Symbolic Connectionism in Natural Language Disambiguation

Samuel W. K. Chan and James Franklin

***Abstract*—Natural language understanding involves the simultaneous consideration of a large number of different sources of information. Traditional methods employed in language analysis have focused on developing powerful formalisms to represent syntactic or semantic structures along with rules for transforming language into these formalisms. However, they make use of only small subsets of knowledge. This article will describe how to use the whole range of information through a neurosymbolic architecture which is a hybridization of a symbolic network and subsymbol vectors generated from a connectionist network. Besides initializing the symbolic network with prior knowledge, the subsymbol vectors are used to enhance the system's capability in disambiguation and provide flexibility in sentence understanding. The model captures a diversity of information including word associations, syntactic restrictions, case-role expectations, semantic rules and context. It attains highly interactive processing by representing knowledge in an associative network on which actual semantic inferences are performed. An integrated use of previously analyzed sentences in understanding is another important feature of our model. The model dynamically selects one hypothesis among multiple hypotheses. This notion is supported by three simulations which show the degree of disambiguation relies both on the amount of linguistic rules and the semantic-associative information available to support the inference processes in natural language understanding. Unlike many similar systems, our hybrid system is more sophisticated in tackling language disambiguation problems by using linguistic clues from disparate sources as well as modeling context effects into the sentence analysis. It is potentially more powerful than any systems relying on one processing paradigm.**

***Index Terms*— Bayesian network, constraint satisfaction, hybrid systems, natural language understanding, neural network applications, semantic analysis.**

I. Appeal of Symbolic Connectionism in Language Understanding

COGNITIVE linguists view language as a conventional system for coding communicative intent and grammar as a schematic coding system that is induced from a large number of semantic associations and rules of meaning [22], [23]. They have sought to show that neither the form nor the meaning of expressions can be adequately described without reference to speakers' encyclopedic knowledge, their construction of mental models, and their ability to map concepts from concrete to abstract domains. Johnson-Laird [18] has shown that semantic and functional motivations are sought for grammatical patterns. Semantic resolution is viewed as the consequence of constraint satisfaction. In view of this, the correspondences between cognitive linguistics and connectionism are intriguing. Connectionist frameworks have potential for capturing important aspects of linguistic behavior. The parallelism of connectionist networks supports a style of knowledge representation in which decisions are not based on individual rules with large conditions and actions which are more likely to be brittle, but rather on interactions between multiple, simpler connections. There are general advantages to having smaller units of knowledge operating in parallel relative to having larger units of knowledge operating individually. The meaning evoked by an utterance in a connectionist architecture is the result of narrowing down the search space of the possible meanings of the successive words in the utterance. Parallel constraint satisfaction can in principle capture the most striking aspect of human expert performance: experts tend to arrive quickly at a small number of the best solutions to a problem, without serial search through alternative possibilities. Connectionism is in a position to account for the behavior of both regular and irregular patterns.

While current research appears to indicate that the connectionist approach may be better for modeling many cognitive processes and it seems there is a developing trend in applying connectionism in understanding language structures [5], [11], [26], [27], [33], [36], [43], there remains a crucial difference between human cognition and that of other animals. It seems unlikely the difference can be captured only by semantic associations without symbolic manipulation [28], [29], [32]. Exploring human thinking and language require more than simply explaining the purely associative mechanisms found in most animals, although understanding these mechanisms is necessary for a deeper understanding of human cognition. It is clear that the human style of language processing involves both discrete symbol manipulation, and the grounding of those symbols in lower-level neurally implemented associations [13]. Thus a connectionist science which addresses only associative learning in the brain, without regard for the symbolic cognitive abilities resulting from that learning, is inadequate for a full understanding of human cognition. Human problem-solving requires abilities in symbol manipulation and representation which are not well modeled by the current connectionist approach. Scholars of language have been among the first to stress the importance of symbolic rules in describing human behavior. Knowledge representation is an integral part of the

Manuscript received February 25, 1997; revised January 6, 1998.

S. W. K. Chan is with the Department of Computer Science and Language Information Sciences Research Centre, City University of Hong Kong, Hong Kong.

J. Franklin is with the School of Mathematics University of New South Wales, Sydney, Australia.

Publisher Item Identifier S 1045-9227(98)06184-0.

learning system in connectionist networks [14]. There is no reason not to incorporate such an integration of knowledge representation and learning into a massively parallel micromodular symbolic system. At the same time, while there have been several approaches to modeling symbolic behavior with connectionist networks, they have not yet come close to capturing the essence of it. Symbolic connectionism, as the name implies, is based on the integration of theoretical ideas drawn from both symbolic and connectionist models. Hybrid models will certainly have a better chance of modeling complex linguistic behavior. Some hybrid models have already been proposed; they are reviewed and compared with our system in Section VI. Our motivation is to take seriously what has been discovered about the interactions between these two processing paradigms in language understanding. In the following sections, we will present an architecture to explain how symbolic connectionism can be incorporated in language disambiguation.

II. The Basic Rationale and System Architecture

This section proceeds to sketch the architecture itself. We continue to concentrate on describing the methodology of the framework. The following section illustrates the basic rationale and the overall structure of the system is briefly sketched.

A. The Basic Rationale

The reader of a text is faced with a formidable task: recognizing the individual words of the text, determining how they are structured into sentences, and deciding the explicit meaning of each sentence in the face of ambiguities. Understanding a text is considered to involve a series of specific processing phases whose final result is a complete semantic, mental representation [18], [20]. This result is not so much a representation of the text, but rather of what the text is about. On top of that, the reader must also make inferences about the likely implicit meaning of each sentence, and the connections between sentences. In trying to give a complete account of how a reader interprets sentences which occur as part of a larger discourse, it is useful to draw a distinction between two main types of information which may contribute to the understanding of the sentence. On the one hand, there is information given in the sentence itself which determines the meaning of the sentence. On the other hand, there is also information deriving from the adjacent sentences which determines the significance of the current sentence, when used in that particular context and under those particular circumstances. A reader can only build a full interpretation of sentences by recovering relevant contextual information in these two main types of information. As a consequence, in this article, three fundamental issues in natural language understanding are addressed.

- Language understanding should be paragraph-based, in order to bring about a coherent “thought” about the sentences. The paragraph is the smallest complete linguistic representation of the main ideas that the text is about.
- Semantic resolution should combine knowledge from a diversity of information sources in connecting the focal sentence as well as providing a mechanism in dynamically selecting the correct hypothesis among multiple hypotheses.
- Language processing systems should have the ability to compensate for the limitations of syntactic parsing instead of requiring a perfect parse from their parsers.

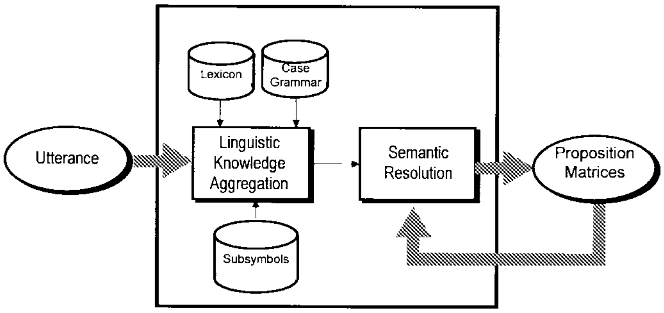


Fig. 1. The overall architecture of the system.

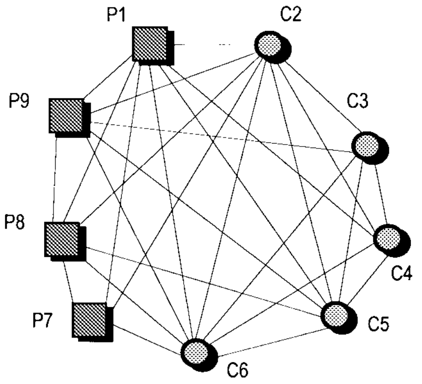
In the following sections, we will first describe a novel hybrid system to explain how the whole range of linguistic knowledge can be incorporated in the system. It also demonstrates how semantic meanings can be resolved by using both linguistic rules and semantic associations and how context effects can be modeled and carried over into the sentence analysis. The simulation results show the model is capable of addressing the language ambiguity problems effectively.

B. System Architecture

The schematic diagram of overall architecture proposed to disambiguate sentences is shown in Fig. 1. It consists of two major components: linguistic knowledge aggregation and semantic resolution.

An input utterance, which is expressed in terms of symbolic propositions, is first processed by a parser, which is the first stage in the linguistic knowledge aggregation module. The parser has two important tasks: first, it identifies constituents like noun phrases and prepositional phrases, and, second, it assigns the possible case-roles for each concept. A semantic relational network (SRN) which represents a network of symbolic nodes between the linguistic concepts or phrases is then formed. Each node is connected to the others to which it is relevant. In addition, the SRN contains the localized representations of concepts and is used to incorporate prior linguistic knowledge. It allows one to calculate the certainty of the nodes in the network given that the values of some of the nodes have been observed. At the same time, every node in the SRN is linked with its corresponding subsymbol vectors which are regarded as the grounding symbols [15] and that is a way of giving purely syntactic symbols a bundle of natural meaning. The subsymbol vectors are generated through a dual backpropagation architecture, as described later, in order to capture the semantic associations.

After the aggregation, an associative network for semantic resolution is constructed from the input proposition in accordance with the SRN and the subsymbol vectors. The associative network sets up a series of nodes. Each of them



Node	Description
P1	Go[Rosalind, Store]
C2	Stock
C3	Store
C4	Shopping
C5	Supermarket
C6	Bargain
P7	Destin.(Store)
P8	Patient(Store)
P9	Agent(Rosalind)

Fig. 2. The formation of the associative network in the semantic resolution after the linguistic knowledge aggregation process in analyzing the sentence "Rosalind went to a store." The associative network is a fully connected network in which each of the nodes represents one of the linguistic elements. Some of the nodes are activated in the SRN through symbolic reasoning, such as P1, C3, P7, P8 and P9, while others are the concepts of subsymbol vectors triggered in the interactions. Subsymbol vectors are the grounding symbols which encode the knowledge associated with the concepts.

represents one of the linguistic concepts or phrases that have occurred in the text being processed as shown in Fig. 2.

Connections among nodes have strength values indicating the degree of relevance of one concept to another. A semantic resolution is evoked in order to resolve the linguistic facts from sentences in a filtering fashion. As a result, multiple hypotheses about the input sentence among the nodes in the associative network are filtered out progressively by semantic resolution. The distilled hypothesis of each input sentence is stored in the form of a proposition matrix which is used as a knowledge source and can be carried over into the next input sentences. The whole architecture is intended to demonstrate how semantic meanings can be resolved by using both linguistic rules and semantic associations and how context effects can be modeled and carried over into the sentence analysis. Each component and their interactions are further described in detail as follows.

III. Linguistic Knowledge Aggregation

In order to resolve ambiguities in language understanding, our strategy in the knowledge aggregation is, first, to take in an input text, determine all the possible concepts that each word or phrase denotes, and identify all the possible case-role relationships that link these concepts. Obviously, it is a highly knowledge-intensive process. In order to utilize all the information accurately, this knowledge aggregation process is made as flexible as possible, requiring a minimum amount of customization for different sources of information. In this section, we will describe how the system is supported both by a semantic relational network and the subsymbol vectors. They are designed to provide generic information about word senses and semantic relations so that they will be able to handle texts spanning more than a single knowledge source.

