# InfoJapan'90
# Information Technology Harmonizing with Society

THE 30TH ANNIVERSARY
InfoJapan'90
THE INFORMATION PROCESSING SOCIETY OF JAPAN

*Proceedings of an International Conference organized by the IPSJ to Commemorate the 30th Anniversary*

Part: 1

October 2 - 5, 1990

**Information Processing Society of Japan**

Proceedings of an International Conference organized by
the IPSJ to Commemorate the 30th Anniversary

# Symbolic and Subsymbolic Massive-Parallelism for Speech-to-Speech Translation:

## Hybrid Time-Delay, Recurrent, and Constraint Propagation Connectionist Architecture

Hideto Tomabechi
Carnegie Mellon University
109 EDSH, Pittsburgh, PA 15213-3890, U. S. A.
tomabech@cs.cmu.edu
and
ATR Interpreting Telephony Research Laboratories*
Sanpeidani, Inuidani, Seika-cho, Sorakugun, Kyoto 619-02, JAPAN
tomabech%atr-la.atr.co.jp@uunet.UU.NET

## Abstract

This paper describes the HMCPN (Head-Driven Massively-Parallel Constraint Propagation Network) architecture which is a hybrid architecture of symbolic and subsymbolic neural networks. We claim that traditional natural language models have assumed a monotonic compositional buildup of the meaning of constituents. However, they are inadequate in performing meaning assignment based on the dynamic recognition as a whole. Our proposed architecture handles contextually sensitive linguistic phenomena through constraint application from both *a priori* given procedural knowledge as well as subsymbolic pragmatic knowledge learned directly from the actual sentential inputs, while retaining the strict syntactic and semantic processing that has been attained in traditional computational linguistic schemes. Also, as an architecture for natural language processing, the integration of a head-driven massively-parallel constraint propagation network with a time-delay neural net and recurrent neural net is a viable paradigm for future speech-to-speech translation systems.

## 1 Introduction

"In attacking the formalist conception of arithmetic, Frege says more or less this: these petty explanations of the signs are idle once we *understand* the signs. Understanding would be something like seeing a picture from which all rules followed, or a picture that makes them all clear. But Frege does not seem to see that such a picture would itself be another sign, or a calculus to explain the written one to us." ([Wittgenstein, 1933]). Over 20 years of theoretical and practical investigation in natural language processing under the massive parallel hypothesis and the recent progress in parallel distributed (neural-net) processing of natural language is now beginning to show us a picture of human language processing, in which the long tradition of the Frege-Montague view of decomposable "intelligence" (or monotonic compositionality of "meaning") [1] leaves the inevitable impression of being obsolete.

The HMCPN (Head-Driven Massively-Parallel Constraint Propagation Network) model, introduced in this paper supports the paradigm of natural language processing as a memory activity and diverges from Fregean traditional parsers in that: 1) it presupposes the dynamic participation of symbolic and subsymbolic information from different levels of abstraction to determine the identity of constituents; 2) the *meaning* of even the smallest parts (such as lexical definitions) is modified by the sentential and extra-sentential environment because the time-sensitive representational transformation network constitutes part of the lexical configuration[2].

The model is hybrid in that the massively-parallel symbolic constraint propagation network which retains the ability to perform symbolic operations such as *variable-binding*, *compositionality*, and other symbolic inferential tasks[3] is integrated with an acoustic time-delay neural network and a recurrent neural network, both of which contribute the strength of gradient descent network learning and fully distributed representation. In particular, the subsymbolic time-sensitive patterns of activation which are captured in a recurrent network contribute contextual priming effects based on the *a posteriori* captured sentential regularities, whereas the syntactic, semantic, and pragmatic knowledge in the symbolic layers of the architecture is *a priori* given. As a speech-to-speech translation system, the HMCPN-MT system has five parts:

- Time-Delay Neural Network Phoneme Recognition

*Visiting Research Scientist.

[1] *Fregean Principle of Compositionality:* "The meaning of the whole is a function of the meaning of the parts and their mode of combination" ([Dowty, *et al.*, 1988]).

[2] In other words, the state of the recurrent network which is connected to a lexical node constitutes a part of the lexical *meaning* representation.

[3] Case-based reasoning, for example ([Martin, 1989]). Also [Pinker and Prince, 1988] claim "symbolic models of language were not designed for arbitrary reasons and preserved as quaint traditions; the distinction they make are substantive claims motivated by empirical facts and cannot be obliterated unless a new model provides equally compelling accounts of those facts."

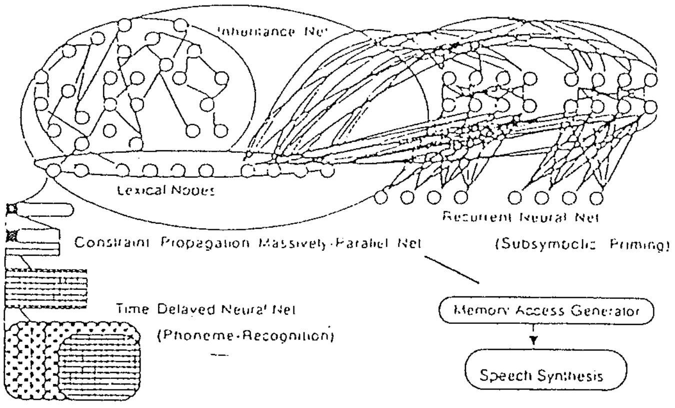

Figure 1: HMCPN-MT Architecture

- Constraint Propagation Inheritance Network
- Subsymbolic Priming Recurrent Neural Network
- Memory Access Generator
- Speech Synthesis Module

In this paper, we will be focusing our discussions on our architectural paradigm for symbolc and subsymbolic recognition of natural language input. We will also review our underlying philosophy of natural language recognition. We will include some of our proposals for implementing our model on a parallel machine hardware. We will not discuss the issues of generation in this paper, which is discussed in detail in [Tomabechi, *et al.*, 1989].

