

# Sentence Understanding and Learning of New Words with Large-Scale Neural Networks

Heiner Markert, Zöhre Kara Kayikci, and Günther Palm

University of Ulm  
Institute of Neural Information Processing  
89069 Ulm, Germany  
{heiner.markert, zoehre.kara, guenther.palm}@uni-ulm.de  
<http://www.informatik.uni-ulm.de/neuro/>

**Abstract.** We have implemented a speech command system which can understand simple command sentences like “Bot lift ball” or “Bot go table” using hidden Markov models (HMMs) and associative memories with sparse distributed representations. The system is composed of three modules: (1) A set of HMMs is used on phoneme level to get a phonetic transcription of the spoken sentence, (2) a network of associative memories is used to determine the word belonging to the phonetic transcription and (3) a neural network is used on the sentence level to determine the meaning of the sentence. The system is also able to learn new object words during performance.

**Keywords:** Associative Memories, Hidden Markov Models, Hebbian Learning, Speech Recognition, Language Understanding.

## 1 Introduction

A variety of speech recognition systems are currently in use in applications such as command & control, data entry, and document preparation. In this study, we have applied neural associative memories to a speech processing system in a hybrid approach with hidden Markov models (HMMs) [1][2]. The system is able to recognize spoken command sentences such as “Bot show plum” or “Bot pick blue plum”. Those sentences are processed in three stages: At the first stage, the auditory signal is transformed into a sequence of corresponding triphones via a HMM based triphone recognizer. At the second stage, the generated stream of triphones is forwarded to a word recognition module, which consists of a number of binary heteroassociative memories and is able to recognize single words. The module determines the best matching words from the triphone data. As last processing step, the words are forwarded to a sentence recognition module which parses the sentence with respect to a given grammar. After successful processing, each word in the sentence is labeled with its grammatical role, giving a relatively straightforward representation of the meaning of the sentence.

The system is able to learn new object words. Learning is triggered by the special phrase “This is” followed by a novel word, e.g. “This is lemon”. After learning, the

system is able to understand this new object word like any other object and can successfully process sentences like “Bot show lemon” or “Bot pick yellow lemon”.

## 2 Neural Associative Memories

A neural associative memory (NAM) is a single layer neural network which maps a set of input patterns to a set of corresponding output patterns. The patterns are binary and sparsely coded. A NAM can be represented as a synaptic connectivity matrix. Pairs of patterns are stored in the binary matrix using clipped binary Hebbian learning [3][4]. In case of autoassociation, input and output patterns are identical. Autoassociation allows for pattern completion, fault tolerance and long-term activation of patterns via feedback links. Heteroassociation is used to map one pattern onto another (e.g. to map a pattern “X” to a representation “Y”). They are also often used to translate between different neural coding schemes of different populations.

We have decided to use Willshaws model of associative memory [5][6] as the basic model. Different retrieval strategies are employed in different parts of the system. One of these strategies is one step retrieval with threshold [4][7], where the threshold is set to a global value:

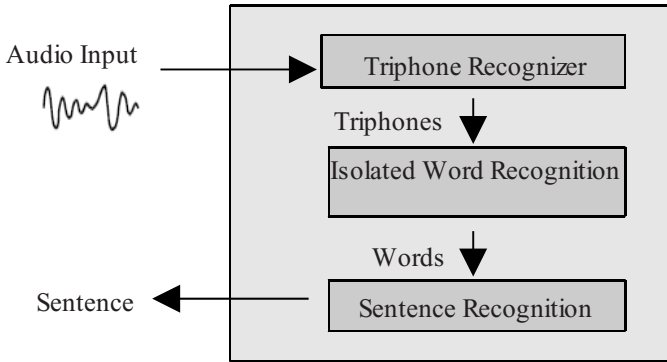
$$Y_j^k = 1 \Leftrightarrow (X^k W)_j \geq \Theta, \quad (1)$$

where  $\Theta$  is the global threshold,  $X$  is the input,  $Y$  is the output and  $W$  is the matrix of synaptic strengths. A special case of this strategy is the Willshaw strategy, where the threshold is set to the number of ones in the binary input vector  $X$ .

