第五世代コンピュータ国際会議 1988

PROCEEDINGS OF THE

INTERNATIONAL CONFERENCE ON FIFTH GENERATION COMPUTER SYSTEMS 1988

FGCS'88

Nov. 28–Dec. 2, 1988
Tokyo Prince Hotel
Tokyo, JAPAN

Institute for
New Generation Computer Technology

VOL.3

PROCEEDINGS OF THE INTERNATIONAL CONFERENCE
ON FIFTH GENERATION COMPUTER SYSTEMS 1988,
edited by ICOT. © ICOT, 1988

DIRECT MEMORY ACCESS TRANSLATION FOR SPEECH INPUT
A Massively Parallel Network of Episodic/Thematic and Phonological Memory

Hideto Tomabechi, Teruko Mitamura, and Masaru Tomita
Center for Machine Translation
Carnegie Mellon University
Pittsburgh, PA 15213 U.S.A.

ABSTRACT

This paper describes the Phoneme-Based Direct Memory Access Translation System (DMTRANS) which is a speech to speech translation system developed at the Center for Machine Translation at CMU. DMTRANS utilizes a phonological and episodic/thematic memory network, and performs spreading activation guided marker passing which is massively parallel in nature. DMTRANS handles the problem of multiple hypothesizations of input phonetic streams through network memory-based encoding of knowledge for language-specific phonology and morphophonetics, as well as episodic/thematic memory that supplies contexual disambiguations of the input. The architecture is ideal for massively parallel computer systems that are currently researched by hardware developers.

1 Introduction

Recently a few efforts have been made in the area of processing speech input to a natural language understanding system. These include the work of Hayes, *et al*[1986], Tomita[1986], Poesio&Rullent[1987], Saito&Tomita,[1988], Tomabechi&Tomita[1988a], and Hauptmann, *et al*[1988]. Among them, Tomabechi&Tomita and Hauptmann, *et al* use contexual 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.

ΦDMTRANS uses parallel processing, and our experiments with the prototype ΦDMTRANS at Center for Machine Translation at the Carnegie Mellon University show that ΦDMTRANS is a promising framework for translating speech input cross-linguistically in new generation parallel computers.

2 Some Background and History:

2.1 Recognize-and-Record

ΦDMTRANS is a "Phoneme-Based Direct Memory Access Translation" architecture which represents what we call the "recognize-and-record" paradigm of natural language processing usually grouped as DMA (Direct Memory Access) models. In this model, natural language understanding is viewed as a memory activity which identifies input with what is already known 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 past linguistic and non-linguistic experiences are being remembered during the course of understanding the input, and each sentence 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 lost after the processing of each sentence.

2.2 A Brief History

The Direct Memory Access method of parsing originated in Quillian's[1968] notion of semantic memory, used in his TLC (Quillian[1969]), which 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 a semantic memory, leaving interpretation of text as an intersection of the paths. Thus, interpretation of input text was directly performed on semantic memory. DMA was not explored as a paradigm for parsing (except as a scheme for disambiguation) until mid 1980's when DMAP0 (Riesbeck&Martin[1985]) followed by

[1]This includes the work of Fahlman[1979], Hirst&Charniak[1982], Small&Reiger[1982], Charniak[1983], Haun&Reimer[1983], Hirst[1984], Charniak[1986], Charniak&Santos[1987], Norvig[1987], and recent connectionist and distributed models such as Granger&Eiselt[1984], Waltz&Pollack[1984], Berg[1987], Bookman[1987].

Tomabechi[1987a,b] developed the DMA paradigm into theories of parsing and translation respectively. These projects were part of the Yale AI Project and were aimed at building a DMA natural language system to be integrated with case-based reasoning systems developed under the XP (eXplanation Patterns) theory of Schank[1986]. Since DMA parsers work directly on memory through spreading activation, integration of natural language understanding with the experiential memory of the case-based system became possible. These DMA systems 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. P-markers (Prediction markers) and A-markers (Activation markers) are markers passed around in memory, adopting the notion of concept sequence which guides marker passing along the known ordering of concepts. Recently, the paradigm was adopted as a scheme for a natural language interface for development of knowledge-based systems (Tomabechi&Tomita[1988b]).