## 2 Overview of our paradigm

### 2.1 Critique of Traditional Paradigm

"A word's full concept is defined in the memory model to be all the nodes that can be reached by an exhaustive tracing process, originating at its initial, patriarchical type node, together with the total sum of relationships among these nodes specified by within-plane, token-to-token links." ([Quillian, 1968]). [Quine, 1971] observes "Concepts are not composed, layer by layer, from more primitive, already acquired concepts; instead, the whole cluster of concepts forms a complexly interacting web with no clear levels. The task of acquiring a single concept is at best an idealization, for in learning a new concept we will almost certainly alter others, so that our beliefs are as consistent, coherent and accurate as we can make them." These original claims sound intuitively unquestionable. However, in the long tradition of natural language processing, it has been taken for granted that we can extract a certain portion of a declarative semantic network (or simply a frame type predicate value aggregation) and call it a semantic representation of an input. Sometimes these frame representations, isolated from the original semantic network with certain variable instantiations, are called "conceptual structures" and viewed as a meaning representation of an input sentence. However, there is a need to distinguish 'representing' and 'assigning meaning'. For example, a machine (program) readable 'representation' of a linguistic input is no more than a scheme of changing the form of represcntional content of the input from one representational scheme (such as English, which a large number of humans use) to another (such as configurational trees and functional structures, which CommonLisp programs use). In other words, a parser which converts English to functional structures has not 'understood' a sentence by such an activity. It has simply changed the representational form of a linguistic input into something else[4]. Thus, a better way to characterize existing 'natural language understanding programs' and 'parsers' would be 'representation transformers' under our view.

Following the tradition of Frege, the method of building up the semantics of a sentence as a composition of the smaller parts (words and phrases) of the sentence is the prevailing method which traditional representational transformers (parsers) utilize in "building up" their meaning representations. Some schemes use direct semantic compositional rules such as the ones originally introduced by Montague (e.g. [Montague, 1970]). Others use more implicit schemes of building up compositional semantics, such as through a series of feature-structure unifications. While we are not proposing to abandon the basic principle of combining the semantics of parts to build the whole, we claim that the meaning of the parts may also be determined by the meaning of the whole. A problem with the schemes of 'compositional semantics' is that at any given point in the analysis, constraints for building constituents are local and predetermined (such as in unification augmented context-free parsers) and every locally acceptable structure (which may be numerous) is built-up from the input. This is because there is no way in the existing framework to dynamically modify the constraints for constituent build-up based upon the global environment at the time of recognition. A further paradox[5] for the schemes of the traditional compositional semantics is that in order to assign meaning to the result of the composition in the utterance environment (context), the elements of composition need to be decomposed again to be evaluated in the given environment. [6] This is because an environment cannot be determined (in a unidirectional upward composition) until the whole is composed. In the Fregean principle of compositionality, there

[4] For example, in Montague's framework, "translation into Intentional Logic is merely a convenience in giving the semantic interpretation of a natural language, not an essential part of the process" ([Dowty, *et al.*, 1988]). It is the *evaluation* of the representation in the context (or model) which performs the meaning assignment.

[5] Criticisms for the Fregean Principle of Compositionality is not at all new (e.g. [Chomsky, 1975]). However, such criticism combined with the *autonomy of syntax* hypothesis has led to research in syntax being severed from semantics (and pragmatics) and to treating semantics as a black box. Thus narrowing the coverage of *linguistic phenomena* to an arbitrarily small size excluding any memory-based activity.

[6] Actually, in unification-based formalisms, if such an action is allowed, behavior of the parser is no longer capturable through operations in the subsumption lattice of feature structures. In other words, such a grammar of language is not formulatable within the unification-based grammar formalisms, such as HPSG and LFG.

is an underlying assumption that the environment for evaluating the parts is all predetermined (i.e., a Montagovian 'model' is already selected). However, in real utterance situations, such an environment is dynamic and cannot be determined without the knowledge of the whole (which itself is a target of composition).

### 2.2 Constraint Propagations

The HMCP paradigm can be viewed as a model of natural language recognition in which input (natural language or other sensory input) imposes constraints that are propagated from the lower classes in the abstraction hierarchy to the higher. In other words, the new features or constraints are reversely inherited from the lower class to the upper in order to determine the *meaning* (or identity) of the input captured by the network of concepts which received (reverse inherited) the new features from the nodes that are below them in the abstraction hierarchy. To be more precise, we view natural language "understanding" as a recognition process, in which already known concepts (ideas, episodes, memory about things, etc.) collectively receive new features which are screened through the *grammar*[7] of the language. By such an activity, the input language is recognized and identified with the existing concepts in memory, while the existing network itself is modified by accepting the constraints that are imposed by the input activations.

