W. (1985): The Connection Machine. The MIT Press.

[17] HIRST, G and CHARNIAK, E. (1982): Word Sense and Slot Disambiguation. In: Proceedings of AAAI-82.

[18] HIRST, G. (1984): A Semantic Process for Syntactic Disambiguation. In: Proceedings of AAAI-84.

[19] HOBBS, J. (1988): Interpretation as Abduction. In: Proceedings of ACL-88.

[20] HYMAN, L. (1975): Phonology: Theory and Analysis. Holt, Reinhart and Winston.

[21] KITANO, H. (1988): Multilingual Information Retrieval Mechanism using VLSI. In: Proceedings of the RIAO-88.

[22] KITANO, H., TOMABECHI, H., and LEVIN, L. (1988): Ambiguity Resolution in the DMTRANS PLUS. Submitted to Fourth Conference of the European Chapter of the Association for Computational Linguistics.

[23] LEE, L., TSENG, C., CHEN, K. J., and HUANG, J. (1987): The Preliminary Results of A Mandarin Dictation Machine Based Upon Chinese Natural Language Analysis. In: Proceedings of the IJCAI-87.

[24] LYTINEN, S. (1984): The organization of knowledge in a multi-lingual, integrated parser. Ph.D. thesis Yale Univesity.

[25] MITAMURA, T., MUSHA, H., KEE, M. (1988): The Generalized LR Parser/Compiler Version 8.1: User's Guide, Tomita, M., (Ed.) CMU-CMT-88-MEMO. Carnegie Mellon University.

[26] MORII, S., NIYADA, K., FUJII, S. and HOSHIMI, M. (1986): Large Vocabulary Speaker-independent Japanese Speech Recognition System. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP85).

[27] NIRENBURG, S., MONARCH, I., KAUFMANN, T., NIRENBURG, I., and CARBONELL, J. (1988): Acquisition of Very Lage Knowledge Bases: Methodology, Tools and Applications. CMU-CMT-88-108.

[28] NORVIG, P. (1987): Inference in Text Understanding. In: Proceedings of the AAAI-87.

[29] NYBERG, E. (1988): The FrameKit User's Guide Version 2.0. CMU-CMT-88-107. Carnegie Mellon University.

[30] PRITCHETT, B. (1988): Garden Path Phenomena and the Grammatical Basis of Language Processing. LANGUAGE, Journal of LSA, Vol. 64, Number 3.

[31] QUILLIAN, M. R. (1968): Semantic Memory. In: Semantic Information Processing, Minsky, (Ed.) M. MIT Press.

[32] QUILLIAN, M. R. (1969): The teachable language comprehender. BBN Scientific Report 10.

[33] RASHID, R., A. TEVANIAN, M. YOUNGE, D. YOUGE, R. BARON, D. BLACK, W. BOLOSKY and J. CHEW (1987): Machine-Independent Virtual Memory Management for Paged Uniprocessor and Multiprocessor Architectures. CMU-CS-87-140. Carnegie Mellon University.

[34] RIESBECK, C. (1975): Conceptual Analysis. In: Conceptual Information Processing, Schank, R. C. (Ed.) North Holland.

[35] RIESBECK, C. (1986): From Conceptual Analyzer to Direct Memory Access Parsing: An Overview. In: Advances in Cognitive Science 1, Sharkey, N. E. (Ed.) Ellis Horwood.

[36] RIESBECK, C. and MARTIN, C. (1985): Direct Memory Access Parsing. Yale University Report 354.

[37] SAITO, H. and TOMITA, M. (1988): Parsing Noisy Sentences, In: Proceedings of the COLING-88.

[38] SCHANK, R. (1982): Dynamic Memory: A theory of learning in computers and people. Cambridge University Press.

[39] SCHANK, R. (1986): Explanation Patterns: Understanding mechanically and creatively. Lawrence Erlbaum Associates, Publishers.

[40] SIDNER, C. (1981): Focusing for Interpretation of Pronouns. American Journal of Computational Linguistics, Vol. 7. Number 4.

[41] SMALL, S. and REIGER, C. (1982): Parsing and comprehending with word experts (a theory and its realization). In: Strategies for natural language processing. Lenhert G. and Ringle M. Lawrence Erlbaum (Ed.).

[42] TOMABECHI, H. (1987a): Direct Memory Access Translation. In: Proceedings of the IJCAI-87.