In more complex memories we also use the so-called spike counter model [8]. With this model, the network is simulated in global steps, where in each global step a complete retrieval is calculated by each of the associative memories. In contrast to Willshaws model [5][6] which interprets incoming activation as membrane potential values, incoming activation levels per neuron are interpreted as value of the membrane potentials derivative in the spike counter model. Thus, if a neuron receives strong input, its membrane potential is rising fast, while with low input, the membrane potential is increasing only slowly. In this simple model, the membrane potential is modified linearly. The neurons spike whenever the membrane potential reaches a given threshold and the membrane potential is reset to zero after the spike is emitted. In each retrieval, every neuron is allowed to spike at most twice and the retrieval is stopped when no more neuron is able to spike (which happens either when each neuron has spiked twice or there is no more neuron left that receives any positive postsynaptic input). The spike patterns are then forwarded through the heteroassociative connections and the next global time step starts.

## 3 The System

The system is composed of a chain of three modules. The first module is a triphone recognizer based on hidden Markov models, which is responsible for generating a sequence of triphones corresponding to the spoken command sentence. The second module is an isolated word recognizer from which single words are retrieved. The last



**Fig. 1.** Overview of the language processing system. The three main components of the system are connected in a feed-forward manner. Besides the triphone recognizer, all boxes are implemented as neural networks. Communication between the boxes is based on neural representations that allow ambiguities to be forwarded between the different units of the system. In case the word recognizer cannot decide between “ball” and “wall”, a superposition of both words is sent to the sentence recognizer which can then use additional contextual input to resolve the ambiguity (see examples in text below).

module in the system is the sentence recognition network which analyzes the retrieved words with respect to simple grammar rules. Fig. 1 shows a block diagram of the system.

The triphone recognizer module, based on hidden Markov models, receives audio input via a microphone and converts it to a stream of triphones. This stream is then transferred to the isolated word recognizer module which extracts words from the stream of triphones. Afterwards, the output of the word recognizer is sent to the sentence recognition module which parses the stream of words against a given grammar and assigns grammatical roles to the words. The last two modules are networks of neural associative memories.

In the following, the three parts are described in more detail.

### 3.1 Triphone Recognizer

**Speech Material.** To train triphone HMMs, the training data set of TIMIT continuous speech corpus without the “SA”-type sentences has been used [9]. TIMIT is manually labeled and includes time-aligned, manually verified phonetic and word segmentations. The original set of 61 phonemes was reduced to a set of 39 phonemes in this study. The TIMIT training data consists of 462 speakers and 1716 sentences.

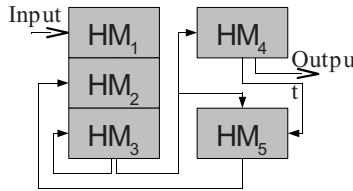
We added our own recordings to the training material in order to adapt the system to our scenario. A set of 105 different sentences with a vocabulary of 43 words has been recorded from four different speakers of our institute. From this data, 70 sentences were used for training and the remaining 35 sentences were used for testing. For training and testing the same speakers were used. The total number of words in the test set of our own data is 504, the training set consists of 1068 words in total.

**Hidden Markov Models and Language Models.** The HMMs used in the triphone recognizer utilize three-state continuous 8-Gaussian triphone models, i.e, a set of 7421 triphones that are seen in the training set of TIMIT speech corpus and in our own speech data consisting of 105 sentences. Word internal models are used to determine the word boundaries in the sentence. The models are trained using the decision tree based state tying procedure using HTK [2]. The triphone models are trained with TIMIT speech corpus. To adapt the HMMs to our scenario, the models are also trained with our own training data.

To recognize the triphone sequence for a given sentence, a bigram language model on triphone level is also used, which is created with respect to the triphone transcriptions of the sentences in the training set of TIMIT speech corpus and our speech data.

### 3.2 Isolated Word Recognition

The isolated word recognition module consists of five heteroassociative memories. Fig. 2 shows the general structure of the module. Each box denotes a heteroassociative memory, the arrows correspond to auto- or heteroassociative connections. Due to the large number of triphones, we decided to use diphones as basic processing units in the neural network for isolated word recognition. This requires a translation of the triphones from the HMMs into diphones before they are processed by the word recognition network. The total number of diphones used in the memories is 1600.