## 3 The HMCPN Recognition

### 3.1 Neural-Net Phoneme Recognition

Time-Delay Neural Net is a neural network which has the ability to represent relationships between events in time. Such a relational feature abstraction is learned by the network invariantly under time transitions. Input to a unit is multiplied by the number of delays (plus undelayed input). Weights are associated with each delayed input to a hidden unit and the weighted sum is passed to a semilinear threshold function to compute the output of the unit. When the Time-Delay Neural Network is used for phoneme recognition, the lowest input layer of the multilayer network receives the spectral coefficients as input. We have adopted the Time-Delayed Neural Network which was developed at ATR ([Waibel, *et al.*, 1989], [Sawai, *et al.*, 1989], etc.), and is currently jointly researched at ATR and Carnegie Mellon University. Currently, TDNN architecture has been shown to be a suitable architecture for *vocabulary-independent* connected speech recognition ([Miyatake, *et al.*, ms]) recording a rate of over 95% accuracy[8] without top-down (and lexical) selectional restrictions (i.e., full vocabulary-independence).

The phonemic knowledge in the HMCPN-MT architecture is recorded in terms of time-delayed patterns of activations captured in the hidden layers of the time-delay network. Also, the acoustic knowledge captured in the time-delay hidden layers is modular in the sense that it is vocabulary independent. In the hidden layers, no specific nodes represent the specific phoneme activations. Instead, it is the time-delayed patterns of activations which are captured in the weights of the time-delay links. Thus, the acoustic representation in the phoneme-recognition network is fully distributed. It is at the output layer that specific phonemes are activated (in sequence) as a result of the recognition of the time-delay network. Each unit in the output layer of the time-delay network is connected to a phoneme node in the symbolic inheritance network. Thus, the phonemic activation from the TDNN to the constraint propagation inheritance net is the bridge from the sub-symbolic acoustic input network to the symbolic constraint propagation network.

### 3.2 HMCP Sentential Recognition

Under the HMCP model, conceptual nodes representing argument-taking predicates carry subcategorization features which specify syntactic properties (such as case) of constituents which can fill their argument positions. Syntactic information such as case, number, and person is propagated up from noun phrases in a package of 'head features' which eventually collides with the constraints in subcategorization frames. The following three things are propagated from lexically activated nodes: 1) head-features attached to the node, 2) identity of the instance node associated with the current lexical activation (i.e., which specific instance should be associated or created with the current lexical activation) and 3) the specific cost (weight) associated with a given lexical activation[9].

Before introducing the HMCP algorithm, we would like to briefly discuss our processing principle used to attain hybrid symbolic and subsymbolic architecture. One major difficulty of constructing a hybrid architecture of a massively parallel symbolic network and a fully-distributed connectionist network lies in the fact that the representational units of the nodes in the two systems are incompatible. This is due to the fact that the granularities of node activities are different and the grain sizes[10] of mas-

[7] If we can view *the grammar of language* to be the information that maps the input language to the collection of concepts which are recognized and organized in a manner consistent with the already existent knowledge (memory) about the world, then the *grammar of language* for the conceptual inheritance network is the constraints that are imposed in order to guide (map) the recognition and reorganization of the conceptual inheritance (i.e., memory) network in order to accept the input language. As we will review in a later section, "meaning" representation in such a network is a time-sliced state of the network after the application of the constraints imposed by the input language itself, which is not extractable by isolating certain feature value pairs from the whole network.

[8] See references cited above.

[9] The cost-based ambiguity resolution schemes are discussed in detail in [Tomabechi, *et al.*, 1989] and [Kitano, Tomabechi, and Levin, 1988] and are not discussed in this paper.

[10] By way of definition, we will be using the following notion of levels of parallelism in this paper: Fine grain – the level where basic operations of the system are parallelized. For example, firing of each node, basic arithmetic operations on each input, etc.. Medium grain – the level where functional units are parallelized. By this definition, concurrent applications of various constraints at various locations of

sively parallel activities in the two networks are incompatible. The activity in the constraint propagation network is medium to coarse grain, requiring application of functional constraints (such as in role-filling activities), and the node representations are symbolic and structural. On the other hand, the activity in the neural network is fine-grain (sigmoid firings) and node representations are nothing more than simple vectors. Due to the difference in the granularity of the parallelisms (and representational units), attaining the coexistence of both systems in the same massively-parallel processing architecture is not trivial. For example, realizing neural-network recognition on massively-parallel register machine hardware (such as *Connection Machine*) may be straightforward. However, realizing a realistic massively-parallel symbolic constraint propagation mechanism on the same architecture is non-trivial. This is because the granularity of parallel symbolic processing is too large for such a machine architecture[11].