A. Semantic Relational Network

The SRN is built prior to the semantic resolution of sentences. It is a Bayesian network which represents the linguistic knowledge of the input sentence and is used for semantic analysis [8]. For any input sentence, we adopt the propositional representation as proposed by many psycholinguists as the basic psychological processing units [2], [12], [19]. Initially, for each input proposition, a set of all possibly relevant nodes is extracted and forms a semantic relational network. The network we propose appears in Fig. 3 which depicts the general outline of the network. Fig. 3 shows how the information in the sentence

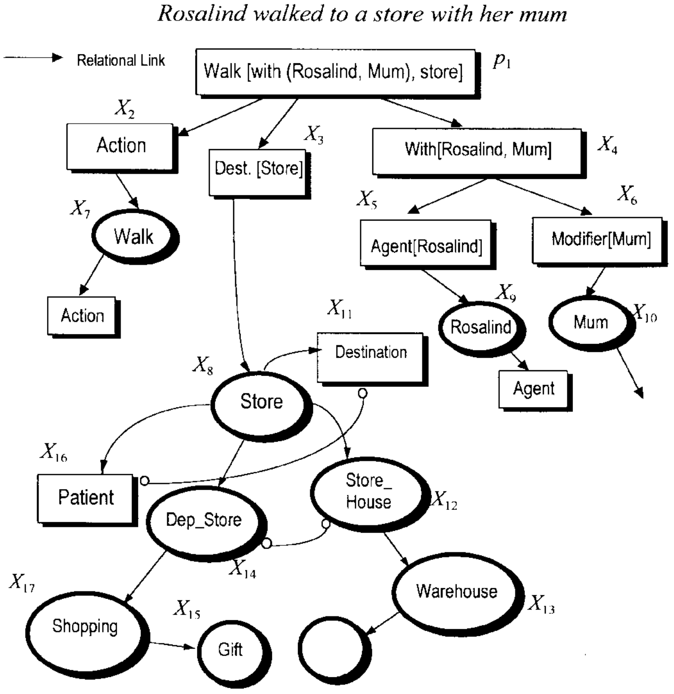


Fig. 3. An example of semantic relational network, showing especially the proposition nodes.

(S-1) *Rosalind walked to a store with her mum*

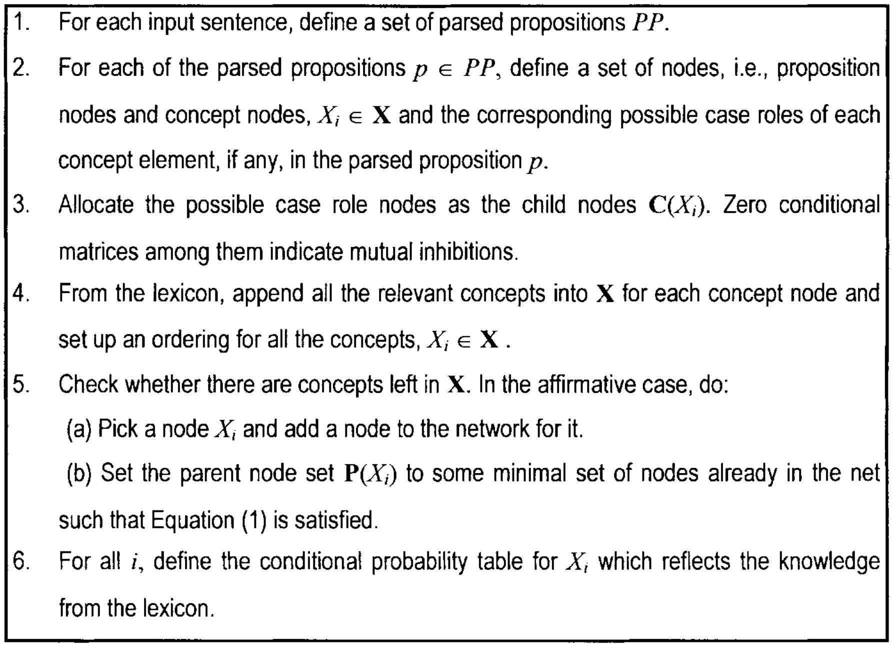
could be represented in the semantic relational network. Elliptical nodes represent the concepts, such as node X_9 representing the concept *Rosalind*, but not the lexical units. In addition, for each of the basic concept nodes, there is a case-role node associated with it. The organization of the nodes is not unique, particularly for the case-role nodes, as Fig. 3 might imply, but its spirit can be adopted for discussion purposes. Derivations can be found in the next section.

Node p_1 in Fig. 3 stands for the main proposition. It is divided into other subpropositions $X_2, X_3, \cdots, X_6$. On the other hand, if the proposition is embedded into another proposition, as in the sentence,

(S-2) *Patrick thought Rosalind walked to a store with her mum*

connections would be made to node p_1. Thus, these connections would indicate that the proposition node p_1 is the argument of the predicate *Think*, i.e., *Think(Patrick,* p_1*)*. Fig. 3 should be viewed as a part of a much larger network. Each node in a semantic relational network represents a proposition or concept. The attached nodes are all the facts we know about the concepts. For instance, node X_8 has many connections

TABLE I
ALGORITHM FOR THE FORMATION OF THE SEMANTIC RELATIONAL NETWORK



```
1. For each input sentence, define a set of parsed propositions $PP$.
2. For each of the parsed propositions $p \in PP$, define a set of nodes, i.e., proposition nodes and concept nodes, $X_i \in \mathbf{X}$ and the corresponding possible case roles of each concept element, if any, in the parsed proposition $p$.
3. Allocate the possible case role nodes as the child nodes $\mathbf{C}(X_i)$. Zero conditional matrices among them indicate mutual inhibitions.
4. From the lexicon, append all the relevant concepts into $\mathbf{X}$ for each concept node and set up an ordering for all the concepts, $X_i \in \mathbf{X}$ .
5. Check whether there are concepts left in $\mathbf{X}$. In the affirmative case, do:
   (a) Pick a node $X_i$ and add a node to the network for it.
   (b) Set the parent node set $\mathbf{P}(X_i)$ to some minimal set of nodes already in the net such that Equation (1) is satisfied.
6. For all $i$, define the conditional probability table for $X_i$ which reflects the knowledge from the lexicon.
```

which specify the facts about the concept node *Store* as in Fig. 3.

The lower part of Fig. 3 shows in more detail the concept nodes and their interconnections for determining the correct meanings. Some of the implications that the concept nodes may reflect are also included. The semantic deep cases of each concept unit, such as the agent, patient, or destination based on Fillmore's case grammars [12] are also represented in our semantic relational networks. In addition to the case role, other knowledge concepts may also be included. For instance the node *Store* may have two child nodes *Dep_store* and *Store_house* which represent the fact that two of the possible meanings of *Store* are *Department store* and *Storehouse*. This gives our semantic relational network a feature called indexing by concept. That is, if we can retrieve a concept's location in memory, we will find adjacent to that location all the facts we know explicitly about this concept. The semantic relational network provides a complete description of the domain of analysis. Given this formalism, the conditional probabilities associated with input nodes indicate the possible roles and meanings for each concept and all probabilities can be calculated using Bayes' theorem as shown in (1)

$$P(x_1, x_2, \cdots, x_n) = \prod_i P(x_i|x_j\colon X_j \in \mathbf{P}(X_i)) \qquad (1)$$

where x_i are possible values at node X_i, $P(x_1, \cdots, x_n)$ stands for $\mathbf{P}(X_1 = x_1 \cdots$ and $X_n = x_n)$, and $P(X_i)$ is the set of parent nodes of node X_i.

From the standpoint of utilizing Bayesian networks as a modeling tool in semantic disambiguation, what needs to be done is to specify the parent nodes $\mathbf{P}(X_i)$, for each node X_i and the conditional probability matrices associated with the links. In our semantic relational network, zero conditional matrices are to indicate the inhibitions between the nodes and they are denoted by the links with open circles at both ends. Zero links may arise when concepts, with both alternative meanings of a homonym, are constructed. For example, the activation of *Store_house* will inhibit the *Dep_store*, and as a result ensure the noncoexistence of both concepts in the final interpretation. The construction of the semantic relational network is summarized in Table I. Essentially, if input linguistic concepts match with the existing knowledge in the lexicon, the chunks of knowledge containing another piece of network are then constructed and appended to the corresponding nodes in the relational network. For example

$$Dep\_store \rightarrow Shopping \rightarrow Gift$$

may give positive links between the nodes *Dep_store, Shopping,* and *Gift* and they are attached onto the node *Dep_store* as constructed. Both alternative concept nodes of a homonym are constructed once encountered. In particular, for each concept, there are restrictions on the semantic cases that each linguistic entity may adopt and embed within the linguistic entity. These restrictions, in the form of probability matrices, are based on the occurrence of each concept in the proposition. For instance, the case of *Store,* in Fig. 3, can either be a destination or a patient, but not an agent or an action, since it is not at the relevant position in the proposition. Further, there are restrictions on the semantic cases that an action may adopt. For example, the action *Wake* must always take an agent or both an agent and a patient

$$Wake(John)$$
$$Wake(John, Mary).$$

This process need not be comprehensive but just powerful enough to ensure some of the plausible nodes are activated. The procedure for incremental network construction is described explicitly in Table I.

B. Formation of Subsymbol Vectors

Many problems in applied natural language processing hinge on relating lexical units to others that are similar in meaning. Our subsymbol vectors provide a similarity measure between all lexical units in order to capture lexical preference which is the key to resolving ambiguity in natural language understanding [44]. Taraban and McClelland [41] also show that the structural models of language analysis are not in fact good predictors of human behavior in resolving ambiguity. Obviously, category formulation of lexical units is a fundamental problem in natural language understanding. Unfortunately, it is not clear where the necessary information about lexical preferences is to be found. Various techniques have been tried to group concepts or lexical units into appropriately related classes. Jensen and Binot [17] describe the use of dictionary definitions for disambiguation, but it is not easy to calculate the similarity between the lexical units.

Although there are many theories for studying lexical meanings, for many linguists the sense of lexical items has at least one fundamental characteristic—it has a bundle of elementary semantic components. Imagine being asked what *boy* means. One might answer by listing the semantic triples that seem to be necessary for something to be a *boy*, say

$$\langle boy\rangle - \langle a - kind - of\rangle - \langle human\rangle$$
$$\langle boy\rangle - \langle is\rangle - \langle male\rangle$$
$$\langle boy\rangle - \langle is\rangle - \langle nonadult\rangle.$$

The sense of boy has been broken up into components. This componential approach for linguistic information is the basis of our subsymbol vector formation. The formation of subsymbol vectors, inspired by Dyer *et al.* [10], is achieved through a dual backpropagation neural network [6], [7]. It maps the triples, $\langle lexical\_item1\rangle - \langle relation\rangle - \langle lexical\_item2\rangle$, through a recursive process. To transform a specific lexical item, say $lexical\_item1$, into a subsymbol vector, we manually collect all the triples that relate to $lexical\_item1$. After constructing these semantic triples, the semantic relations, $relation$, of the lexical item are input into a backpropagation network with the $lexical\_item2$ as the target output. The resultant weight matrix in the backpropagation network is compressed. As a consequence, the subsymbol vector of the $lexical\_item1$ is gradually formed. Our subsymbol vectors are acquired through an automatic learning procedure, rather than encoded manually, in contrast to the system proposed by McClelland and Kawamoto [25]. In their system, each word is classified along a number of dimensions (such as human, softness, gender), and the resulting bits are concatenated into a long representation vector. Their hard-coded representations, however, hamper the development of the system to a larger scale. In addition, our approach differs from the subsymbolic representations proposed by Miikkulainen [26], since our subsymbol vectors have no special clone mechanism and the vectors are arranged on a semantic basis. As a result, the lexical cohesion that arises from semantic relationships between lexical units can be achieved. Fig. 4 shows some of the subsymbol vectors. Each of them is a 16-dimensional data vector which associates with its semantic meaning. It is most appropriate to regard the subsymbol vectors as sets of clues that constrain the meanings of the lexical units. The use of subsymbol vectors for sense disambiguation is also comparable to approaches that favor sense linked in a semantic network. In fact, our subsymbol vectors capture most of the useful associations found in semantic networks without specifying what the exact nature of the associations are. As a result, no redundant paths need to be traced and no extra mechanism is needed to prevent the side-tracking of all possible paths. Associations using subsymbol vectors are fairly straightforward. On the other hand, it is conceivable that lexical items with similar meanings are represented using profiles which are similar but not identical, such as the subsymbol vectors of bank and credit union. As a result, a whole spectrum of lexical similarity and cohesion can be captured. These subsymbol vectors allow a greater tolerance of errors in activation values. This is certainly not the case with conventional symbolic approaches to the representation of meanings, such as in semantic networks. The similarity measure between the subsymbol vectors is therefore defined by

$$S(x_i, x_j) = \begin{cases} \langle x_i, x_j\rangle & \text{if } \langle x_i, x_j\rangle \geq d_{\max} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $\langle x_i, x_j\rangle$ is the dot product of the subsymbol vectors x_i, x_j and $d_{\max}$ is proportional to the number of subsymbol vectors in the system.