3 Problems in Speech Input

3.1 Phonetics, Phonology and Morphology

The difficulty of parsing speech input is that unlike written text input, a parser receives multiple hypotheses as input for 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 input.

For example, when testing the CMU-CMT speech parser (a phoneme-based Generalized-LR parser (ΦGLR, Saito&Tomita[1988])), the Japanese input "atamagaitai" ("I have a headache") was spoken into a speech recognition system[3] (under ordinary office environment) and accepted by the integrated[4] parser with 57 ambiguous interpretations. Each of the ambiguous interpretations are semantically legitimate, ...meeting the local restrictions set forth by case-frame instantiation restrictions. Below are some of the highly scored interpretations:

```
atamagaitai   (I have a headache.)
kazokuwaikou  ((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.)
```

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 variations 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.

3.2 Need for Contextual Knowledge

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 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

[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.

[3] Matsushita Research Institute's speech recognition hardware. The speech recognition system and the speech input enhanced LR parser are described in detail in Saito&Tomita[1988].

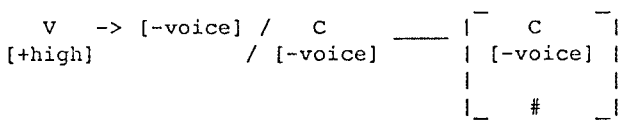
[4] By 'integrated', we mean concurrent processing of syntax and semantics during parsing as opposed to some parsing methods where syntax and semantics are separately processed.

understanding system, the input can be interpreted in a way that is not possible in text input systems, 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).

4 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 segments of words. Speakers have mental representation of sound systems, which are different from actual physical properties. Speakers of English feel /p/ in 'pin' and 'spin' are identical (and spelled the same in text inputs), but physically they are different sounds. /p/ in 'pin' is aspirated, represented in [ph], whereas /p/ in 'spin' is not aspirated represented in [p]. Both aspirated and unaspirated sounds do not differentiate the meaning in English, and they are predictable from a given environment. These units of phonetic segments are called phones. Thus, there are two levels of sound representation: a phonological level and a phonetic level.

Phonological rules convert phonological representations into phonetic ones. They can change, delete, or add segments. They can also coalesce or permutate segments. In Japanese, high vowels become voiceless between voiceless consonants or after a voiceless consonant in the word final position.



Phonological rules[5] apply to classes of phonetically related segments. In order to capture the common features that certain phonological segments have, phonologists use distinctive features to represent them (Jakobson& Halle[1956]; Chomsky&Halle[1968]). For example, Japanese vowels are represented using the SPE system (Chomsky&Halle) in the following matrix.

	i	e	a	o	u
high	+	−	−	−	+
back	−	−	+	+	+
low	−	−	+	−	−

The five vowels can be distinguished by using three kinds of features, and the matrix shows the phonemic relations. We can see the phonemic distance by counting the differences which are representable as weights:

	i	e	a	o	u
i	0	1	3	2	1
e		0	2	1	2
a			0	1	2
o				0	1
u					0

We can assume that lower distance numbers have higher confusion probabilities (i.e., higher weights). Therefore, when the input phone is [a], we can test from the segment which has a lower distance number, such as [a], then [o], and so on. With this matrix we can limit the test to close segments instead of testing all the segments[6] and group close sounds in the network with certain thresholds. For consonants, distinctive feature matrix is more complex than our example of vowels and is provided in the Appendix 2 which is used as a base for encoding weights of the links.

The utilization of distinctive feature matrices described above; however, is a static knowledge that are 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. The example of voiceless high vowel (specifically [u] and [i] in Japanese) between two voiceless consonants (or word final after voiceless consonant) is one case that we have seen in the phonological rule above. 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[7] so that the phonemic activations are 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.

[5]Phonological rules that dynamically affect processing due to phonological environments are captured via memory network representation (utilizing daemons in our system in 'FrameKit' (Nyberg[1988]) system) stored locally to each phones which was transformed from declarative description of rules originally supplied as phonological knowledge.

[6]Since we use Matsusita Research Institute's Speech 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.

[7]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 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.