Our solution to this issue is the separation of algorithmic massive-parallelism from the fine-grain data-level massive-parallelism assumed in massively parallel spreading activation architecture through the introduction of the notion of *light weight processes*[12] (*lwps*). A *lwp* is a process that is spawned explicitly by other *lwps* (or by an initial process)[13]. When a *lwp* completes its evaluations, it simply goes away (i.e., need not be killed by an external process). By making numerous *lwps* work at the same time (with little or no synchronization between them) on different nodes, massively parallel processing can be attained without a hardware massive parallelism. Also, any number of *lwps* can work on one node and therefore, if necessary, the parallelism can be even finer than the node level parallelism. Since a *lwp* may work on a functional constraint, a mixture of fine-grain and medium-grain parallelisms can be supported. Thus our model has three distinct levels of processing: 1) *Node level:* this is the level where phonemic and conceptual nodes receive and fire activations, i.e., the representational level of memory nodes. 2) *Light weight process (lwp) level:* this is the level at which actual massively parallel processing is performed. Any number of *lwps* may be created during processing, independent of the number of nodes or processing units. 3) *Processing unit level:* this is the level of actual processing hardware. Any number of processors may be configured depending on the hardware architecture. Thus, in our model, the representational level, the process level and the hardware level are explicitly separated.

### 3.2.1 HMCP Algorithm:

We have three types of nodes in the constraint propagation network: lexical nodes, inheritance nodes, and memory-instance nodes. Lexical nodes are the nodes with phonological entries (phonemic nodes) attached to them. Two kinds of lexical nodes exist: head-node and complement-node. Head-nodes have subcategorization feature attached to them (i.e., package complement nodes). Complement-nodes do not. Inheritance nodes are the nodes which are organized as a hierarchy and are a superclasses of lexical nodes. Memory-instance nodes are the specific instances of lexical and inheritance nodes recorded in the network as experiential memory.

We have four kinds of layers in the network: 1) Static Layer (SL); 2) Potential-activation Layer (PL); 3) Activation Layer (AL); and 4) Decaying Layer (DL). The SL is where nodes by default belong. The PL is where head nodes and nodes packaged by head nodes initially belong. The AL is where nodes which received constraint propagation belong. The DL is where nodes in the AL move to after a given period. Nodes in the DL eventually move to the PL. Now let us provide the HMCP algorithm below:

```
function PREPARE-NODES;
  for (NODE in ALL-NODES) do
    if (NODE is a head)
       then PREPARE-GRAMMATICAL-LINKS(NODE);
    if (NODE is a head or an element of sucat list of a head)
       then push NODE into PL;
end;

function SENTENTIALLY-RECOGNIZE(INPUT-STREAM);
  for LEXICAL-STREAM in INPUT-STREAM do
    ACTIVATE-LEXICAL-NODE(NODE);
  GLOBAL-INCIDENTS;
end;

function ACTIVATE-LEXICAL-NODE(NODE);
  timestamp NODE with the initial activation time;
  push NODE into AL;
  create an instance of NODE;
  create an HF-MARKER containing:
                 1) head-features of NODE
                 2) pointer to the created instance.
  for all PARENT of NODE do
    recursively climb up abstraction hierarchy
    and evaluate ACIVATE-NODE(PARENT);
end;

function ACTIVATE-NODE(NODE);
 if (NODE is in PL)
    then leave HF-MARKER on NODE;
 if (NODE == *TOP-OF-INHERITANCE-NETWORK*)
    then GLOBAL-INCIDENTS;
end;

function GLOBAL-INCIDENTS;
  for (NODE in ALL-NODES) do
    INVOKE-ROOT-INSTANCE(NODE);
    INSPECT-ROOT-INSTANCE(NODE);
  start all created light-weight-processes
          ;;;; (i.e., massive-parallelism).
end;

function INVOKE-ROOT-INSTANCE(NODE);
 if (NODE is an instance of a head
     and its subcat is still unsaturated)
```

memory would be medium grain. Coarse grain - parallelism at the level of sub-modules of the whole system. Parallel processing of input in different system modules may be coarse.

[11] Also, the node communication speed in a loosely-coupled, local-memory massively parallel machine architecture (normally the target machine for hardware-supported, marker-passing algorithms) will be a bottle-neck.

[12] As supported in Mach through 'thread'. In our implementation, *lwps* are explicitly provided by CLiP Parallel CommonLisp.

[13] Each *lwp* may run on any available processing unit (processor) and is scheduled by a separate process. Each lwp is capable of accessing the entire shared memory and may lock or unlock and read/write any part of the shared memory.

```
      then create light-weight-process for
            GRAB-ROLE-FILLERS(NODE);
end;

function INSPECT-ROOT-INSTANCE(NODE);
 if (NODE is an instance of a head
     and its subcat is now saturated)
    then
         timestamp NODE with the accepted time;
         create an HF-MARKER containing:
                1) head-features of (the parent of) NODE
                2) pointer to NODE
         ACTIVATE-NODE(NODE);
end;

function GRAB-ROLE-FILLERS(NODE);
 for (ROLE in ROLES of the NODE) do
     create light-weight-process for
        applying all constraints, i.e., head-feature
        constraint, linear-precedence, obliqueness-order,
        control, etc.  If all constraints are met, fill the
        ROLE with the role-filler.
end;
```

As an input to the (symbolic recognition) top-level *sententially-recognize*, TDNN provides the HMCP sentential recognizer with a stream of phonemes. These phonemes may be *noisy*. The schemes to handle noisy input are described in [Tomabechi, *et al.*, 1989]. Here we assume the TDNN provided correct streams of phonemes[14]. The interface between the TDNN and the sentential recognizer is attained by directly connecting the phonemic nodes which are packaged by lexical-nodes in the HMCP sentential recognizer to the output units of the TDNN.