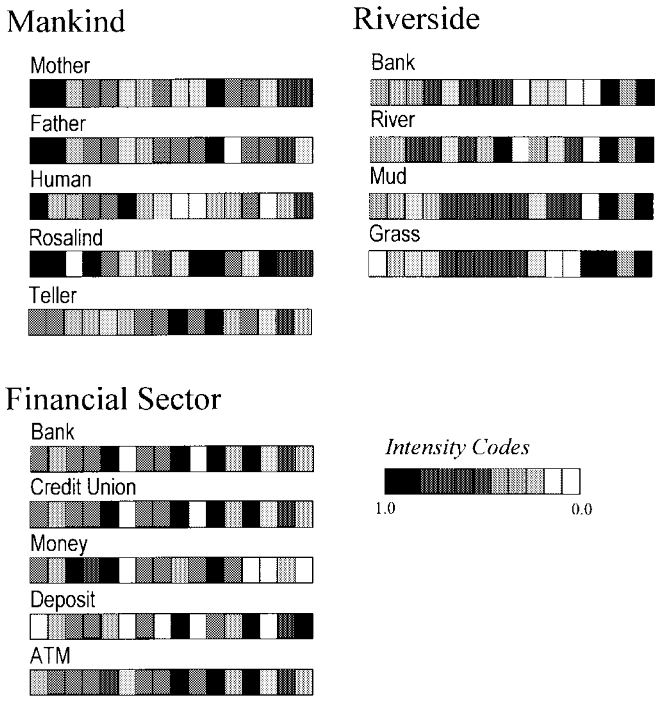


Fig. 4. Sample of subsymbol vectors formed in a dual backpropagation neural network.

In short, symbolic reasoning is hard to implement computationally, because it leads to a blowout of computational complexity, and also requires knowledge of many probabilities that in fact one does not know; connectionist associations nicely complement the deficiencies in the symbolic paradigm, and have proved sophisticated in many areas where depth of reasoning is not required, in particular in sentence interpretation. On the other hand, it is not straightforward to calculate the semantic distances between lexical items in a semantic network. In order to discover the relative strength of association between the lexical units to support the higher-level reasoning in our system, our subsymbol vectors are developed through an automatic learning procedure. These subsymbol vectors serve as a means to reflect the lexical cohesion which is the result of a chain of related words that contribute to the continuity of lexical meaning. The lexical cohesion, as shown in the simulations below, provides an easy-to-determine context to aid in the resolution of ambiguity and in the narrowing to a specific meaning of a word. Moreover, our subsymbol vectors measure the similarity not simply between pairs of words but over a succession of a number of nearby related words spanning a topical unit of the text.

C. Linking the Semantic Relational Network with Subsymbol Vectors

In order to retrieve the related semantic meanings of each lexical item in the linguistic processing, once a concept node in the semantic relational network is activated by an external input, usually the input proposition, the related subsymbol vectors will be activated subsequently as shown in Fig. 5. All the activations of concept nodes in the semantic relational network are dispersed via the manifold links between two sets of subsymbol vectors representing an antecedent and a consequent of a relational link, respectively. At the same time, some of the subsymbol vectors will also be activated despite not being involved in the relational links in the relational network. Subsymbol vectors having high similarity measure defined by (2) will also be activated. As a result, activations are propagated, in a massively parallel fashion, from an input proposition down to the subsymbol vectors. The parallelism in this structure accounts well for the similar parallelism and spontaneity in human reasoning processes [4], [38]. Moreover, the later constraint satisfaction mechanism ensures that the system avoids the problem of combinatorial explosion of search along all possible paths, which afflicts similar purely symbolic spreading activation systems. Next, we will explain how this mechanism can be incorporated into our linguistic resolution process.

IV. The Semantic Resolution Process

In most accounts of what a reader does during comprehending a discourse, it is accepted that readers generate a rich variety of potential inferences while they construct a situation model of what the text is about [18], [42]. It is undisputed that a diversity of linguistic knowledge and discourse contexts facilitates the final interpretation of a syntactically or semantically ambiguous sentence. Comprehending a text is a cyclical process, with the processing cycle roughly corresponding to propositions [20]. In each processing cycle, whenever a proposition is encountered, all its possible meanings are facilitated but then, irrelevant possibilities are rapidly eliminated in the understanding process. In our approach, after the knowledge aggregation process, an associative network for semantic resolution is then constructed from the input proposition in accordance with the SRN and the subsymbol vectors in each processing cycle. A semantic resolution process which is a kind of parallel constraint satisfaction process is then employed to form a coherent structure in the associative network. We will show the detailed resolution process in the following.

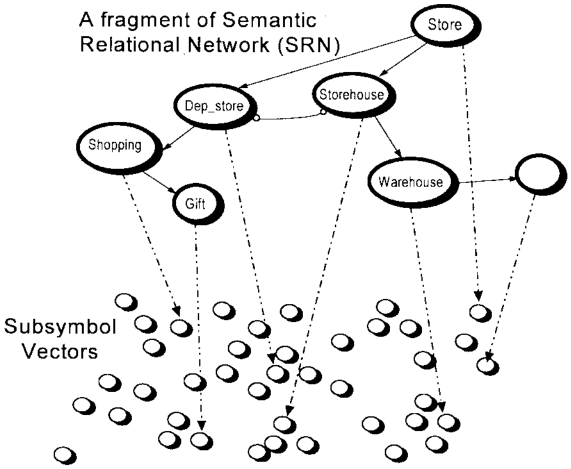


Fig. 5. A fragment showing the connections between the semantic relational network (SRN) and the corresponding subsymbol vectors. The deductions in the SRN are mirrored onto their corresponding subsymbol vectors.

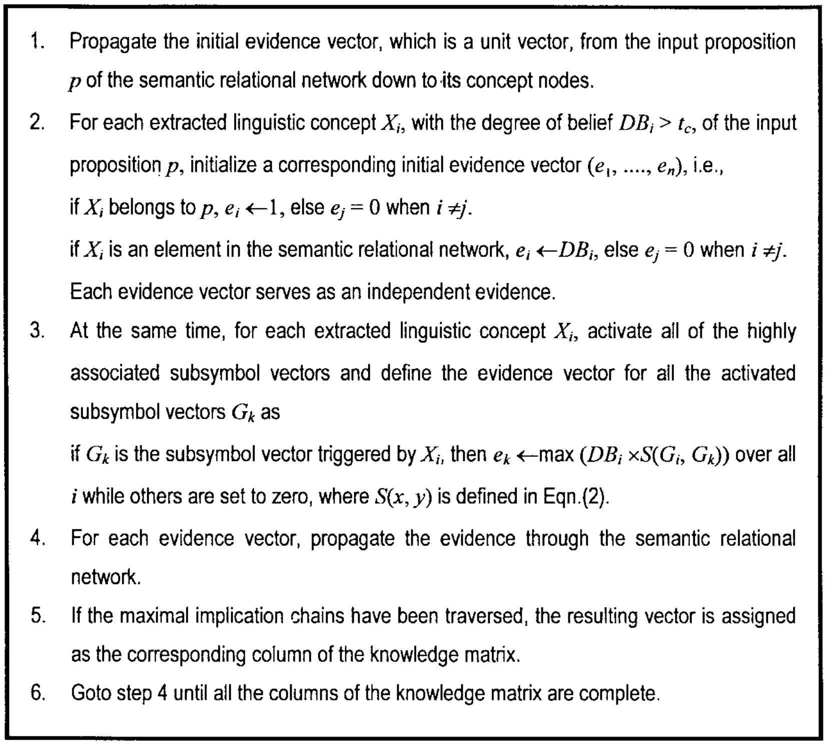
A. An Associative Network for Semantic Resolution

As shown in the foregoing sections, knowledge is first extracted from both the linguistic deduction in the SRN, semantic associations, by means of the subsymbol vectors, and their interrelations. The extraction provides relations between two pieces of implicitly or explicitly stated information. The likelihood of extraction depends on the strength of links both in the semantic relational network and the similarity of the subsymbol vectors. This process is local, associative, and without guidance and control of any central agent. Obviously, under this mechanism, the information which is less relevant, in terms of activation, to the input proposition will also be extracted. Certainly, these irrelevant concepts are unlikely to make any contribution to our semantic understanding. In order to utilize the knowledge in the SRN and the subsymbol vectors efficiently, an associative network in which every element corresponds to the highly activated concept throughout the extraction is first constructed before the semantic resolution process. To clarify the foregoing, let us denote the associative network by a knowledge matrix $\mathbf{K}$ where $\mathbf{K}(i, j)$ specifies the strength of the link between nodes i and j in the associative network. The construction of the matrix $\mathbf{K}$ makes use of the linguistic deduction in the SRN and semantic associations in the subsymbol vectors. The retrieval process involves instantiating a set of elements corresponding to the input proposition by

- first setting evidence for the input proposition to unity and activating the related concept nodes from the semantic relational network;

TABLE II

ALGORITHM FOR THE FORMATION OF KNOWLEDGE MATRIX **K** IN THE ASSOCIATIVE NETWORK



1. Propagate the initial evidence vector, which is a unit vector, from the input proposition p of the semantic relational network down to its concept nodes.
2. For each extracted linguistic concept X_i, with the degree of belief $DB_i > t_c$, of the input proposition p, initialize a corresponding initial evidence vector $(e_1, ..., e_n)$, i.e.,
   if X_i belongs to p, $e_i \leftarrow 1$, else $e_j = 0$ when $i \neq j$.
   if X_i is an element in the semantic relational network, $e_i \leftarrow DB_i$, else $e_j = 0$ when $i \neq j$.
   Each evidence vector serves as an independent evidence.
3. At the same time, for each extracted linguistic concept X_i, activate all of the highly associated subsymbol vectors and define the evidence vector for all the activated subsymbol vectors G_k as
   if G_k is the subsymbol vector triggered by X_i, then $e_k \leftarrow \max\,(DB_i \times S(G_i, G_k))$ over all i while others are set to zero, where $S(x, y)$ is defined in Eqn.(2).
4. For each evidence vector, propagate the evidence through the semantic relational network.
5. If the maximal implication chains have been traversed, the resulting vector is assigned as the corresponding column of the knowledge matrix.
6. Goto step 4 until all the columns of the knowledge matrix are complete.

- for each of these activated nodes with a high degree of belief, selecting a small number of its most closely associated neighbors from the subsymbol vectors with high similarity measure;
- for the concept node i, calculating its impact on the concept node j through the SRN;
- for all pairs of nodes that have been generated, assigning the calculated impacts into the knowledge matrix.