5 Contexual Knowledge in ΦDMTRANS

ΦDMTRANS uses an episodic/thematic memory network, similar to the ones described in Schank[1982] and Schank-[1986], which is capable of dynamic modifications, inference and learning. Context in such a conceptual memory network can be represented as a grouping of concepts that are associated in a certain manner, i.e. an activation of one concept in memory triggers (or can potentially trigger) some other concepts in the memory network. To put it in another way, there is a relationship between concepts in which activation (recognition) of one concept reminds some other concept that it is related in a certain way. As we will see in detail in the following section, ΦDMTRANS uses the lexically-guided spreading activation mechanism for parsing. Context in this scheme is represented as what has been activated so far as 1) accepted concepts representing the previous sentences and 2) the concepts in the currently active concept sequences. These activations represent the recognition of what being said so far and also represents what is likely to be heard under the current context. Readers may find our scheme of spreading activations similar to those researched by connectionists. However, we have not adopted connectionist associative architecture[8] and back-propagation in our thematic conceptual clusters. Our spreading activations are guided and we do not spread everywhere.

6 Understanding in ΦDMTRANS

6.1 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 nodes 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:

```
   "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
```

We have two types of markers (structured objects) passed around in memory. One is called P-Marker (for Prediction-Marker) and the other is called A-Marker (for Activation-Marker). P-Markers are passed along the phonemic sequences and A-Markers are passed above in the abstraction hierarchy (i.e., from phone to phoneme). Both markers contain information about which node originated the marker passing. P-Markers also contain information about which was the immediately preceding node in the sequence. 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").

This is the basic cycle that is used in ΦDMTRANS. In the next subsection we discuss how the same algorithm is used for further processing at the sentential level, activating the episodic/thematic memory network. One thing we omitted in the above algorithm (for the sake of simplicity) is the way the phonological relation link is utilized in the activation of phones. Let us examine how this works:

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

[8] 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].

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 7), and if the evaluation applies the rule and perform the dynamic alteration of the currently active phonetic node, then the phonemic nodes of the altered phone is 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. Since our memory network is a massively parallel network, the spreading activations for each concurrently active phonemic sequences will be parallelly performed.

6.2 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 not include the details of this processing in this paper because it is described in detail elsewhere (Tomabechi[1987b] and Tomabechi&Tomita[1988b]). A brief example would be the processing of the sentence "atamagaitai", which we saw before as a problematic input to other speech understanding systems. We use basically the same algorithm as we saw in the processing at phonetic level, except that each unit in the sequence is not a phoneme but a lexical node or a concept node and we call the sequece of such nodes concept sequences. An example of a concept sequence is <*BODY-LOCATION *PP[GA] *PAIN-SPEC> representing the sequence of concepts appear in "atamagaitai". The concept sequence can be regarded as a kind of subcategorization list (as in HPSG, Pollard&Sag[1987]) or as a generalized version of a phrasal lexicon (Becker[1975]) except that the sequence can be at higher levels in abstraction hierarchy as well as being episodic and thematic such as in MOPs and EXPLANATION PATTERNS (Schank[1982&1986]) encoding the knowledge for contexual processing.

We have nodes such as *HAVE-A-PAIN (representing the concept having a pain) and concept sequence such as <*BODY-LOCATION *PP[GA] *PAIN-SPEC> attached to the node (we call it root node if a concept sequence is attached to it). The elements of the sequence are the nodes in the memory network representing certain concepts[9].

Below is our algorithm for word and sentential level activity:

1. initially predict (put P-Marker on) all the first elements of concept sequences in memory.
2. when a whole phonemic sequence is accepted (i.e., a word is recognized), we activate (put an A-Marker on) the lexical node, i.e., activate the node with the accepted phonemic sequence attached to it, and activate the corresponding conceptual node.
3. when a node receives an A-Marker it sends an activation to (A-Marks) the node in its abstraction.
4. when an A-Marker and P-Marker meet, send a P-Marker to the next element of the concept sequence.
5. when the whole concept sequence is activated, then activate the root of the sequence and perform concept refinement.

Concept refinement is an activity to locate the most specific node in memory, below the activated root node, which represents the specific instance of the input text. Such a node must have links to all the specializations (or instances) of the nodes that appeared in the concept sequence with relations that are equivalent to (or subclasses of) the relation links from the root node to the packaged nodes in the accepted concept sequence. The search for such a node underneath the root node is called concept refinement. This activity, which locates the concept that is identified with the specific input speech, is central to the understanding in the DMA parsing.