The function *prepare-grammatical-links* creates virtual grammatical nodes as instances of each head node role-filler complement inheriting all information from the role-filler complement. Any grammatical constraints may be specified on the links between the head and the grammatical instances. Grammatical constraint checking will be performed on these grammatical instances[15]. At the beginning of recognition, *prepare-nodes* is evaluated once. Sentential recognitions are performed by evaluating *sententially-recognize.* After recognition of one sentence, all AL elements are moved to the DL.

### 3.2.2 A Walk through a Parse

Let us review a HMCP parsing session by walking through the parse of *John persuaded Sandy to give Mary the book.* The verb *persuade* specifies that the entity associated with its object be shared with that of the unexpressed subject of its VP complement. In other words, *persuade* specifies that it subcategorizes for a complement which is itself unsaturated[16]. Thus, there is a dependency between the embedding object and the embedded subject. This phenomenon is known as *object control*[17].

Prior to the parse, all complement nodes (i.e. all nodes that potentially satisfy an element of a subcategorization list) are put into the Potential-activation Layer (PL). In this example, nodes corresponding to *persuade* and *give* contain subcategorization lists as the value of the subcategorization feature. In the node corresponding to *persuade*, the constraint NP[NOM] in the subcategorization list is provided with *PERSON in the *persuader (actor)* role, so *PERSON is added to the PL. *ACTION and all other concepts coindexed with subcategorized positions are concurrently added to the PL. All other nodes in the network are in the Static Layer (SL) (or in a DL if a previous utterance exists).

For example, the lexical concept representing the verb *persuade* is encoded[18] in the network as below:

```
(def-lex *PERSUADE
 (inherits-from *ACTION)
 (phonology |p| |r| |s| |w| |e| |i| |d|)
 (spelling persuade)
 (head-feature v-inf-plus)
 (control object)
 (subcat (n-nom n-acc v-inf))
 (roles (actor
        arg1
        arg2))
 (holders (*person
          *person
          *action)))
```

The head-feature *v-inf-plus* is also a node which is a part of a subsumption relation subnetwork for grammatical categories. It represents the features [maj: v, vform: inf, aux: plus][19]. Each element at one particular position in the SUBCAT, ROLES and HOLDERS lists represents the constraint for another element in the same position in the other two lists. Therefore, *n* in SUBCAT represents the subcategorization constraint for the *actor* to be a noun and the semantic (relational) constraint for this position is **person* as represented in the HOLDERS list. Since word order is captured through a separate application of obliqueness order constraints ([Tomabechi and Levin, 1989]), each set of lists is order independent. The CONTROL feature specifies the constraints for complement control relations. For example, if the value of the control feature is *object*, it postulates that its object (*arg1*) is unified[20] with the first role in the complement.

[14] In other words, *input-stream* is an ordered set of phonemes with each set representing the phonemic sequence for a word. Thus, here we can regard *input-stream* to be a sentence and *lexical-stream* to be a word.

[15] This buys us a few advantages. Among them, when one head node has two distinct role links to the same complement node (for example, *giver* and *receiver* roles to the same **person*), there will be no confusion. It also allows for parallel constraint checking for different roles going to the same node.

[16] The vocabulary in this paper describing linguistic phenomena is based on the HPSG framework ([Pollard and Sag, 1987]).

[17] For detail of handling control verbs and word order (obliqueness) constraints, please refer to [Tomabechi and Levin, 1989].

[18] The representation here is taken from our implementation using the 'HyperFrame' ([Nyberg, 1989]) frame-based knowledge representation tool.

[19] In this particular implementation, head-features and subcategorization constraints are checked by traversing the subsumption relation network. This could be performed by a unification operation as well.

[20] In our model, the notion of unification in the context of control relation is specified by the constraint that the memory instance for the *arg1* position is the same node as the memory instance for the first role position in the complement.

The speech recognition subnetwork which receives the acoustic input activates the phonemic nodes in the constraint propagation network. Once the phonemic sequence < /j//o//n/ > is recognized, lexical-node *JOHN is activated. The head-features, phonemic and other cost information, and memory-instance of *JOHN (i.e., *JOHN001, etc.) are propagated upward in the abstraction hierarchy. Phonemic cost information is used for phonemic confusion disambiguation not discussed in this paper ([Tomabechi, *et al.*, 1989])[21]. The memory-instance represents the discourse entity that *John* is referring to in the current utterance[22] for the input *John*. When an upward propagation reaches a node in the PL, in this case *PERSON, the constraints propagated (such as head-features) are left on that node in the PL. In this example, the upward constraint propagation triggered by *John* carries the head feature NP[ALL-CASE], and this is left (along with phonemic cost and memory-instance information) on the node *PERSON.

When an activation reaches the top of the inheritance network, *lwps* are spawned for all head-nodes in the network. If any head-node is already activated, the spawned *lwps* in turn spawn children *lwps* for each role of the head-node to check their constraints. Grandchildren *lwps* may be further spawned[23] for different types of linguistic constraints which may be applied nondeterministically (including subcategorization, linear-precedence, obliqueness-order constraints). Since at the input of the first word *John*, no head-node is already activated, nothing happens and all spawned *lwps* disappear.