An important feature of the model, which has not been brought out so far, is the formation of the evidence vector. After activating all the related concepts both from the semantic relational network and the associated subsymbol vectors, each activated concept serves as an independent piece of evidence as shown in Table II and has its own evidence vector $\mathbf{E}$. The following criteria are used to determine the evidence of each concept.

1) If the concepts are directly derived from the proposition, their evidences are set to be unity.
2) If the concepts are deduced through the links in the semantic relational network, their evidence values are assigned according to their degree of belief propagated from the main proposition.
3) If entities are not involved in any links, their evidences are assigned to be the similarity measure, as discussed in (2), of their closest activated nodes.

The formation of knowledge matrix can be summarized in the algorithm as shown in Table II.

B. The Semantic Resolution Process

The retrieval process described above uses weak, robust construction rules and is followed by a spreading activation stage of hybrid inferences in the semantic resolution process. The process is used to integrate the meanings in the associative network into a coherent whole. It strengthens the contextually appropriate elements and inhibits unrelated and inappropriate ones, so that smart and complex deductions can be achieved. Each element in the associative network has some sort of activation value, i.e., central, important concepts are more highly activated than peripheral ones. In other words, the process is able to reduce the dimensionality of the stimulus so that a very complicated stimulus could act as if only a small number of independent elements are involved. This process can be used to exclude unwanted elements from the associative network. The algorithm for the resolution process is described in Table III below.

Initially, $\mathbf{U}(0)$, representing the initial activation values of all concepts, is passed into the knowledge matrix $\mathbf{K}$. The process is defined as the repeated application of the function φ until convergence. For each of the vectors $\mathbf{U}$, the vector $\varphi(\mathbf{U})$ is calculated which represents the updated activity. This procedure is applied repeatedly and is an analogy of the spreading activation process through a vector–matrix multiplication [21]. Continued spreading by repeated vector multiplication leads to equilibration. The process stops at iteration m if $|\mathbf{U}(m) - \mathbf{U}(m-1)| < t_u$. It is a kind of parallel constraint satisfaction process. The final activation vector shows how strongly related items have strengthened each other, while unrelated or contradictory items have near zero activation values. They will have become deactivated in the semantic resolution process either because they are not strongly connected with the main part of the associative network or because they are inhibited by activated nodes in the network.

TABLE III
THE ALGORITHM FOR THE SEMANTIC RESOLUTION PROCESS IN THE ASSOCIATIVE NETWORK

1. Set up the associative network and the knowledge matrix $\mathbf{K}$ as described in Table II. These elements form a fully connected network.
2. Associated with each element is a real number called its activation level, which represents the element's strength. The vector $\mathbf{U}$ is defined by $\mathbf{U}(t) = (u_1, ..., u_n)$, where u_i is the activation level for the element i and n is the number of elements extracted from the SRN and the related subsymbol vectors at some discrete time t.
3. Production firings direct the flow of activation from one element, the source, to another element through the knowledge matrix $\mathbf{K}$ and then are subjected to a normalizing operation. Mathematically, they are defined by the function
   $\varphi: R^n \rightarrow R^n$, $\varphi(\mathbf{U}) = \varphi_2(\varphi_1(\mathbf{U}))$, where
   (i) $\varphi_1(\mathbf{U}) = \mathbf{U}\,\mathbf{K}$, that is, φ_1 is a linear mapping given by multiplication by the matrix $\mathbf{K}$.
   (ii) $\varphi_2(\mathbf{U}) = \dfrac{\mathbf{U}}{\sum_{i<n} |u_i|}$
4. Continued excitation by repeated vector multiplication leads to equilibration. The process stops at iteration m if the $|\mathbf{U}(m) - \mathbf{U}(m\text{-}1)| \leq t_u$ where t_u is the tolerance, another preset threshold which is used to control the accuracy of convergence in the process.

C. Memory Representation of Sentences

After the semantic resolution process, the remaining linguistic elements are contextually relevant and all conflicts and irrelevancies should have been eliminated. They form a proposition space, though use of this term is not intended to imply that the information in a space is limited to metric or visual properties. A given proposition space is a complex collection of information that contains the *distilled* elements, lexical, syntactic, and semantic information, as well as associative and contextual information. Fig. 6 shows one of the proposition spaces obtained after analyzing the sentence, *Rosalind went to a store to buy a present.*

In order to specify how strongly each distilled linguistic element in the sentence is related to every other, the elements constructed from the sentence and their interrelations are translated into a *proposition matrix* Γ where the rows and the columns correspond to the distilled linguistic elements and the connections between them are represented by the nonzero entries. Unlike the knowledge matrix $\mathbf{K}$, the proposition matrix Γ is constructed from the asymptotic activation of each element. It represents the interrelations between elements remaining after the semantic resolution process. The strength of connection between elements is defined as

$$\Gamma(r, s) = u_r \times u_s \tag{3}$$

where u_r, u_s are the final asymptotic activation values of the rth and sth linguistic elements, respectively, as shown in the right-most column in Fig. 6(a).

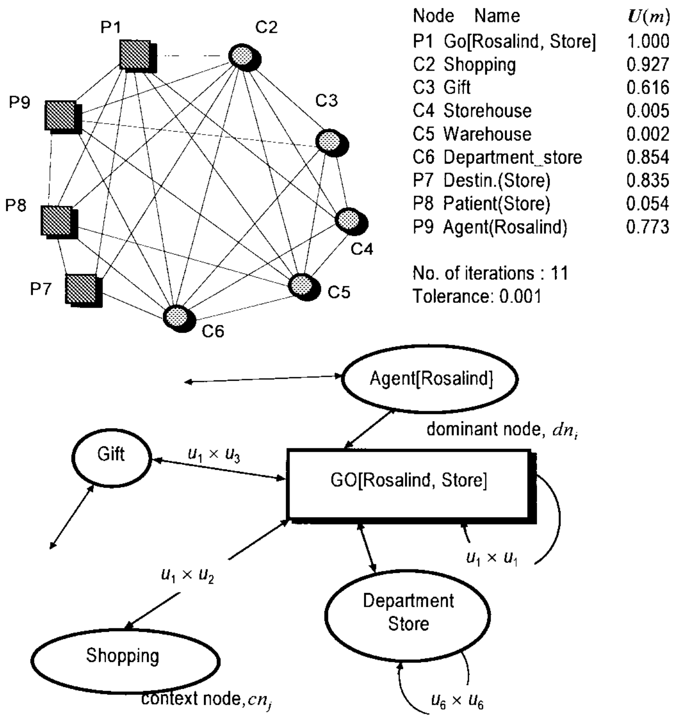
The diagonal values of Γ represent the strength of linguistic elements in the space and the off-diagonal elements represent the strength of the relations between any two elements. The element with largest strength is said to be the *dominant node* of the sentence while the others are called the *context nodes*. Fig. 6(b) shows the corresponding dominant node and context nodes. The proposition, *GO*[*Rosalind*, *Store*], is the dominant linguistic node with the largest strength, $u_r \times u_r$, for the sentence. Fig. 6 also illustrates the connection strengths between the dominant node and its context nodes and demonstrates the computation of the connection strengths in the sentence. For each input sentence, the proposition matrix Γ summarizes the interrelations among the distilled elements constructed in the sentence. In addition, the activation of the ith sentence Ω_i, is defined by

$$\Omega_i = \Sigma\Gamma(m, m) \text{ over all nodes } m \text{ in the } i\text{th sentence.} \tag{4}$$

Clearly, the activation of a sentence relies on the number of the concepts activated and persevered with in the process of semantic resolution. If the number of the activated concepts is large in a sentence, then it is highly activated with greater significance.

V. ANNOTATED SIMULATIONS AND APPLICATIONS

A system prototype is written in C and implemented on a DEC 5000/20 under the UNIX environment. Links between nodes in our semantic relational network are represented by memory pointers in C. There are around 400 nodes encoded



Node	Name	$U(m)$
P1	Go[Rosalind, Store]	1.000
C2	Shopping	0.927
C3	Gift	0.616
C4	Storehouse	0.005
C5	Warehouse	0.002
C6	Department_store	0.854
P7	Destin.(Store)	0.835
P8	Patient(Store)	0.054
P9	Agent(Rosalind)	0.773

No. of iterations : 11
Tolerance: 0.001

Fig. 6. (a) After reading a sentence, *Rosalind went to a store to buy a present*, a fully connected network with rough, or even outright contradictory nodes is formed and subjected to the semantic resolution process. The top right column shows the asymptotic activation after the process. It is apparent that the incorrect meaning of *Store, Storehouse*, is deactivated. (b) After analyzing a sentence *Rosalind went to a store to buy a present* in a semantic resolution process, a proposition space is formed containing the distilled linguistic elements which may be classified into dominant nodes or context nodes.

in the lexicon and not less than three dozen of them are triggered for every input utterance both in the semantic relational network and in the subsymbol vectors. In order to demonstrate the foregoing in more detail, in this section, we present a number of simulations that demonstrate the capability of the architecture in accounting for certain kinds of case-role resolution, semantic disambiguation and anaphora resolution, to supplement, verify and strengthen our theoretical considerations above. The detection of inconsistencies in texts has been widely studied by researchers studying linguistic comprehension. In the three different applications with detailed discussions presented below, we will exhibit how the system deals with case-role resolution, lexical disambiguation as well as anaphora resolution. Each of them shows one of the aspects of human sentence processing.

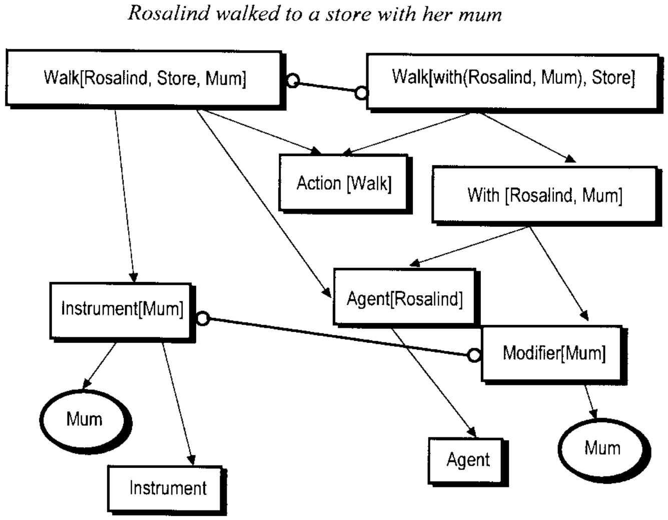
A. Case-Role Resolution

To illustrate how the model can account for applying contextual constraints in sentences to arrive at a conceptually consistent case-role resolution, let us first compare the following sentences.