**Fig. 2.** Isolated word recognition module. It consists of five heteroassociative memories (depicted as gray boxes) that are connected by auto- or heteroassociations respectively (arrows). The input diphones extracted from the HMMs is sent to  $HM_1$ , while the output words are taken from  $HM_4$  after a complete word has been processed (see text for details).

The word boundaries are determined with respect to diphones in the input sequence: The HMMs detect diphones due to a language model which knows about common di- and triphones. In our experiments word boundaries were recognized quite stable using this approach.

The tasks of the heteroassociative memories  $HM_1$  through  $HM_5$  are described in more detail in the following:

$HM_1$  is a matrix of dimension  $L \times n$ , where  $L$  is the length of the input code vector and  $n$  is the number of diphones, ( $n=1600$  in our system). The memory receives input diphones from the HMMs and presents them to the word recognition network as 1 out of  $n$  code.

$HM_2$  has the same structure as  $HM_1$ , but receives diphones expected as input in the next step (see  $HM_5$  below).

$HM_3$  is a matrix of dimension  $n \times n$ . It stores diphone transitions of the words in the vocabulary.

The memories  $HM_1$ ,  $HM_2$  and  $HM_3$  can be regarded as one combined memory  $HM_*$ . In each retrieval step,  $HM_1$  represents the diphones from the HMMs, while  $HM_2$  and  $HM_3$  represent the diphones predicted by the network from the previous input. The outputs of the three memories are summed up such that the influence of  $HM_2$  and  $HM_3$  is reduced compared to that of  $HM_1$ , giving the acoustic input a higher priority. After summation, a global threshold is applied.

The network generates a list of already heard diphones and presents it to memory  $HM_4$ . Meanwhile, the currently heard diphone(s) are presented to memory  $HM_5$ .

$HM_4$  is a matrix of dimension  $n \times M$ , where  $M$  is the number of output units, ( $M=200$ ). The memory is used to store all words known to the system and to activate those that match the current list of diphones, generating a list of word hypotheses. During retrieval, the activated words are forwarded to memory  $HM_5$ . The output pattern is a randomly generated 5 out of 200 code vector.

$HM_5$  is used to predict the diphones expected in the next step. It takes the word hypotheses from  $HM_4$  and the currently heard diphone into account and tries to predict which diphones are to be expected in the next step.

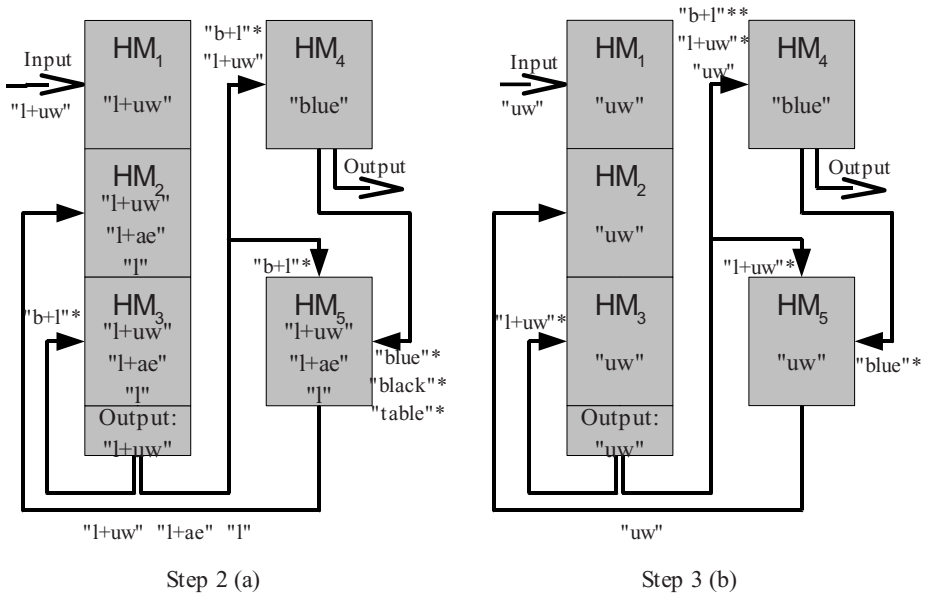
$HM_5$  is organized in columns. Each column belongs to one specific diphone. For each word containing that diphone, the corresponding column stores a heteroassociation from the word representation (input) to the diphone following the column-specific diphone in that word (output). By exciting only those columns that belong to diphones matching the current output of memory  $HM_*$ ,  $HM_5$  generates a prediction which diphones are about to enter the network in the next time step.