The next word, *persuaded*, activates[24] the (lexical) head-node *PERSUADE, which is subcategorized for NP[NOM] coindexed with *PERSON. The head-nodes which are activated perform local activities to find their complement role fillers (this is performed by spawned *lwps* for each of the activated head-nodes). The constraint application activities for subcategorization, obliqueness-order, complement-order, etc. are performed on the memory-instances under the current utterance. If all constraints are met, (memory-instances of) complement-nodes fill the relational roles of their heads. In our example, *PERSUADE001 tries to find (memory-instances of subclasses of) *PERSON to fill its *persuader (actor)* role. The head-feature NP[ALL-CASE] constraints successfully meet the subcategorization constraint NP[NOM] and therefore, *JOHN001 fills the *persuader (actor)* role. NP[NOM] is removed from the subcategorization list and the parse continues looking for the other subcategorized argument of *persuade*. Other activated head-nodes (memory-instances) continue their role filler constraint application activity (performed by their *lwps*) concurrently[25]. Recognition of *to give Mary the book* continues in a similar manner. When the embedded VP *give* grabs all three complements satisfying the constraints, the memory instance node for the head *GIVE propagates its head feature upwards (which is *((Maj V) (Vform bse) (Aux Minus))*). The intermediate subject control infinitival VP head *to*[26] intern grabs the accepted VP (headed by *give*) as its auxiliary action complement (and *to* infinitival VP gets saturated) and propagates its own head feature upwards. Finally the sentential head *persuade* grabs the VP headed by *to* for *circumstance* role (*arg2*). The memory instance of *give* propagates its head-feature upwards and the sentential recognition ends. (The recognition of the next sentence in the current utterance continues.)

# 4 Subsymbolic Priming

Our recurrent network under study is based on Elman's Simple Recurrent Network (SRN) ([Elman, 1988]), which we modified to predict two consecutive words rather than just one by adding an additional context layer and hidden layer (Figure 2). The context units in the first context layer are fully connected to both the first hidden layer and the second hidden layer. The input units are also fully connected to both hidden layers. The context units in the second context layer are fully connected only to the second hidden unit layer. Each hidden layer is fully connected to only one output layer. Thus we have 7 weighted connection layers in the whole recurrent network[27].

The outputs of hidden units in the first hidden layer and the second hidden layer at time t - 1 are copied one to one into each respective context unit. At time t during the forward propagation, the copied hidden patterns of activations from the previous forward propagation are fed into the hidden layers along with the input unit activations. [28]

[21] Other cost information includes reverse cost that is given by the subsymbolic recurrent network as contextual priming. Detailed discussions of the schemes to use recurrent net activations as reverse cost in the HMCP network is found in ([Tomabechi, ms]).

[22] [Kitano, Tomabechi, and Levin, 1988] and [Tomabechi, ms] discuss the schemes for identity resolution when multiple candidate discourse entities exist for a noun phrase using top-down ([Kitano, Tomabechi, and Levin, 1988]) and subsymbolic ([Tomabechi, ms]) contextual priming.

[23] Since at this grandchildren level, *lwps* need to be coordinated through 'and' parallelism, depending upon implementations, parallel spawning may not be advantageous over sequential constraint satisfactions. Such a trade-off between making the grain size finer and an increase in overhead varies depending on the specific machine architectures.

[24] Nodes for past tense morphology inherit all lexical node information from the default finite verb forms except that (TENSE PAST) is added to the verb form features.

[25] The system's recognition is massively-parallel in nature and multiple subcategorizations can be active at a given time, as well as different hypotheses for discourse entity reference and phonemic, lexical and conceptual ambiguity.

[26] Our syntactic constraints are based on the HPSG analysis

[27] Also, in this configuration, it is possible to increase the number of context-hidden-output layer sets. Each additional hidden layer will receive activations from the input layer, its own added context layer, and preceding context layer(s).

[28] We have adopted Quickprop ([Fahlman, 1988]) as the backpropagation learning algorithm which uses:

$$\Delta w(t) = \frac{S(t)}{S(t-1) - S(t)} \Delta w(t-1)$$

where S(t) and S(t-1) are the current and previous values of $\partial E/\partial w$. We treat a whole dialog as one data set, because we would like to capture the recognitions of preceding sentences in the dialog to influence the recognition of current sentences. We reset the slopes and deltas once at the beginning of each epoch. Also, weight-updates is

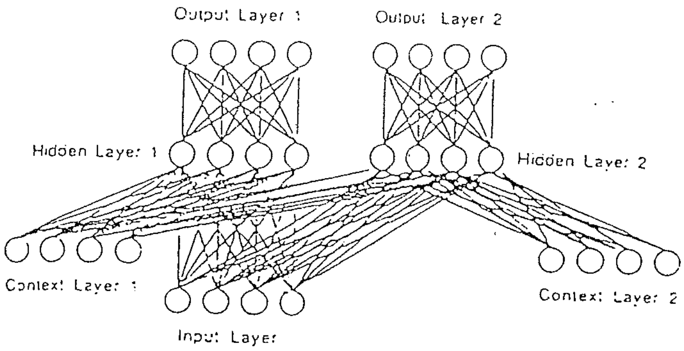

Figure 2: Recurrent Network with Two Context Unit Layers

Instead of representing the token of each word as the input and the token of the next word as target as [Elman, 1988] and [Servan-Schreiber, *et al.*, 1988] did, we have encoded:

- syntactic head-features (major categories, forms[29], etc.) including agreement features.
- locations of nodes in the inheritance hirarchy which is decodable from left to right in descending levels of abstractions[30].