(*) *Rosalind walked to a store with her mum.*

(**) *Rosalind walked to a store with her stick.*

In the former, the correct interpretation is to attach the prepositional phrase, *with her mum*, to the agent *Rosalind* as an agent modifier while, in the second case, it is more



Rosalind walked to a store with her mum
Walk[Rosalind, Store, Mum]
Walk[with(Rosalind, Mum), Store]
Action [Walk]
With [Rosalind, Mum]
Instrument[Mum]
Agent[Rosalind]
Modifier[Mum]
Mum
Agent
Mum
Instrument

Fig. 7. The semantic relational network for two possible case-role resolutions.

appropriate to attach the phrase, *with her stick*, to the verb *walked* to indicate it is an instrument of the action *Walk*. The two different interpretations in the form of propositions can be shown as follows:

(*)’ *Walk*[Agent: *with(Rosalind, Mum)*, Destination: *Store*]

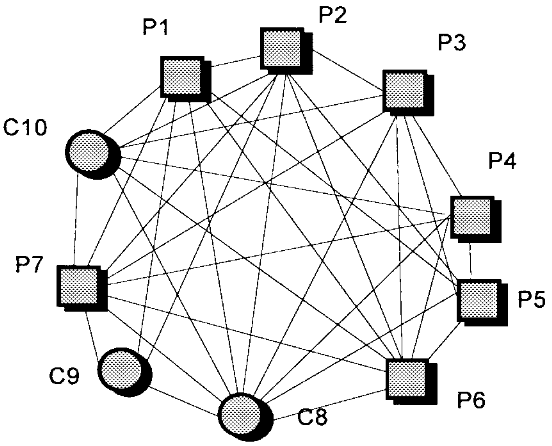
(**)’ *Walk*[Agent: *Rosalind*, Destination: *Store*, Instrument: *Stick*]

The problem remaining is how our system can cope with these two subtle case-role interpretations whenever a sentence is input, say,

(S-3) *Rosalind walked to a store with her mum.*

In order to resolve the case role in the sentence (S-3), in this empirical study, two possible interpretations are represented using different proposition formats. The modifier-role of *Mum* for the agent *Rosalind* in the sentence is represented as *Walk*[Agent: *with*(*Rosalind*, *Mum*), Destination: *Store*] while the instrument-role of *Mum* for the action *Walk* is represented as *Walk*[Agent: *Rosalind*, Destination: *Store*, Instrument: *Mum*].

Fig. 7 shows the semantic relational network of the input utterance. Both of these interpretations are activated in parallel, and concept nodes are added. To simplify the example, all the elaborations are ignored and only the most essential are displayed. Obviously, the two potential propositions are connected by inhibitor links to ensure their noncoexistence in the final interpretation. More importantly, in this example, the meaning of the last word is decisive. The concept *Mum* is connected to two possible case-roles, *Instrument[Mum]* and *Modifier[Mum]* due to the two different proposition formats and they inhibit each other as shown in the figure. Fig. 8 shows the corresponding associative network so formed. The initial activation of each node is shown as $\mathbf{U}(0)$ in Fig. 8. Due to the weak prior probability of *Instrument* given *Mum*, the activations of the node *Instrument[Mum]* and *Walk[Rosalind, Store, Mum]* are imperiled. In contrast, buoyed by the closed associations with *Rosalind, Mum,* and *human* in terms of subsymbol vectors as shown in Fig. 4, this leads to the dominant role of the node *With[Rosalind, Mum]*. As a result, by reading the final



Node	Name	$U(0)$	$U(m)$
P1	Walk[With(R, M), Store]	1.0	0.981
P2	Walk[Rosa, Store, Mum]	1.0	0.011
P3	With(Rosa, Mum)	0.0	0.807
P4	Action[Walk]	0.0	1.000
P5	Agent[Rosalind]	0.0	0.871
P6	Modifier[Mum]	0.0	0.792
P7	Instrument[Mum]	0.0	0.016
C8	Mum	1.0	0.835
C9	Human	0.0	0.814
C10	Instrument	0.0	0.054

No. of iterations : 24
Tolerance: 0.005

Fig. 8. The resulting associative network for case-role resolution.

Sentences	Propositions
1. *Rosalind walked to a store with her mum.*	Walk[With(Rosalind, Mum), Store]
2. *She wanted to give her mum a present.*	WTG[She, Mum, Present]
3. *Rosalind found out that* *everything was too expensive.*	Is[Everything, Expensive] Find[Rosalind, Is[Everything, Expensive]]
4. *She decided to knit a sweater for her.*	DTK[She, Her, Sweater]

Note: WTG = want to give
DTK = decide to knit

Fig. 9. Propositional coding for the excerpt.

activation of each element in the associative network as shown in the rightmost column of Fig. 8, *Walk[With(Rosalind, Mum), Store]* wins the competition in the network. After 24 iterations, the semantic resolution process deactivates the inappropriate instrument attachment proposition. The three nodes representing the incorrect interpretations *Instrument, Instrument[Mum],* and *Walk[Rosalind, Store, Mum]* are deactivated while the embedded proposition *With(Rosalind, Mum)* remains highly activated.

The top rightmost column in Fig. 8 indicates the asymptotic activation vector in the case-role resolution of the sentence (S-3). Obviously, if the sentence

(S-4) *Rosalind walked to a store with her stick.*

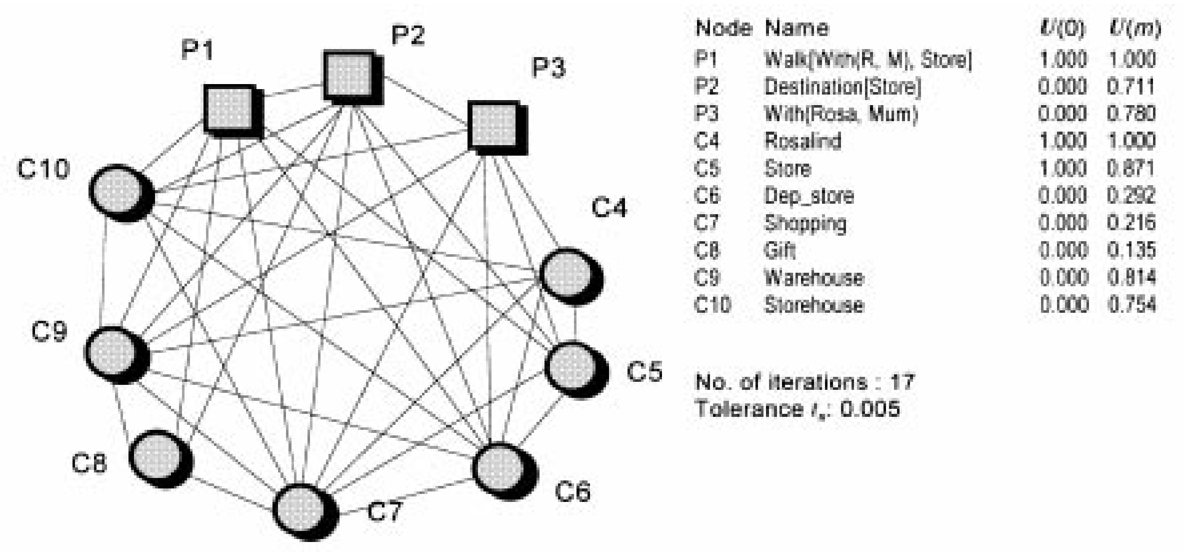
is input, the semantic relational network is broadly identical. The associative network differs only with respect to how the meaning of the concept *Stick* emerges as having a strong association with *Instrument*. The discrepancy between the groups *Rosalind* and *Stick* as well as the strong association between *Stick* and *Instrument* cause the activation of *Walk[With(Rosalind, Stick), Store]* to decay. As a result, the proposition *Walk*[Agent: *Rosalind*, Destination: *Store*, Instrument: *Stick*] will survive against the proposition *Walk*[Agent: *With(Rosalind, Stick)*, Destination: *Store*]. This demonstrates that the system has some capabilities in remedying case-role assignment by taking into account both world and linguistic knowledge, without demanding that either kind of knowledge be dominant.

B. Lexical Disambiguation

Our second application concerns the use of context to achieve word-sense disambiguation. Let us consider the following narrative excerpt: *Rosalind walked to a store with her mum. She wanted to give her mum a present. Rosalind found out that everything was too expensive. She decided to knit a sweater for her.*

The following context propositions are extracted from the narrative excerpt as in Fig. 9. Now, suppose the excerpt is processed in five cycles, one for each of the propositions.

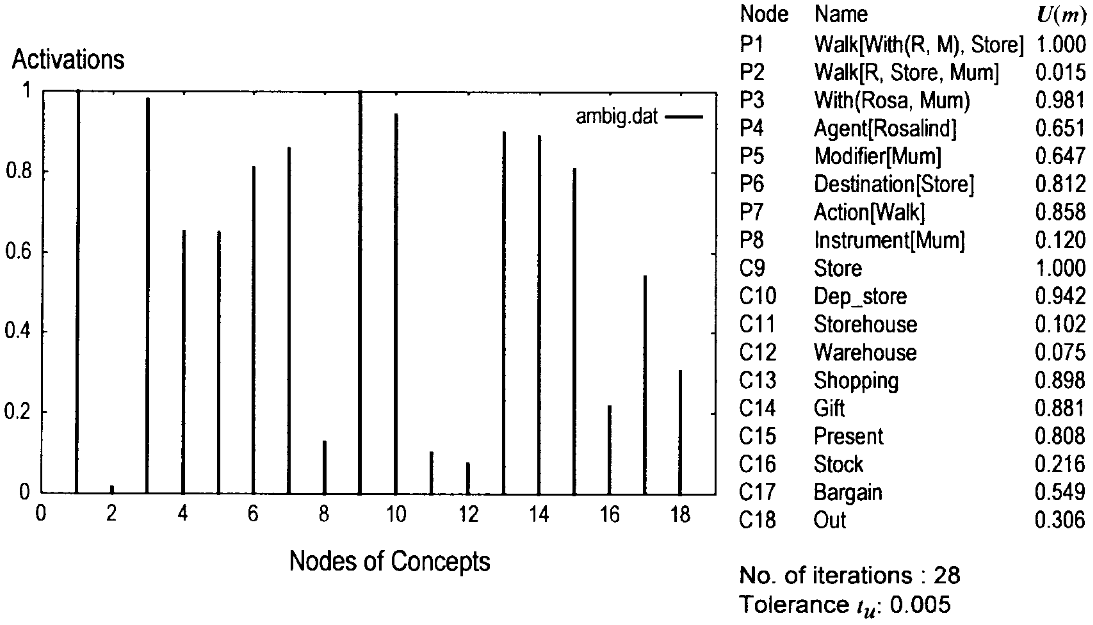
It is quite difficult to differentiate what the *Store* is in the first sentence. In order to demonstrate the disambiguating property of the framework, first suppose the word *Store* is presented as in Fig. 5. It activates the nodes *Storehouse* and *Dep_store*, plus some other associates both from the semantic relational network and their associated subsymbol vectors. The associative network is formed as in Fig. 10 and semantic resolution is achieved with the initial activation $\mathbf{U}(0)$ where Fig. 10 shows the graphical representation of the simulation result. The top right column in the figure shows the asymptotic activation vector $\mathbf{U}(m)$. It is obvious that *Storehouse* seems to dominate *Dep_store*, based on the encoded knowledge. However, the concept *Dep_store* is compatible with the context so far. While the concept *Present* is input via the next



Node	Name	$U(0)$	$U(m)$
P1	Walk[With(R, M), Store]	1.000	1.000
P2	Destination[Store]	0.000	0.711
P3	With(Rosa, Mum)	0.000	0.780
C4	Rosalind	1.000	1.000
C5	Store	1.000	0.871
C6	Dep_store	0.000	0.292
C7	Shopping	0.000	0.216
C8	Gift	0.000	0.135
C9	Warehouse	0.000	0.814
C10	Storehouse	0.000	0.754

No. of iterations : 17
Tolerance t_u: 0.005

Fig. 10. Connections in associative network in lexical disambiguation after the first sentence has been processed (only the related nodes in the disambiguation are shown).