In praxis, the input to  $HM_5$  is not a single word but a superposition of possible words generated in  $HM_4$ , thus, the output is usually not unique but also a superposition of possible diphones. Furthermore it is possible that the same diphone occurs twice in a word and thus the output can be ambiguous even if the memory is addressed with a single word.

**Retrieval.** When input is given to the network as a sequence of triphones from the HMMs, e.g. “sil b+ow b-ow+t ow-t sh+ow sh-ow b+l b-l+uw l-uw p+l p-l+ah l-ah+m ah-m sil” for the spoken sentence “bot show blue plum”, the first step is to divide it into subsequences for isolated words with respect to diphones, e.g. “sil b+ow b-ow+t ow-t / sh+ow sh-ow ...”, where “/” denotes the word boundaries. Afterwards, the subsequence of triphones is decomposed into diphones, because the following parts of the network are processing diphones to limit the use of memory resources.

For the subsequence “b+l l+uw uw”, firstly the diphone “b+l” enters the network. For the first diphone of each word, the memories  $HM_2$  and  $HM_3$  are not activated and the output of  $HM_*$  is the input diphone itself. The resulting diphone is then forwarded to  $HM_4$  to generate a superposition of the words which contain the diphone “b+l”. For the first diphone “b+l”, the words “blue”, “black” and “table” are activated simultaneously. The output words from  $HM_4$  are then sent to  $HM_5$  to predict the next possible diphones.

Fig. 3 illustrates the module when processing the last two diphones. In the next step,  $HM_1$  receives the second diphone “l+uw”,  $HM_5$  predicts the next diphones and



**Fig. 3.** The isolated word recognition module when processing (a) the second diphone “l+uw” and (b) the last phoneme “uw”. The text on the arrows denotes the input that the areas receive via the corresponding connection. \* means that the input is from the previous step (delay), \*\* means delay by two time steps.

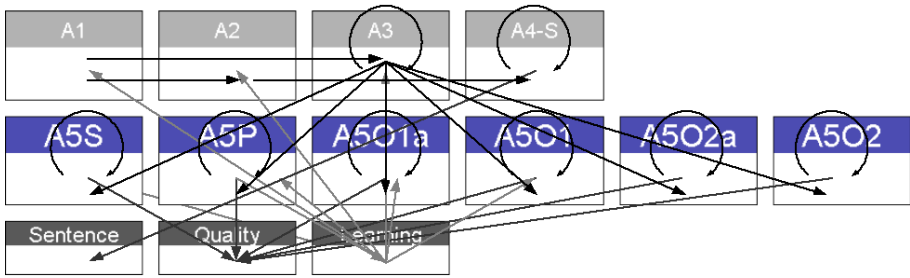
forwards them to  $HM_2$ .  $HM_3$  takes the output diphones of  $HM_*$  in the previous step as input. The resulting diphone “l+uw” in  $HM_*$  is added to the diphone in the first retrieval step and forwarded to  $HM_4$ . In  $HM_4$ , the word “blue” is activated. In the following step, the last phoneme “uw” is processed by the network in the same way.

After retrieving a complete word, the module is reset to its initial state in order to start with the next word in the input stream. In particular, to “reset” the module means to delete the list of already processed diphones that is fed into  $HM_4$ . After successfully retrieving a word it is forwarded to the sentence recognition system.

The experimental result on our test set (see Section 3.1) shows that our system slightly outperforms the adapted HMMs: the system recognized 98% of the words (2% word level error), while a set of adapted HMMs recognized 96% of the words correctly. Due to the fault-tolerance property of associative memories, the system is able to deal with spurious inputs, such as incorrectly recognized subword units. Thus, the correct word can be retrieved in spite of possible a corresponding noisy or incomplete subword-unit transcriptions to a certain degree.