Processing of the example sentence "atamagaitai" is as follows: when the lexical node "atama" is activated after the acceptance of the phonemic sequence <a t a m a>, then we activate the corresponding conceptual node "*HEAD" and spread the activation upward in the abstraction hierarchy. One of the abstractions is the concept "*BODY-LOCATION". At the beginning of understanding, we have all first elements of the concept sequences P-Marked (just as we did so with first elements of phonemic sequences). So "*BODY-LOCATION" was P-Marked by the root node "*HAVE-A-PAIN". Therefore, when "*BODY-LOCATION" is activated from below, we have a collision of A-Marker and P-Marker. When the collision happens, we send a P-Marker to the next element of the concept sequence (i.e., "*PP[GA]"). This is continued and the last element "*PAIN-SPEC" gets accepted after acceptance of <i t a i>. So we activate the root node "*HAVE-A-PAIN". One thing that happens (that we did not have at phonetic level) is that we perform the 'concept refinement'[10], which is essentially what understanding in DMA means. It involves identifying the specific instance of the accepted root concept that represents the input to the understanding system. In our case, the memory searches for the node "*HAVE-A-HEADACHE" (or creates it if non-existent yet), that is underneath "*HAVE-A-PAIN" and packages the nodes "*HEAD", "*PP[GA]", "*PAIN-SPEC[UNSPEC]" that are specific to the current input. Since concept sequences are generic and attached to relatively higher nodes in abstraction hierarchy, it is this concept refinement that specifies (or identifies) the specific input to the system. After concept refinement, we now have the node "*HAVE-A-HEADACHE" activated, and that is the result of the understanding. Of course, in the actual system, the spreading activation con-

[9] *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 represented using our frame language 'FrameKit'.

[10] Lytinen[1984] and Tomabechi[1987b] have detailed discussions of 'concept refinement'.

tinues in a parallel manner because the concepts "*BODY-LOCATION" and "*HAVE-A-HEADACHE" (and the concepts in between them) may be a part of some other higher level concept sequences in abstractions such as scriptal and episodic memory packets.

6.3 Contexual Activity

We have two types of contexual activity in ΦDMTRANS: 1) C-Marker based activity; and 2) episodic/thematic based activity. C-Marker passing is an algorithm introduced in Tomabech[1987a] in which text input based DMTRANS passed C-Maker (for Contexual-Marker) around in memory every time a contexual (thematic) root node was activated. The contexual (thematic) root nodes are the nodes that increases the potential activities of the nodes that are likely to be heard under the given context and the DMTRANS paper contains an example handling the semantic ambiguity of "paper" for 'physical object paper' and for 'thesis' under different contexts using the C-Marker passing. Tomabechi&Tomita[1988a] has a similar thematic mar passing which integrates memory-based pragmatics into unification-based syntax and semantics.

The episodic/thematic based activity is triggered by the concept sequences that are with normally extra-sentential span. These include scriptal knowledges and explanation patterns that are triggered by acceptance of series of concepts that constitute such sequences. These episodic and thematic predictions are utilized because P-Markers are passed around at these abstract levels just as in the phrasal levels. This way, strong predictions are always active as part of higher level (episodic/thematic) concept sequences as well as increases in potential[11] contexual activities through C-Marker passing.

7 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 des 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 Morii, *et al*[1985] and Hiraoka, *et al*[1986]. 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 stablizes. Generation is performed directly from the state of the memory after the understanding. In essence, generation is performed to output the sentences in the target language that are identified by what is left after understanding. Interested readers may want to refer to the explanatory generation section of Tomabechi[1987b] that describes DMTRANS, which translates a written text input. Through explanatory generation, DMTRANS translates "Gionshoja no kane no koe shogyomujo no hibiki ari" into "Sound of bell at Gionshoja has the tone of "shogyomujo" (impermanence of all phenomena in world). The concept "shogyomujo", which does not have lexical entry in English, was explanatorily translated as "impermanence of all phenomena in world". The generation mechanism outputs the original word in double quotes and generates an explanation of the source lexical entry in parenthesis in the output. ΦDMTRANS utilizes the same explanatory generation mechanism as DMTRANS, and is capable of performing the same type of generation.