Node	Name	$U(m)$
P1	Walk[With(R, M), Store]	1.000
P2	Walk[R, Store, Mum]	0.015
P3	With(Rosa, Mum)	0.981
P4	Agent[Rosalind]	0.651
P5	Modifier[Mum]	0.647
P6	Destination[Store]	0.812
P7	Action[Walk]	0.858
P8	Instrument[Mum]	0.120
C9	Store	1.000
C10	Dep_store	0.942
C11	Storehouse	0.102
C12	Warehouse	0.075
C13	Shopping	0.898
C14	Gift	0.881
C15	Present	0.808
C16	Stock	0.216
C17	Bargain	0.549
C18	Out	0.306

No. of iterations : 28
Tolerance t_u: 0.005

Fig. 11. The asymptotic activations of the related nodes after *Present* is involved.

sentence in the next processing cycle, the similarity measure to its activated closest node *Gift* is calculated as associated subsymbol vectors. A new knowledge matrix, $\mathbf{K}^*$ and a new activation vector $\mathbf{U}^*$, are constructed as before. The activation values of the competing linguistic concepts in the semantic resolution process are displayed in Fig. 11.

After the semantic resolution is complete, the elements representing the incorrect meaning of *Store*, such as *Storehouse*, are deactivated, while the propositions representing *Dep_store*, *Gift* remain highly activated. This shows the word *Store* is more closely related with the concepts *Dep_store*, *Gift*, *Present* or *Shopping*. After the first two sentences have been processed, the system successfully integrates the word *Store* with its context.

C. Anaphora Resolution

In the examples so far, we have used one or two sentences to build up the associative network. Another simulation concerns the use of existing sentence context to arrive at a conceptually consistent interpretation. We try to resolve the problem of multiple pronoun referents, the *She* and *her* in the fourth sentence of the above narrative excerpt as shown in Fig. 9.

(S-5) *She decided to knit a sweater for her.*

Our approach uses different propositional representations to designate alternative interpretations. Propositions are constructed for both possibilities as the referents of *She*, *DTK[Rosalind, Mum, Sweater]* and *DTK[Mum, Rosalind, Sweater]*. The process of associative elaboration generates

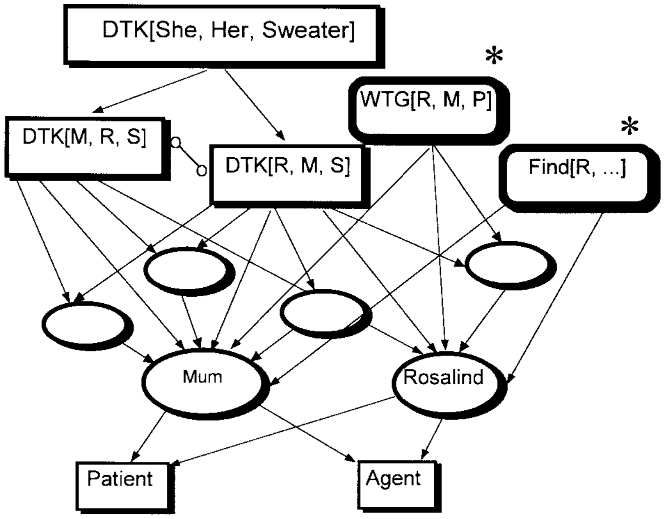


Fig. 12. A portion of the semantic relational network of pronoun resolution. The node with (*) is carried over from the previous analysis.

the additional information for each of them. Obviously, there is no way to justify either interpretation solely by the sentence itself. The meaning of the sentence relies heavily on each of the previously analyzed sentences. To cope with this, we make use of the proposition matrices Γ from the preceding sentence analysis as shown in (3). Indeed, they contain the distilled information which is carried over into the current processing cycle. They are superimposed into the current knowledge matrix and can be blended together with the current sentence as shown in Fig. 12.

Let Γ_2, Γ_3 be the distilled proposition matrices of the second and the third propositions in the excerpt, respectively. The current knowledge matrix $\mathbf{K}_4$ for the fourth sentence has the following components:

$$\Gamma_2 = \begin{array}{c|cccc|} & N3 & N4 & N5 & N6 \\ N3 & 0.8 & 0.7 & 0.6 & 0.5 \\ N4 & 0.7 & 0.9 & 0.3 & 0.8 \\ N5 & 0.6 & 0.3 & 0.5 & 0.9 \\ N6 & 0.5 & 0.8 & 0.9 & 0.6 \end{array}$$

$$\Gamma_3 = \begin{array}{c|cc|} & N3 & N4 \\ N3 & 0.9 & 0.8 \\ N4 & 0.8 & 0.9 \end{array}$$

$$\mathbf{K}_4 = \begin{array}{c|cccccc|} & N3 & N4 & N5 & N7 & N11 & N12 \\ N3 & 0.9 & 0.7 & 0.6 & 0.9 & 0.5 & 0.5 \\ N4 & 0.6 & 0.7 & 0.5 & 0.3 & 0.1 & 0.0 \\ N5 & 0.5 & 0.8 & 0.7 & 0.6 & 0.0 & 0.1 \\ N7 & 0.8 & 0.7 & 0.9 & 0.3 & & \\ & & & & & & \\ N11 & 0.6 & 0.1 & 0.0 & & 0.8 & 0.5 \\ N12 & 0.5 & 0.0 & 0.1 & & 0.4 & 0.7 \end{array}$$

where

$$\begin{pmatrix} N3 \\ N4 \\ N5 \\ N6 \\ N7 \\ \\ N11 \\ N12 \end{pmatrix} = \begin{pmatrix} Rosalind \\ Agent[Rosa.] \\ Patient[Mum] \\ Present \\ Sweater \\ \\ Agent[Mum] \\ Store \end{pmatrix}.$$

The submatrix in Γ_2 between $N3 \cdots N6$ shows the high correlation between the corresponding concepts. That is, the concept *Rosalind* is highly activated with other concepts, such as *Agent[Rosalind], Present,* and *Patient[Mum],* since the proposition matrix Γ_2 contains all the distilled information from the sentence after the semantic resolution process. This suggests *Rosalind* is the *Agent* to give a *Present* to *Patient[Mum]* in the second sentence. For the same reason, Γ_3 suggests that *Rosalind* is the *Agent* in the third sentence. In the current sentence, since *Rosalind* and *Mum* are both involved in the two alternatives, *DTK[Rosalind, Mum, Sweater]* and *DTK[Mum, Rosalind, Sweater],* the context effects from Γ_2, Γ_3 are superimposed into the knowledge matrix $\mathbf{K}_4$ under the common concepts $N3$, $N4$, $N5$ in Γ_2 and $N3$, $N4$ in Γ_3 even though the proposition matrices Γ_2, Γ_3 may not be conformable for addition. The new knowledge matrix $\mathbf{K}_4^*$ is defined by

$$\mathbf{K}_4^* = \mathbf{K}_4 \oplus (\Gamma_2 + \Gamma_3) \tag{5}$$

where

$$\boldsymbol{C} = \boldsymbol{A} \oplus \boldsymbol{B} \text{ iff } c_{ij} = \begin{cases} a_{ij} + b_{ij}, & \text{if } i, j \text{ are common} \\ & \text{concepts in } \boldsymbol{A}\&\boldsymbol{B} \\ a_{ij}, & \text{or otherwise.} \end{cases}$$

One of the most important implications in this simulation is that all the previously analyzed propositions in the excerpt behave like an group of experts in the interpretation and resolution of the ambiguities in the current proposition. Each of the proposition matrices inherits the factual and circumstantial facts from the corresponding context. They encode some of the context knowledge which can be superimposed into the current knowledge matrix and carried over into the analysis as shown in the above simulation. Obviously, the resultant knowledge matrix is more robust than an individual knowledge matrix because the linguistic information is derived from a multiplicity of sources from the analyzed sentences, making the links in the current associative network less prone to error.

The resulting activations for the two competing propositions are shown in Fig. 13. After 12 iterations in the semantic resolution process, the propositions in which *She* has been identified as *Mum* have activation values of zero, whereas the corresponding propositions, *Rosalind, DTK[Rosalind, Mum, Sweater]* have activations values of 0.822 and 1.000, respectively. The framework finds the correct interpretation for the pronoun *She.*

Although these preliminary simulation results seem promising, they should be viewed with considerable caution. For example, it may be a little simplistic to assume *She* and

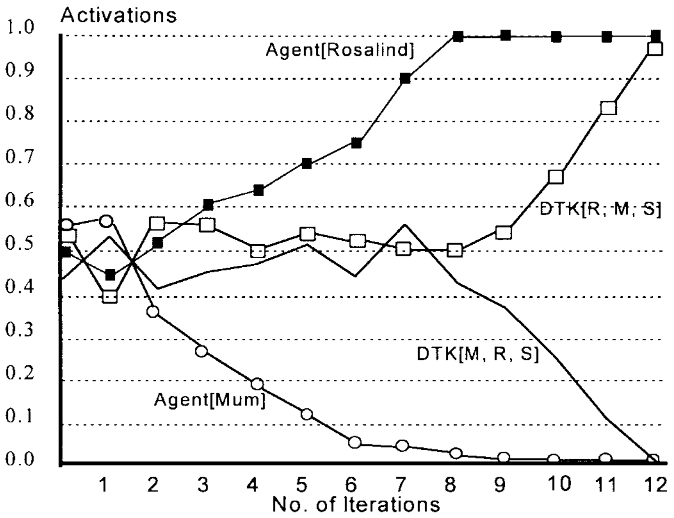


Fig. 13. The competition between *DTK[Rosalind, Mum, Sweater]* and *DTK[Mum, Rosalind, Sweater]* for the pronoun *She* and *her*.

her to be different people in Section V-C and the opposite possibility is not wholly ruled out syntactically. Certainly, these problems should be tackled thoroughly in any full-scale applications. However, in our demonstration system, in order to avoid unnecessary generation of syntactic alternatives as most syntactic systems tend to have, only two of the alternatives are used as an illustration.

VI. Comparison with Other Connectionist and Hybrid Systems

This paper advocates the integration of subsymbolic connectionist and high-level symbolic reasoning models. It describes a framework that encompasses both association and rule-governed inferences. These two are more usefully viewed as complementary in our domain of natural language understanding. In the following, we briefly compare our work with some other connectionist approaches in the natural language understanding literature.

Spreading activation, or marker-passing, is a form of bidirectional search and was first developed and applied to the problem of text understanding by Quillian [31]. In spreading activation, continuous values are propagated through weighted links between nodes [9], [43]. Initial activation is supplied by instantiating some of the nodes and the activation spreads through the weighted links. The activation radiates out in all directions and eventually settles into a stable pattern in which the set of most highly activated nodes constitutes the interpretation of the input. Spreading activation is a search technique to be applied to problems where search cannot be easily directed. The propagation mechanism is extremely simple, but unfortunately, also fragile due to no prior knowledge being incorporated into the systems. Lange [24] describes a structured localized connectionist model ROBIN which explores the integration of language understanding and episodic memory retrieval in a single spreading activation mechanism. ROBIN is capable of using the constraint satisfaction of evidential activation in combination with its parallel dynamic inferencing. However, rules are prewired into the structure of the network, resulting in a semantic network like system. At the same time, ROBIN does not address any syntactic restrictions or role-related expectations in the problem of language disambiguation.

In a related system with very similar capabilities and properties, Ajjanagadde and Shastri [1], Shastri and Ajjanagadde [34], and Shastri *et al.* [35] propose a partial connectionist model which is able to represent a large body of systematic knowledge and to perform forward and backward reasoning on a long term memory network. A prominent mechanism for solving the dynamic variable binding problem consists in using a phased clock and matching rhythmic patterns of activity using temporal synchrony within the network. It is inspired by the human capacity to perform a wide variety of cognitive tasks with extreme ease and efficiency. Dynamic variable binding is represented by argument nodes and their constants, sharing the same phase in any particular period of oscillation during a reasoning process. Shastri has called corresponding inferences reflexive since human reaction is as quick as a reflex. Activation propagates through the network during each phase, implementing the reflexive inferences. Shastri and Ajjanagadde's approach is remarkable in many ways. They offer efficient reasoning in a connectionist knowledge representation system. This representation system has an expressiveness that facilitates the realization of a number of knowledge structures. One of the advantages of Shastri and Ajjanagadde's system [34] is that knowledge can be encoded in parallel. Encoded rules and facts can be fired in parallel provided they are connected to the same predicate nodes when the nodes become active. But most importantly, they attempt to close the gap between high-level reasoning and neural processing and show how reflexive inferences can be drawn efficiently. Although the application of the network in natural language processing is still under exploration, it is an undeniable achievement of this work that it has brought to light a bold new idea for solving the binding problem with processes available in the brain.