The decoded locations in the inheritance hierarchy points to the nodes in the symbolic network[31]. Because of this feature-bundle representational scheme, our network learns to predict the head-features and semantic features which can be used by the symbolic part of the system to assist contextual disambiguation and other inferences.

By compensating for the lack of context sensitiveness of the syntactic and semantic constraints on discourse entity reference resolutions, unbounded dependencies, and other phenomena captured in the constraint propagation network, the 'context sensitive' learning of the recurrent network can assist contextual decision making.

## 5 Discussion:

The interaction of the constraint propagation inheritance network and the recurrent network can be viewed as an interaction between the initially encoded symbolic grammar of language and the learned subsymbolic grammar of language. In the constraint propagation network, the syntactic knowledge is provided in the head lexical nodes in the form of subcategorization lists. Also, further constraints written in the lexical head nodes can restrict the recognition of sentential configurations such as *control* constructions. Semantic knowledge in the constraint propagation network is also *a priori* provided in the form of an inheritance hierarchy and role packaging links from the head nodes to the complement nodes. In the recurrent network, it is the patterns of activations of the lexical nodes which are learned by the network. Therefore, the grammar of the language (without any distinction of syntax, semantics and pragmatics) is simply acquired from actual input words and is not originally provided. The implication of this cooperative activity of the *a priori* given conceptual network and the *a posteriori* acquired knowledge of sentential regularity in the recurrent gradient descent learning network is that we now have an architecture that begins to diverge from the Fregean compositional semantic schemes.

As [Servan-Schreiber, *et al.*, 1988] observe, "In the simple recurrent network, internal representations encode not only the prior event but also relevant aspects of the internal representation that was constructed in predicting the prior event from its predecessor. When fed back as input, these representations provide information that allows the network to maintain prediction-relevant features of an entire sequence." The representations in the recurrent network and the time-delayed network are fully distributed and subsymbolic whereas the representations in the constraint propagation network are symbolic and local. In the whole HMCPN-MT network, the meaning of the input language is dynamically captured at each point of activation (time t), and is fully influenced by the utterance recognitions at preceding time points (t - n). The activity of *meaning assignment* is performed as activations of different layers at different parts of each network at each time point, and the time-sliced state of the whole network itself is the meaning representation. Because the representation is distributed (fully distributed in TDNN and RNN and partially distributed as activated nodes in the CPN) and time-sensitive, we cannot cull out a specific portion of the whole network and call it a meaning representation.

## 6 Conclusion

It is our claim that the meaning of a sentence cannot be represented in a stand-alone representation built by mechanisms such as traditional syntax/semantics parsers. In other words, we are claiming that sentence meaning is idiosyncratic and specific to the time of recognition. Also the monotonic buildup of semantic compositions which has been assumed in traditional natural language processing systems is inadequate since meaning assignment cannot be severed from the whole (symbolic and subsymbolic) memory of the natural language recognizer. We have proposed an architecture to support a hybrid massively-parallel natural language recognition at both symbolic and subsymbolic levels with different levels of abstractions interacting with one another during constraint propagation. Given that what is being pointed to by a symbol changes dy-

performed once per epoch.

[29] Such as for verbs, distinctions of finite, present particple, passive, infinitival and gerundive forms. For noun, distinctions of expletive extrapositions, pseudocleft, non-reflexive and reflexive pronouns, and others.

[30] Our method is to devide the vector into groups, each group representing a unique level in the abstraction hirearchy. Each bit in a vector group represents the branching point of the next level down. The details of this scheme and the results of our experiments using the scheme are described in [Tomabechi, ms].

[31] Activations from the recurrent network is used as reverse costs in the symbolic network.

namically even at the lowest level of the symbolic network (i.e., at the level of lexical-nodes), due to the time-sensitive (context-sensitive) changes of the recurrent network which constitutes part of the lexical-node definitions, our model proposes the context-sensitive *meaning* assignment of *parts* which is influenced by the *whole* which is not simply a contextual choice from an initially provided finite set of lexical (or structural) choices. The lexically (symbolically) pointed memory entity itself is *a posteriori* modified during the recognition and shaped up idiosyncratic to the time of the utterance.

As a scheme for practical natural language processing, through the testing of hypotheses both by the *a priori* given (constraint propagation inheritance net) and the *a posteriori* learned (recurrent net) knowledge of grammar[32], we claim we can minimize the danger of having the natural language system be dependent on *ad hoc*, toy-domain schemes for contextual and lexical ambiguity resolution. We believe that integrated symbolic, subsymbolic natural language recognition is a viable model for future robust natural language processing and machine translation systems.

## ACKNOWLEDGMENTS

The author would like to thank Akira Kurematsu, Tsuyoshi Morimoto, Hitoshi Iida, Hidefumi Sawai and other members of ATR. Thanks are also due to Jaime Carbonell, David Evans, Carl Pollard, Alex Waibel, Rich Thomason, David Touretzky, Lori Levin, Sergei Nirenburg, Masaru Tomita and other members of the Carnegie Mellon community. I would also like to thank Jun-ichi Tsujii, Hiroaki Kitano, Margalit Zabludowski, and Kiyoaki Hara for their continued support.