8 Future Possibilities

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. Combined with the DMTRANS generator, ΦDMTRANS is a translation system and with the appropriate speech synthesis hardware added (we utilize DECtalk[12] at CMT), the system is a speech to speech translation system with strong contexual 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, Tomabechi&Tomita[1988b]), which ΦDMTRANS can replace its parser 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.

As we have seen, the spreading activation guided marker passing algorithm is massively parallel in nature. It leads to our understanding that ΦDMTRANS is ideal for the new generation computer architectures where massively parallel processings are supported from the hardware level. We currently have a version of ΦDMTRANS on MULTILISP parallel lisp environment; however, we would like to see the system to run on much more massively parallel architecture[13] which can support the parallelism of every phonemic and concept sequence recognitions performed concurrently at all levels of abstractions and triggered by multiple morphophonetic, phonological and semantic hypothesizations of con-

[11] It is potential in the sense that C-Markers do not activate the node directly but will activate the node, when the node gets ambiguous activations in the future, by chosing the node over other candidate nodes that did not receive previous C-Marker passing.

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

[13] Such as neuro-computer type architectures and connection machine (Hillis[1985]) type architectures.

tinuous speech inputs.

9 Conclusion

We have reported an integration of phonological and contexual knowledge in speech understanding in a massively parallel spreading activation marker passing network. As we have seen, the method of marker passing spreading activation is uniform from the phone level up to abstract thematic structures. Because a phonetic input stream can be hypothesized in multiple ambiguous and semantically acceptable ways, we have seen the necessity of both phonological knowledge and contexual knowledge participating during the course of direct memory access translation. Parallel processing of concurrently active phonemic and conceptual sequences seems solely attainable in a DMA style spreading activation architecture. In the traditional build-and-store model, since the result of parsing is lost after the processing of each sentence, the context for subsequent translations is hardly ever established, whereas in our DMA model, context is naturally rec 'ed as what is left in memory after understanding previous sentences as well as what is being recognized as parts of currently active concept sequences. With the explanatory generation mechanism added, the ΦDMTRANS model of translating a speech input is an extremely viable option for future parallel (fifth generation) computers, in which massively parallel processing activity is hardware-supported[14].

ACKNOWLEDGMENTS

The authors would like to thank members of the Center for Machine Translation for fruitful discussions. We would also like to thank Dr. Morii of Matsushita Research Institute for his generous contribution of the speech recognition hardware used by our project. Eric Nyberg and Lori Levin were especially helpful in preparing the final version of this paper.

References

[1] Becker, J.D. (1975) *The phrasal lexicon.* In 'Theoretical Issues in Natural Language Processing'.

[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 Contexual Influence in Language Comprehension.* Cognitive Science 7.

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

[6] 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'.

[7] Chomsky, N., and Halle, M. (1968) *The Sound Pattern of English.* New York: Harper and Row.

[8] Fahlman, S. (1979) *NETL: A system for representing and using real-world knowledge.* The MIT Press.

[9] Granger, R. and Eiselt, K. (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'.

[10] Hahn, U. and Reimer U. (1983) *World expert parsing: An approach to text parsing with a distributed lexical grammar.* Technical Report, Universitat Konstanz, West Germany.

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

[12] Haupmann, A., Young, R. and Ward, W. (1988) *Using Dialog-Level Knowledge Sources to Improve Speech Recognition.* In 'Proceedings of the AAAI-88'.

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

[14] Hillis, Daniel W. (1985) *The Connection Machine.* The MIT Press.

[15] 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).

[16] Hirst, G. and Charniak, E. (1982) *Word Sense and Slot Disambiguation.* In 'Proceedings of AAAI-82'.

[17] Hirst, G. (1984) *A Semantic Process for Syntactic Disambiguation.* In 'Proceedings of AAAI-84'.

[18] Hyman, L. (1975) *Phonology: Theory and Analysis.* Holt, Rienhart and Winston.

[19] Jakobson, R. and Halle, M. (1956) *Fundamentals of Language.* Mouton.

[20] Lytinen S. (1984) *The organization of knowledge in a multi-lingual, integrated parser.* Ph.D. thesis Yale University.

[21] Morii, S., Niyada, K., Fujii, S. and Hoshimi, M. (1985) *Large Vocabulary Speaker-independent Japanese Speech Recognition System.* In 'Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing" (ICASSP85).