### 3.3 Sentence Recognition

The sentence recognition network parses the stream of words detected by the isolated word recognizer with respect to a certain grammar. Fig. 4 gives a short overview of this module.



**Fig. 4.** The sentence recognition network. Each box corresponds to one autoassociative memory, arrows denote heteroassociations. Boxes with circular arrows use the autoassociative short-term memory (e.g. they stabilize patterns in time).

The module is composed of several autoassociative memories (depicted as boxes in Fig. 4) that are connected by heteroassociations (arrows in Fig. 4). Heteroassociations are used to exchange activation between the associative memories and to translate between the different neural representations used in different autoassociative memories. After a sentence is successfully processed, the sentence module has determined the type of sentence (e.g. Subject-predicate-object, SPO) and assigned to each word its grammatical role.

The memories in the sentence module use the so called “spike counter model” of associative memory [10]. Each memory stores a set of different patterns (assemblies). To allow for easy display of the network state, each pattern has an associated name stored in a simple look-up table. Whenever a network activity is displayed or a name of a pattern is mentioned, we really mean that the underlying group of neurons is activated.

The different memories in the module serve the following purposes:

A1 is the input memory. Activation from the isolated word recognizer arrives here. This memory is only required to allow for developing and testing the module individually. A1 holds all words known to the system in the 5 out of 200-code from the isolated word recognition module.

A2 and A3 distinguish between semantical elements (words, A3) and syntactical elements like word boundary signals and sentence end markers (arbitrary patterns, A2) respectively.

A4-S is a sequence memory. Sequences are realized by an additional hetero-association from the memory onto itself that stores the state transitions of the sequence elements. The heteroassociation is delayed and weaker than the auto-association, so normally an active pattern is stable. With short inhibition, the auto-association can be inhibited and the heteroassociation, which is still effective because of its higher delay, will activate the next pattern in the sequence.

A4-S holds the grammatical information, i.e. the sentence types that are known to the system. It stores sequences like S->P->O->OK\_SPO or S->P->A->O->A->O, where S means subject, P means predicate, A adjective and O object.

A5S, A5P, A5O1a, A5O1, A5O2a, A5O2: These memories hold the subject, predicate, attribute to first object, first object, attribute to second object and second object of the sentence, respectively. They are subsequently filled while the sentence is

parsed against the grammar. Because these memories use a short term mechanism, the words assigned to them are active for a longer period of time, allowing to access the information later, e.g. for action planning.

The other fields (Sentence, Quality, Learning) are additional status information fields that can distinguish between only two states. The Sentence-memory activates a “complete”-state when a complete sentence was correctly parsed, the Quality-memory activates a “good”-pattern when there were no ambiguities on single word level (see below) and the Learning-box activates a special learn signal when a “this is”-sentence is recognized.

## 4 Disambiguation

The system is able to deal with ambiguities on the single word level. When a word was not uniquely understood by the HMM (e.g. an ambiguity between “ball” and “wall”), the isolated word recognition module is not able to assign a unique interpretation to the stream of triphones generated by the HMMs. In this case, a superposition of several words that all match the stream closely is activated and forwarded to the sentence recognition network. The sentence recognizer is able to deal with that ambiguity and keeps it active until further context information can be used to resolve the ambiguity. The ambiguous sentence “bot lift bwall”, where “bwall” is an ambiguity between ball and wall, can be resolved to “bot lift ball” in the sentence processing network because a wall is not a liftable object. To achieve this disambiguation a bidirectional link between A5P and A5O1 is used, which supports matching pairs of predicates and objects.

## 5 Online Learning

New object words can be learned by the system during performance without further training the triphone models or changing the structure of the system. The online learning performance highly depends on the performance of HMMs that need to be trained with enough speech data and have a comprehensive language model in order to enable the HMMs to generate a plausible phonetic representation for novel words. Learning of a new word is initiated by a sentence of the type “This is cup”, where “cup” is the word that has to be learned. “This is” arouses the system to learn a new object word.

While learning of a novel word, in the isolated word recognition module,  $HM_1$  and  $HM_2$  are not updated, whereas  $HM_3$  is updated according to the sequence of diphones of the novel word. To store the new object word in  $HM_4$  and  $HM_5$ , a new word vector is randomly generated and stored in the associative memories.