Distributed models, where items are represented by bit patterns, have become prominent in the past decade [3], [25], [26], [36]. Due to their tolerance of error and capabilities of representing imprecise concepts, distributed models have attracted a lot of attention in natural language processing. Unfortunately, they suffer a few drawbacks. The most serious one is that they solely rely on a single linguistic clue in their understanding processes. Only a limited source of knowledge, mainly association, can be captured in the systems, and as a result, its disambiguation abilities are highly limited. Generality is another serious problem in most of the networks which are trained only on preparsed, explicitly parenthesized structures, or other knowledge structures such as frames or scripts, whose syntactic structure has already been analyzed. Obviously, they are unable to process nonstereotypical but still comprehensible text. Some of them are discussed as follows.

McClelland and Kawamoto's case-role assignment model provides a good example of how distributed connectionist models have been used to model language understanding. The main task of their model is to learn to assign proper semantic cases roles for sentences by a backpropagation network. Once the network has been trained, it exhibits a certain amount of generalization by mapping the case roles for all new inputs which are similar to the training sentences. However, one of the main limitations in language understanding is that it analyzes input sentences relying on their surface clues in role-expectation. No syntactic or semantic knowledge can be incorporated. Certainly, the ability in disambiguation is in doubt.

XERIC [3] combines a simple recurrent network with a recursive autoassociative memory (RAAM) that encodes and decodes parse trees. RAAM is a three-layer PDP network with auto-association mapping ability. RAAM exhibits a degree of systematicity and productivity within a limited domain [30]. First, the RAAM network is trained to form compressed representations of the syntactic parse tree. Second, a recurrent network is trained to predict the next word in the sequence of words that make up the sentence. The only linguistic clue employed in the recurrent network is the lexical association. Third, a standard three-layer feedforward network is trained to map the hidden-layer of the recurrent network into the RAAM parse tree representation. Unlike the systems discussed so far, XERIC does not extract its knowledge from quasinatural sentences, but from structured parse trees whose constituents are represented, in Lisp fashion, as nested lists. The difficulty with XERIC with respect to connectionist linguistic analysis is that the RAAM network is trained only on preparsed, explicitly parenthesized structures whose syntactic structure has already been analyzed. Thus, the training regime depends upon an external agent. Pollack displays some awareness of this difficulty when he notes, "*The simplifying assumption, that internal representations can first be devised and then used as target patterns, is questionable*."

St. John and McClelland [36] present a connectionist model which learns to assign semantic representations to English-like sentences. The task of the model is to process a single clause sentence into a representation of the event it describes. Although the details of their model are somewhat complex, the overall gist is that, via association, a network is trained to produce a correct semantic representation of the situation described by each input sentence. Situations (or events) described by the input sentences consist of relationships, and the objects involved in those relationships. Although the model demonstrates some capacities in natural language understanding, it is noteworthy that it has several limitations. The most important one is the limitation on the complexity of the sentences, and of the events they describe. In general, it is necessary to characterize the roles and fillers of sentences with respect to their superordinate constituents. It seems extremely difficult for the system to work in complex events in which there may be more than one actor, each performing an action in a different subevent of the overall event or action.

Similarly, Miikkulainen [26] presents a connectionist model, called DISCERN, which reads partial script-based stories and paraphrases them as causally complete output stories using distributed representations. Episodic memory is used to store hierarchical script representations, in which the top level represents the script, such as restaurant, shopping, or travel. The second level contains the traces and specific instantiations in the lower level. Each level receives the facts specific to the story at that level of description. A partial story can cue the hierarchical memory to retrieve the stored trace of the most similar story. Obviously, DISCERN is trained only on preparsed script structure whose syntactic structure has already been analyzed. Whatever disambiguation is necessary for processing the stories is done only by the sentence parser. No explicit disambiguation mechanism has been actually implemented in DISCERN.

Closer to our approach are the hybrid connectionist models of language understanding. Hybrid models aim at combining all the merits of the symbolic approach, localized neural networks and parallel distributed processing. They are designed to integrate low-level mechanisms into higher-level computations. A common approach to hybrid modeling is to explicitly separate the two kinds of information. The semantic knowledge is maintained as *types* while each symbol is created as a *token*, an instance of a type. The parallel distributed semantic networks [27], [37] are examples of this approach. Their approach is to view the localized network as a superstructure to the distributed level. At the macro level, the system is a semantic network, representing knowledge about types and their relations. Higher-level reasoning and meta-operations take place on the semantic network. At the micro level, the nodes of the semantic network are ensembles capable of representing a large number of different instances, and the links consist of weighted connections between ensembles. Low-level operations and statistical learning are performed in this micro level. Although most of the hybrid architectures are able to maintain the parallel inference mechanisms of semantic networks, all of them fail to provide a computational mechanism for gradually accumulating and combining clues from disparate sources which contribute to context, and through cycles of competition, allow the best interpretation of a sentence to appear gradually.

One of the more difficult problems for artificial intelligence research is the problem of modeling commonsense reasoning. A connectionist architecture CONSYDERR [38]–[40] is developed that integrates rule-based reasoning into connectionist networks and couples localized representation with similarity-based distributed representation in two-level architecture. One level of CONSYDERR is the concept level, which contains primitive knowledge statements, or concepts. This level consists of a collection of nodes, or processing elements, for representing the concepts in the domain. The other level is the microfeature level, which contains nodes each of which represents a fine-grained element in the meanings of the concepts represented in the top level. The architecture turns isolated, context-free, and all-or-nothing type rules into an interacting process enmeshed in a network with graded links and activations, which combines pieces of evidence and produces plausible conclusions based on given input regarding particular situations. Because of its massively parallel architecture, CONSYDERR is able to efficiently perform forward chaining reasoning in a parallel fashion. The deduction will be activated spontaneously following the activation of the initial condition. CONSYDERR is developed with the aim of being an integrated model that can deal effectively with a set of important problems in commonsense reasoning, such as rule application, similarity matching. One of the potential limitations in CONSYDERR is the simplicity of the feature representation adopted in any real life applications. Inspired by the work in CONSYDERR, our language understanding system has been extended to have more enriched knowledge representations, such as using proposition matrices to model context effects, and has the ability to deal with various knowledge sources in a uniform way in the problem of semantic resolution.

LeMICON is designed to integrate low-level mechanisms into higher-level computations [4]. First, LeMICON is constructed to address the learning and knowledge acquisition issues in lexicon construction. LeMICON aims at constructing an interpretation graph for each input message. It makes use of the concept of mutual information to construct automatically a two-tiered, graded knowledge base. The knowledge in LeMICON uses on-line text corpora to calculate cooccurrence relationships between a given set of domain concepts and the strength of connection between concepts is represented by a mutual information value. Further, it demonstrates how a constructed interpretation graph can be used to reason about a text by finding the basic events, the connection between them, their relative importance, and to generate a summary of the text. Although LeMICON has very similar goals to our system, there are significant differences between them. First and foremost, despite a thorough proposal in using mutual information as an alternative to hand-coded knowledge engineering efforts, the most important limitation of LeMICON is that it makes no attempt to deal with the issues that involve integrating syntax and semantics, as in a system like TACITUS [16], and suffers accordingly. Under the basic assumption of the design in LeMICON, the nodes in the relational tier are the distilled concepts that have already been preparsed from the input sentences. These nodes are linked by several concept coherence relations without considering their syntactic and semantic relationships. Obviously, while the system has achieved some degree of success, it does not have any capabilities in linguistic disambiguation, such as anaphora resolution, case-role assignment and lexical disambiguation as shown in our work. In contrast to LeMICON, our approach is motivated in dealing with real-world unrestricted text. As shown in the simulations, our system not only parses the input sentences into propositions, it is also capable of compensating for the limitations of syntactic parsing by the knowledge in the system. This is on the basis that a perfect parse from a parser is rarely possible in any real language system. Second, the associational tier in LeMICON encodes the background linguistic knowledge associated with the concepts in the relational tier. This linguistic knowledge is generated solely based on the collocations between the concepts. However, in any natural language system, linguistic knowledge comes from a diversity of sources including word associations, syntactic restrictions, part-of-speech, case-role expectations, semantic context and the information already active in the discourse focus. It is imperative to incorporate all these important sources of information into any language understanding systems.

Compared to the approaches mentioned, the following features make our system unique.

1) Unlike most natural language systems, which depend solely either on syntactic or semantic knowledge, we combine information from a diversity of information sources—mixing word associations, collocations, case-role expectation, and semantic rules—in the semantic resolution process. The word associations and collocation in our subsymbol vectors are strong sources of information that a reader must weigh against other cues since they make immediate and obvious sense selections. Word associations such as *bank/money* create a bias for the related senses. The case-role relationship encoded in our semantic relational network can both influence and be influenced by the selection of word senses because preferences partially constrain the various combinations of a role, its holder, and the filler. Further, in dealing with the syntactic attachment problem which is a direct result of the ambiguity in determining whether a concept is related to an adjacent object, semantic rules are incorporated by examining all combinations of senses and attachments to locate the semantically best one in the semantic resolution. For example, a lexical preference rule, as the prior knowledge in the semantic relational network, specifies that the preference for a particular attachment depends on how strongly or weakly the verb of the clause prefers its possible arguments. Our system describes an integrated approach to text understanding.
2) Our architecture captures the information which has already been activated in the discourse focus dynamically. The proposition matrices prescribe an expedient mechanism for combining sentential knowledge and can be carried over from the preceding propositions into the current processing cycle. Each distilled proposition matrix can be regarded as a knowledge source which unveils some of the interrelations between the concepts. In fact, this mechanism provides a means to the inclusion of context knowledge by superimposing the proposition matrices on the current knowledge matrix which has the same morphologic structure.
3) Most of the previous language understanding systems assume a front-end that produces a restricted representation, or even a knowledge structure, of the natural language input. As a result, syntactic and semantic cues are unable to be used in their inference processes. Our system starts off from unrestricted texts which are parsed into propositions. A proposition preserves the essential information of a sentence while disregarding details such as word order. Although some difficulties are encountered in parsing sentences into their corresponding propositions in our simulations, it is not necessary to have a perfectly parsed proposition before the disambiguation. As remarked and exemplified earlier, the system has the ability to compensate for the limitations of the parsing as shown in one of the simulations.

4) Whenever an ambiguous input is encountered, our system generates a set of assertions describing lexical items, syntactic and semantic relations from propositions. The subsymbol vectors form associations from one lexical unit to other lexical units which it may link with. When an associative network is fully deployed, every node in the network is connected with weighted links. Obviously, unlike most localized architectures, weights of our associative network are not hand-crafted. They are bound by the prior knowledge encoded in the semantic relational network and the similarity measure of the subsymbol vectors. Knowledge of the foregoing sentences also supplies further constraints to the formation of associative matrices.

In summary, unlike all previous systems mentioned above, our approach demonstrates how semantic meanings can be resolved by using both linguistic rules and semantic associations and how context effects can be modeled and carried over into the sentence analysis. Our model combines all sources of information about the meaning of a sentence in a uniform manner. A single constraint satisfaction mechanism takes the clues from disparate sources, and through cycles of competition, allows the best interpretation of a sentence to appear gradually. The seamless integration from knowledge in both paradigms has shown more competent, compared with the previous systems, in incorporating various knowledge sources to semantic disambiguation.