## Appendix: Implementation

The HMCP recognition and the recurrent-network are implemented using Allegro CLiP version 3.0.3 which is a parallel Common Lisp from Franz Inc. The system is running on a Sequent Symmetry which is a tightly coupled multiprocessor shared memory machine running DYNIX 3.0 parallel Unix. Light weight processes and their scheduling are directly supported by CLiP. Parallel implementation of HMCP was originally done on Multilisp running on Mach at CMU. A serial lazy evaluation version is also running on CMU-CommonLisp. The TDNN is implemented in C. The spectral recognition of the TDNN is non-realtime in the current implementation. A work is underway to integrate the TDNN recognition at realtime.

[32] Which can be a top-down discourse knowledge, semantic selectional restrictions, syntactic head/subcategorization constraints, or acquired statistical knowledge about the regularity of actual sentential uses.

## References

[Chomsky, 1975] Chomsky, N. "Questions of the form and interpretation". *Linguistic Analysis 1*, 1975.

[Dowty, *et al.*, 1988] Dowty, D.R., Wall, R.E., and Peters (1981) *Introduction to Montague Semantics*, D.Reidel Pub., 1981.

[Elman, 1988] Elman, J. *Finding structure in time*. CRL TR-8801. Center for Research in Language, University of California, San Diego, 1988.

[Fahlman, 1988] Fahlman, S. E. 'Faster-Learning Variations on Back-Propagation: An Empirical Study' In *Proceedings of the 1988 Connectionist Models Summer School*. 1988

[Kitano, Tomabechi, and Levin, 1988] Kitano, H., Tomabechi, H., and Levin, L. "Ambiguity Resolution in the *DmTrans Plus*". In *Proceedings of the Fourth Conference of the European Chapter of the Association for Computational Linguistics*, 1988,

[Martin, 1989] Martin, C. "Case-based Parsing" in Riesbeck, C. and Schank, R., eds., *Inside Case-based Reasoning*, Lawence Erlbaum Associates.

[Miyatake, *et al.*, ms] Miyatake, M., Sawai, H., Minami, Y., and Shikano, K. "Integrated Training for Spotting Japanese Phonemes Using Large Phonemic Time-Delay Neural Networks". In *IEEE Proceedings of International Conference on Acoustics, Speech and Signal Processing, S8.10, Vol 1 (ICASSP'90)*, 1990.

[Montague, 1970] Montague, R. "English as a formal language", *Linguaggi nella e nella Tecnica, 1970*. Reprinted in *Formal Philosophy: Selected Papers of Richard Montague*, ed. R.H. Thomason, Yale University Press, 1974.

[Nyberg, 1989] Nyberg, E. *The HyperFrame User's Guide Version 1.0.* Technical Memo. Cognitive Research Laboratories. 1989.

[Pinker and Prince, 1988] Pinker, S., and Prince, A. "On language and connectionism: Analysis of a parallel distributed processing model of language acquisition". In Pinker, S., and Mehler, J. ed., *Connections and Symbols*, MIT Press, 1988.

[Pollard and Sag, 1987] Pollard, C. and Sag, A. *Information-based Syntax and Semantics*. Vol 1, CSLI, 1987.

[Quillian, 1968] Quillian, M.R. "Semantic Memory". In *Semantic Information Processing*, ed. Minsky, M. MIT Press, 1968.

[Quine, 1971] Quine, W. V. "Two dogmas of empricism". In *Readings in the Philosophy of Language*, ed. Rosenberg, J. F., and Tavis, C.. Prentice-Hall, 1971.

[Sawai, *et al.*, 1989] Sawai, H., Waibel, A., Haffner, P., Miyatake, M., and Shikano, K. "Parallelism, Hierarchy, Scaling in Time-Delay Neural Networks for Spotting Japanese Phonemes / CV-Syllables". In *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, 1989.

[Servan-Schreiber, *et al.*, 1988] Servan-Schreiber, D., Cleeremans, A., and McClelland, J. 'Encoding sequential structure in simple recurrent networks'. CMU-CS-88-183, Carnegie Mellon University, 1988.

[Tomabechi and Levin, 1989] Tomabechi, H. and Levin, L. "Head-driven Massively-parallel Constraint Propagation: Head-features and subcategorization as interacting constraints in associative memory", In *Proceedings of The Eleventh Annual Conference of the Cognitive Science Society*, 1989.

[Tomabechi, *et al.*, 1989] Tomabechi, H., Kitano, H., Mitamura, T., Levin, L., Tomita, M. *Direct Memory Access Speech-to-Speech Translation: A Theory of Simultaneous Interpretation*. CMU-CMT-89-111, Carnegie Mellon University, 1989.

[Tomabechi, ms] Tomabechi, H. 'Feature-based Dual-Recurrent Neural Network for Symbolic/Subsymbolic Constraint Interactions', Manuscript. Carnegie Mellon University 1990.

[Wittgenstein, 1933] Wittgenstein, L. "The Proposition, and Its Sense". In *Philosophical Grammar* Ed. Rhees, R.. Trans., Kenny, A., U of California Press., 1974.

[Waibel, *et al.*, 1989] Waibel, A., Hanazawa, T., Hinton, G., Shikano, K. and Lang, K. "Phoneme Recognition Using Time-Delay Neural Networks," IEEE, *Transactions on Acoustics, Speech and Signal Processing*, March 1989.