The representation of the novel word is then forwarded to the sentence recognition network. The associative memories used here employ the so called “spike counter model” [10] which allows for automatic detection of novel address patterns. The modules are able to measure the retrieval quality, where a low quality indicates that the address pattern did not match any of the stored patterns closely. If this is the case and an additional learn signal is applied to the memory, it generates a new pattern and stores it autoassociatively. If however the address pattern matches a stored pattern



closely, no new pattern is generated even if a learn signal is applied. This ensures that already known stimuli will be recognized. After learning a new pattern, the heteroassociations between the concerned memories are updated according to a simple binary Hebbian learning rule.

In the special case of learning new object words, the sentence recognition network uses the phrase “This is” to determine that a new word is probably going to be learned. If “This is” was heard, the Learning-memory (see Fig. 4) activates a special pattern responsible for learning new object words. If this pattern is active, the neurons emit a special learn signal to all memories in the sentence module that are concerned with object words.

After learning a new object word, the system is able to deal with it in exactly the same way as it deals with the vocabulary it knows from initialization. If e.g. “cup” was learned during performance by the command “this is cup”, the system can afterwards understand sentences like “bot show cup”, “bot lift cup” etc.

At the current state of our work, in order to demonstrate the online learning capability of the system, we initially stored 40 out of 43 words in the system and the remaining 3 words were used for learning during performance. This is due to the fact that we did not yet record enough speech material for our own speakers to train the HMMs such that they can generate plausible phonetic representation for arbitrary new words.

## 6 Discussion

We have presented a system that is able to understand spoken language and to transform low level subsymbolic auditory data into corresponding symbolic representations that allow for easy deduction of the meaning of the sentence.

The system is composed of three rather independent main components, the HMM triphone recognizer, the isolated word recognizer and the grammar parser (also called sentence recognizer). Currently, this three components are interconnected in a feed forward manner due to the early stage of development of especially the isolated word recognizer. In particular a feedback connection from the grammar parser to the isolated word recognizer would be beneficial as a connection would allow the word recognizer to focus on words that fit for example grammatical or semantical constraints arising while the current sentence is parsed. Although we did not yet implement this kind of connection, we plan to add feedback connections in the near future.

The system is expected to scale well with the size of the vocabulary. It is well known that associative memories store information efficiently in terms of storage capacity [6][7].

The word recognition network additionally allows for different coding schemes and subword units to be used. In particular, the latter can be either phonemes, context dependent phonemes such as diphones or triphones, semi-syllables or syllables. The dimensionality of the memories will then scale with respect to the number of subword units. For the vocabularies of small size, it is convenient to work with phonemes or diphones in heteroassociative memories instead of triphones (e.g. for a given set of 40 phonemes, the corresponding set of diphones is composed of up to 1600 diphones

whereas the total number of corresponding triphones is around 10000). On the other hand, for large vocabularies it is more convenient to use triphones or semi-syllables.

The sentence understanding network can operate using random k-out-of-n codes for word representation and the dimensionality of the memories can be chosen such that high storage capacity can be reached for a given vocabulary. This means in particular, that memory requirements grow less than linearly with the number of words to be stored. For increased, more complex grammars, population A4-G has to be scaled. Almost the same arguments apply here. Due to the efficient tree-like structure of the grammatical representation, the network scales rather well with respect to a more complex grammar also. The computational complexity in terms of computing time scales basically linearly with the number of neurons in the system, as long as the overall spike activity does not increase significantly (i.e., if the number of active one entries in the code vectors is held constant). As additional sparseness also increases the storage capacity, the above constraint is however fulfilled in practical systems. Note further that the number of neurons does increase less than linearly with the size of the vocabulary, this is also true for the computational time required.

To show the correct semantical understanding of parsed sentences by the system, the model is embedded into a robot [11]. Therefore, the system is extended by a neural action planning part, some simple motor programs (e.g. to pan and tilt the camera) and a RBF-based object recognition system. The scenario is a robot standing in front of a white table with fruits lying on it, and the robot has to perform actions corresponding to spoken commands. We can demonstrate that the robot is able to perform simple actions like e.g. showing a plum by centering it in its camera image after a corresponding command like “bot show plum” was given.