VII. Conclusions

We have proposed and implemented a neurosymbolic model for language processing and described its applications to the modeling of context effects in natural language disambiguation. The model regards the problem of language disambiguation as a resolution of collective evidences that emerge from symbolic inferences at a higher level and semantic associations at a low level in human cognition. The proposed architecture provides a simple framework which is capable of incorporating a wide variety of linguistic information in a uniform way. Although some attempts in language understanding have been made in the neural network community, our system advances the state of the art in two important aspects.

- Having access to a large amount of information and being able to use it effectively are major features in our system. In fact, they are extremely important in understanding unrestricted texts. In this article, we tried to convince the reader that it is possible to develop a hybrid system with an explicit relationship between background knowledge, context effect and the input texts. Our approach to natural language disambiguation uses information drawn from the input proposition, combining the strongest, most obvious sense preference created by lexical association, case-role expectation, encoded semantic rules, and previously analyzed propositions.
- The symbolic structure of our model is capable of providing most of the main features of any rule-based systems. At the same time, the connectionist stratum in our model provides a solution of simplification of computational complexity as well as remedying the brittleness problem found in typical symbolic systems. This synergy helps to deal with problems, such as in natural language understanding, which are characterized by almost infinite variability. Taking into account the merits from these two different strata, our hybrid system is capable of tackling the language disambiguation problem, and is potentially more powerful than systems relying on only one processing paradigm.

There are some encouraging results from applying the system to language disambiguation. Clearly, it is too ambitious to claim that our system can handle all the problems of ambiguity in language understanding and the scheme we have proposed certainly needs further refinement. Additional experimental work is certainly required; nevertheless, the simulations have presented its capabilities in natural language disambiguation.

Acknowledgment

The authors would like to thank B. T'sou of the Language Information Sciences Research Centre, City University of Hong Kong, for many helpful discussions. The authors are also grateful to two of the anonymous reviewers for their valuable and constructive comments which really improved the paper from its previous versions.

References

[1] V. Ajjanagadde and L. Shastri, "Rules and variables in neural nets," *Neural Computation,* vol. 3, pp. 121–134, 1991.

[2] J. R. Anderson, *Language, Memory, and Thought.* Lawrence Erlbaum Associates, 1976.

[3] G. Berg, "A connectionist parser with recursive sentence structure and lexical disambiguation," in *Proc. 10th National Conf. Artificial Intelligence,* San Jose, CA, 1992, pp. 32–37.

[4] L. A. Bookman, "A scaleable architecture for integrating associative and semantic memory," *Connection Science,* vol. 5, no. 3/4, pp. 243–273, 1993.

[5] ——, *Trajectories through Knowledge Space: A Dynamic Framework for Machine Comprehension.* Kluwer, 1994.

[6] S. W. K. Chan and J. Franklin, "Symbolic connectionism in tiers: A strategy of discourse comprehension," in *Artificial Intelligence—Sowing the Seeds for the Future,* C. Zhang, J. Debenham, and D. Lukose, Eds. World Scientific, 1994, pp. 434–441.

[7] S. W. K. Chan, "Inferences in natural language understanding," in *Proc. 4th IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE/IFES'95),* Yokohama, Japan, 1995, vol. 2, pp. 935–940.

[8] S. W. K. Chan and J. Franklin, "A neurosymbolic integrated model for semantic ambiguation resolution," in *Proc. IEEE Int. Conf. Neural Networks (ICNN'95),* Perth, Australia, 1995, pp. 2965–2970.

[9] G. W. Cottrell and S. L. Small, "A connectionist scheme for modeling word sense disambiguation," *Cognition and Brain Theory,* vol. 6, pp. 89–120, 1983.

[10] M. G. Dyer, M. Flowers, and Y. J. A. Wang, "Distributed symbol discovery through symbol recirculation: Toward natural language processing in a distributed connectionist network," in *Connectionist Approaches to Natural Language Processing,* R. G. Reilly and N. E. Sharkey, Eds. Lawrence Erlbaum Associates, 1992, pp. 21–48.

[11] J. L. Elman, "Distributed representations, simple recurrent networks, and grammatical structure," *Machine Learning,* vol. 7, pp. 195–225, 1991.

[12] C. J. Fillmore, "The case for case," in *Universals in Linguistic Theory,* E. Bach and R. T. Harms, Eds. Holt, Rinehart and Winston, 1968, pp. 1–90.

[13] J. Franklin, "How a neural net grows symbols," in *Proc. 7th Australian Conf. Neural Networks (ACNN'96),* Australian National University, 1996, pp. 91–96.

[14] S. J. Hanson, "Conceptual clustering and categorization: Bridging the gap between induction and causal models," in *Machine Learning: An*

Artificial Intelligence Approach, Y. Kodratoff and R. Michalski, Eds. 1990, vol. III, pp. 235–268.

[15] S. Harnad, "The symbol grounding problem," *Physica D,* vol. 42, pp. 335–346, 1990.

[16] J. R. Hobbs, M. E. Stickel, D. E. Appelt, and P. Martin, "Interpretation as abduction," *Artificial Intelligence,* vol. 63, pp. 69–142, 1993.

[17] K. Jensen and J. Binot, "Disambiguating prepositional phrase attachments by using on-line dictionary definitions," *Computational Linguistics,* vol. 13, no. 3/4, pp. 251–260, 1988.

[18] P. N. Johnson-Laird, *Mental Models.* Harvard University Press, 1983.

[19] W. Kintsch, *The Representation of Meaning in Memory.* Lawrence Erlbaum Associates, 1974.

[20] W. Kintsch and T. A. van Dijk, "Toward a model of text comprehension and production," *Psychological Review,* vol. 85, no. 5, pp. 363–394, 1978.

[21] B. Kosko, "Fuzzy cognitive maps," *Int. J. Man-Machine Studies,* vol. 24, pp. 65–75, 1986.

[22] G. Lakoff, "A suggestion for a linguistics with connectionist foundations," in *Proc. 1988 Connectionist Models Summer School,* D. S. Touretzky, G. E. Hinton, and T. J. Sejnowski, Eds. Morgan Kaufmann, 1988, pp. 301–314.

[23] R. Langacker, *Foundations of Cognitive Grammar I: Theoretical Prerequisites.* Stanford University Press, 1987.

[24] T. E. Lange, "A structured connectionist approach to inferencing and retrieval," in *Computational Architectures Integrating Neural and Symbolic Processes: A Perspective on the State of the Art,* R. Sun and L. A. Bookman, Eds. Kluwer Academic, 1995, pp. 69–115.

[25] J. L. McClelland and A. H. Kawamoto, "Mechanisms of sentence processing: Assigning roles to constituents of sentences," in *Parallel Distributed Processing,* D. E. Rumelhart and J. L. McClelland, Eds., 1986, vol. 2, pp. 272–325.

[26] R. Miikkulainen, *Subsymbolic Natural Language Processing: An Integrated Model of Scripts, Lexicon, and Memory.* MIT Press, 1993.

[27] V. L. Nenov and M. G. Dyer, "Perceptually grounded language learning. 2. DETE: a neural/procedural model," *Connection Science,* vol. 6, no. 1, pp. 3–41, 1994.

[28] I. P. Pavlov, *Selected Works of I. P. Pavlov,* Trans. from the Russian by S. Belsky, Foreign Languages Publishing House, 1955.

[29] S. Pinker and A. Prince, "On language and connectionism: Analysis of a parallel distributed processing model of language acquisition," *Cognition,* vol. 28, pp. 73–193, 1988.

[30] J. B. Pollack, "Recursive distributed representations," *Artificial Intelligence,* vol. 46, pp. 77–105, 1990.

[31] M. R. Quillian, "The teachable language comprehender: A simulation program and theory of language," *Communications of the ACM,* vol. 12, no. 8, pp. 459–476, 1969.

[32] G. Razran, *Mind in Evolution.* Houghton Mifflin, 1971.

[33] N. Sharkey, *Connectionist Natural Language Processing.* Boston, MA: Kluwer Academic, 1992.

[34] L. Shastri and V. Ajjanagadde, "From simple associations to systematic reasoning: A connectionist representation of rules, variables, and dynamic bindings using temporal synchrony," *Behavioral and Brain Sciences,* vol. 16, no. 4, pp. 417–494, 1993.

[35] L. Shastri, V. Ajjanagadde, L. Bonatti, T. E. Lange, and M. G. Dyer, "From simple associations to systematic reasoning: A connectionist representation of rules, variables, and dynamic bindings using temporal synchrony (comments and responses)," *Behavioral and Brain Sciences,* vol. 19, no. 2, pp. 326–337, 1996.

[36] M. F. St. John and J. L. McClelland, "Learning and applying contextual constraints in sentence comprehension," *Artificial Intelligence,* vol. 46, pp. 217–257, 1990.

[37] R. A. Sumida and M. G. Dyer, "Propagation filters in PDS networks for sequencing and ambiguity resolution," in *Advances in Neural Information Processing Systems 4,* J. E. Moody, S. J. Hanson, and R. P. Lippmann, Eds. Morgan Kaufmann, 1992, pp. 233–240.

[38] R. Sun, *Integrating Rules and Connectionism for Robust Commonsense Reasoning.* New York: Wiley, 1994.

[39] ——, "Structuring knowledge in vague domains," *IEEE Trans. Knowledge Data Eng.,* vol. 7, no. 1, pp. 120–136, 1995.

[40] ——, "Commonsense reasoning with rules, cases, and connectionist models—A paradigmatic comparison," *Fuzzy Sets and Systems,* vol. 82, no. 2, pp. 187–200, 1996.

[41] R. Taraban and J. L. McClelland, "Constituent attachment and thematic role assignment in sentence processing: Influences of content-based expectation," *J. Memory and Language,* vol. 27, pp. 597–632, 1988.

[42] T. A. van Dijk and W. Kintsch, *Strategies of Discourse Comprehension.* New York: Academic, 1983.

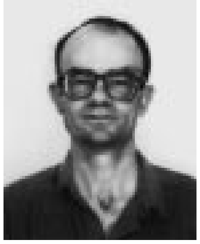
[43] D. L. Waltz and J. B. Pollack, "Massively parallel parsing: A strongly interactive model of natural language interpretation," *Cognitive Science,* vol. 9, pp. 51–74, 1985.

[44] G. Whittemore, K. Ferrara, and H. Brunner, "Empirical study of predictive powers of simple attachment schemes for postmodifier prepositional phrases," in *Proc. 28th Annu. Meet. Assoc. Comput. Linguistics,* 1990, pp. 23–30.

Samuel W. K. Chan received the M.Sc. degree in 1986 from the University of Manchester, U.K., the M.Phil. degree in 1991 from the Chinese University of Hong Kong, and the Ph.D. degree in 1998 from the University of New South Wales, Australia, all in computer science.

He is currently a Lecturer in the Department of Computer Science and an affiliate member of the Language Information Sciences Research Center of City University of Hong Kong. His interests include natural language processing and computational intelligence.

Dr. Chan received the outstanding student paper award at the 1996 IEEE International Conference on Systems, Man and Cybernetics, Beijing, China.



James Franklin received the M.A. degree from Sydney University, Australia, in 1977 and the Ph.D. degree in algebra from Warwick University, U.K., in 1981.

He is a Senior Lecturer in mathematics at the University of New South Wales, Australia. He is the principal author of *Introduction to Proofs in Mathematics* (Englewood Cliffs, NJ: Prentice Hall, 1988). His research interests include the implementation of clustering algorithms in neural nets and the history of probability.