[22] Norvig, P. (1987) *Inference in Text Understanding.* In 'Proceedings of the AAAI-87'.

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

[14] Thus, although the massively parallel machines are yet to come to be utilized by the end-users of speech translation systems, we have shown the theory and an implementation that whenever personal massively parallel computers are available, we hope to see DmTrans running for businesspeople and for travelers in their personal machines.

4] Poesio, M. and Rullent, C. (1987) *Modified Caseframe Parsing for Speech Understanding Systems*. In 'Proceedings of the IJCAI-87'.

5] Pollard, C. and Sag, A. (1987) *An Information-based Syntax and Semantics*. Vol 1. CSLI.

6] Quillian, M.R. (1968) *Semantic Memory*. In 'Semantic Information Processing', ed. Minsky, M. MIT Press.

7] Quillian, M.R. (1969) *The teachable language comprehender*. BBN Scientific Report 10.

8] 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.

9] Riesbeck, C. and Martin, C. (1985) *Direct Memory Access Parsing*. Yale University Report 354.

0] Saito, H. and Tomita, M. (1988) *Parsing Noisy Sentences*. In 'Proceedings of the COLING-88'.

1] Schank, R. (1982) *Dynamic Memory: A theory of learning in computers and people*. Cambridge University Press.

2] Schank, R. (1986) *Explanation Patterns: Understanding mechanically and creatively*. Lawrence Erlbaum Associates, Publishers.

3] Small, S. and Reiger, C. (1982) *Parsing and comprehending with word experts (a theory and its realization)*. In 'Strategies for natural language processing' Eds. Lenhert G. and Ringle M. Lawrence Erlbaum.

4] Tomabechi, H. (1987a) *Direct Memory Access Translation*. In 'Proceedings of the IJCAI-87'.

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

6] 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'.

7] 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'.

8] 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).

9] 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'.

0] Waltz, D. and Pollack, J. (1984) *Phenomenologically plausible parsing*. In 'Proceedings of the AAAI-84'.

APPENDIX 1: Implementation

Speech recognition hardware was by Matsushita Research Institute and is used in our system by the courtesy of the Institute. In addition to the firmware written control codes, the low-level control program is written in 'C' for the device hardware. Current implementation of ΦDMTRANS runs real-time[15] on HP9000 AI Workstations and is written in HP CommonLisp. The object-code of the speech recognition control programs is directly called from inside the CommonLisp code. Also, non-real-time[16] 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 version is supported at the operating system level on 'Mach' (Rashid, *et al*[1987]) at CMU. MULTILISP is described in Halstead[1985], which is a parallel lisp developed at MIT for Concert multi-processors and is now implemented on the 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.

APPENDIX 2: Distinctive Feature Matrix Using SPE

Below is the distinctive feature matrix used in our system for Japanese:

	p	t	(c)	k	b	d	g	(*)	s	z	r
cons	+	+	+	+	+	+	+	+	+	+	+
syll	−	−	−	−	−	−	−	−	−	−	−
son	−	−	−	−	−	−	−	+	−	−	+
high	−	−	+	+	−	−	+	+	−	−	−
back	−	−	−	+	−	−	+	+	−	−	−
low	−	−	−	−	−	−	−	−	−	−	−
cor	−	+	+	−	−	+	−	−	+	+	+
voice	−	−	−	−	+	+	+	+	−	+	+
cont	−	−	−	−	−	−	−	−	+	+	−
nasal	−	−	−	−	−	−	−	+	−	−	−

	m	n	=	w	j	h	i	e	a	o	u
cons	+	+	+	−	−	−	−	−	−	−	−
syll	−	−	+	−	−	−	+	+	+	+	+
son	+	+	+	+	+	−	+	+	+	+	+
high	−	−	+	+	+	−	+	−	−	−	+
back	−	−	+	+	−	−	−	−	+	+	+
low	−	−	−	−	−	−	−	−	+	−	−
cor	−	+	−	−	−	−	−	−	−	−	−
voice	+	+	+	+	+	−	+	+	+	+	+
cont	−	−	−	+	+	+	+	+	+	+	+
nasal	+	+	+	−	−	−	−	−	−	−	−

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

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