[43] TOMABECHI, H. (1987b): Direct Memory Access Translation: A Theory of Translation. CMU-CMT-87-105, Carnegie Mellon University.

[44] TOMABECHI, H. and TOMITA, (1988a): 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.

[45] TOMABECHI, H. and TOMITA, M. (1988b): Application of the Direct Memory Access paradigm to natural language interfaces to knowledge-based systems. In: Proceedings of the COLING-88.

[46] TOMABECHI, H. MITAMURA, T. and TOMITA, M. (1988): 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).

[47] TOMITA, M. (1985): Efficient Parsing for Natural Language: A Fast Algorithm for Practical Systems. Kluwer Academic Publishers, Boston, MA.

[48] TOMITA, M. (1986): An Efficient Word Lattice Parsing Algorithm for Continuous Speech Recognition. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP86).

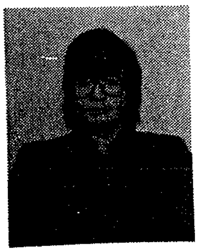
[49] TOMITA, M. and CARBONELL, J. (1987): The Universal Parser Architecture for Knowledge-Based Machine Translation. In: Proceedings of the IJCAI-87.

[50] TOMITA, M., KEE, M., MITAMURA, T. and CARBONELL, J. (1987): Linguistic and Domain Knowledge Sources for the Universal Parser Architecture. In: Terminology and Knowledge Engineering H. Czap, and C. Galinski (Eds.) INDEKS Verlag.

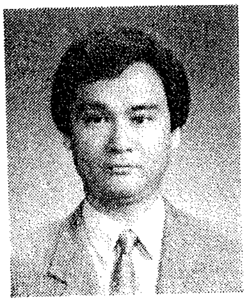
[51] TOURETZKY, D. (1988): Beyond Associative Memory: Connectionists Must Search for Other Cognitive Primitives. In: Proceedings of the 1988 AAAI Spring Symposium Series. Parallel Models of Intelligence.

[52] WAIBEL, A., SAWAI, H., and SHIKANO, K. (1988): Modularity and Scaling in Large Phonemic Neural Networks. ATR TR-I-0034. ATR Interpreting Telephony Research Laboratories.

[53] WALTZ, D. and POLLACK, J. (1984): Phenomenologically plausible parsing. In: Proceedings of the AAAI-84.



Hideto TOMABECHI is a graduate student. He is currently working on a joint Ph. D. Program in Computational Linguistics at Carnegie Mellon University. He is also a member of the Center for Machine Translation, Carnegie Mellon University.



Masaru Tomita is an Assistant Professor in the Computer Science Department, and also the Associate Director of the Center for Machine Translation at Carnegie Mellon University. He holds a Ph.D and a Master's degree in Computer Science from Carnegie Mellon University (1985 and 1983, respectively) and a Bachelor's degree in Mathematics from Keio University (Yokohama, Japan, 1981). During 1984 he was a Visiting Scientist in the Electrical Engineering Department at Kyoto University (Kyoto, Japan). Dr. Tomita's research interests are in the area of natural language processing, including machine translation, parsing, natural language interfaces, computational linguistics and speech recognition. He has published one book, and authored or co-authored over 40 refereed papers. In 1988 he received a Presidential Young Investigators Award from the National Science Foundation. He is a member of the Association for Computational Linguistics, the American Association for Artificial Intelligence and the Information Processing Society of Japan.

Editor-in-Chief: Ivan PLANDER

Executive Editors: Ferdinand GLIVIAK and Rudolf FIBY

Editorial Office: Institute of Technical Cybernetics, Slovak Academy of Sciences, Dúbravská cesta 9, 842 37 Bratislava

Published under the auspices of the Institute of Technical Cybernetics, Slovak Academy of Sciences, by VEDA Publishing House of the Slovak Academy of Sciences, Klemensova 19, 814 30 Bratislava. Printed by Západoslovenské tlačiarne, n. p., závod Svornosť, 832 10 Bratislava. Published bimonthly. Approved by FUTI, No. F 7125.

Subscription Information: Orders in Czechoslovakia are accepted by PNS — UED, Gottwaldovo nám. 6, 813 81 Bratislava. Orders from abroad should be sent to SLOVART Ltd., Gottwaldovo nám. 6, 817 59 Bratislava, Czechoslovakia. Annual subscription rate approx. Rbl 12.— for socialist countries, US $ 120.— for all other countries.