In the robotics scenario described above, the system is not only able to learn new object words but also to bind the words to visual representations (this happens in the action planning network) and also to learn new visual object representations and bind them to object words (new or already known objects). This allows e.g. to generate synonyms by introducing new object words and to bind them to an already known visual object.

If we compare the system represented in this study with a HMM based word recognition system in terms of learning of new words, HMMs have to make some modifications to the pronouncing dictionary and task grammar at each time to add a new word. In contrast, the presented system is capable of learning new word representations by simply adding new patterns to associative memories; it does not need structural and time-consuming modifications as long as the associative memories are not overloaded.

Our approach to language understanding should be understood in the context of a larger model that integrates language understanding, visual object recognition, and action planning in a functional, large-scale neural network in a robotics scenario. Parts of the presented work have been developed earlier within the MirrorBot project of the European Union<sup>1</sup>, and results on sentence understanding [10] and the robotics application [11] have been published earlier. Most other approaches deal only with one of the aspects mentioned above at a time. Closely related to our work are the approaches of Arbib [12], Roy [13], Kirchmar and Edelman [14] and of Billard and Hayes [15].

---

<sup>1</sup> See <http://www.his.sunderland.ac.uk/mirrorbot/>

However, to our knowledge this is the first robot control architecture including simple language understanding, visual object recognition and action planning, that is realized completely by neural networks [11] and that is able to resolve ambiguities and to learn new words during performance [10]. It also represents the first real-time functional simulation of populations of spiking neurons in more than ten cortex areas in cooperation.

## References

1. Rabiner, L., Juang, B.H.: *Fundamentals of Speech Recognition*. Prentice-Hall, Inc, Englewood Cliffs (1993)
2. Young, S., et al.: *The HTK Book (for HTK Version 3.2.1)*. Cambridge University Engineering Department, Cambridge (2002)
3. Hebb, D.O.: *The Organization of Behaviour*. John Wiley, New York (1949)
4. Schwenker, F., Sommer, F., Palm, G.: Iterative Retrieval of Sparsely Coded Associative Memory Patterns. *Neural Networks* 9, 445–455 (1996)
5. Willshaw, D., Buneman, O., Longuet-Higgins, H.: Non-holographic Associative Memory. *Nature* 222, 960–962 (1969)
6. Palm, G.: On Associative Memory. *Biological Cybernetics* 36, 19–31 (1980)
7. Palm, G.: Memory Capacities of Local Rules for Synaptic Modification. A Comparative Review, *Concepts in Neuroscience* 2, 97–128 (1991)
8. Knoblauch, A., Palm, G.: Pattern Separation and Synchronization in Spiking Associative Memories and Visual Areas. *Neural Networks* 14, 763–780 (2001)
9. TIMIT Acoustic-Phonetic Continuous Speech Corpus. National Institute of Standards and Technology, Speech Disc 1-1.1, NTIS Order No. PB91-505065 (1990)
10. Markert, H., Knoblauch, A., Palm, G.: Modelling of syntactical processing in the cortex. *Biosystems* 89, 300–315 (2007)
11. Fay, R., Kaufmann, U., Knoblauch, A., Markert, H., Palm, G.: Combining Visual Attention, Object Recognition and Associative Information Processing in a Neurobotic System. In: Wermter, S., Palm, G., Elshaw, M. (eds.) *Biomimetic Neural Learning for Intelligent Robots*. LNCS (LNAI), vol. 3575, pp. 118–143. Springer, Heidelberg (2005)
12. Arbib, M.A., Billard, A., Iacoboni, M., Oztop, E.: Synthetic brain imaging: grasping, mirror neurons and imitation. *Neural Networks* 13(8/9), 931–997 (2000)
13. Roy, D.: Learning visually grounded words and syntax for a scene description task. *Comput. Speech Lang.* 16(3), 353–385 (2002)
14. Kirchmar, J.L., Edelman, G.: Machine psychology: autonomous behavior, perceptual categorization and conditioning in a brain-based device. *Cereb. Cortex* 12(8), 818–830 (2002)
15. Billard, A., Hayes, G.: DRAMA, a connectionist architecture for control and learning in autonomous robots. *Adapt. Behav. J.* 7(1), 35–64 (1999)