# Word Recognition and Learning based on Associative Memories and Hidden Markov Models

Zöhre Kara Kayikci, and Günther Palm

*Abstract*—A word recognition architecture based on a network of neural associative memories and hidden Markov models has been developed. The input stream, composed of subword-units like word-internal triphones consisting of diphones and triphones, is provided to the network of neural associative memories by hidden Markov models. The word recognition network derives words from this input stream. The architecture has the ability to handle ambiguities on subword-unit level and is also able to add new words to the vocabulary during performance. The architecture is implemented to perform the word recognition task in a language processing system for understanding simple command sentences like "bot show apple".

*Keywords*—Hebbian learning, hidden Markov models, neural associative memories, word recognition.

## I. INTRODUCTION

SPEECH recognition can be generally defined as the problem of recognizing the words from a given dictionary, relying on the information contained in the spoken speech signal.

In this study, the preprocessing of the spoken speech signal is done by hidden Markov models (HMMs) [1][2] to generate a sequence of corresponding subword-units such as context dependent phonemes or syllables and the word recognition is implemented in an architecture based on neural associative memories (NAMs)[3]-[5], where the words within the dictionary are stored.

The word recognition architecture is a network of heteroassociative memories that process the subword-unit stream generated by the HMMs in order to determine the word that matches the input stream best.

The words are stored in the system using sparsely distributed representations with respect to their transcriptions on subword-unit level. Hence, as the number of units changes with the chosen subword-unit type, the memory usage of the architecture depends on the chosen type.

The architecture is also integrated into a language understanding system which is able to understand and to react to simple command sentences like "bot show apple" on a small vocabulary consisting of 43 words. The system receives a spoken command and analyses it with respect to a given grammar to extract the meaning of the command. The architecture on language level is a network that consists of 18 interconnected modules, each containing a NAM of spiking neurons, the so-called spike counter model [6]. After the words (or superpositions of words) are generated by the word recognition network, the stream of words is then forwarded to the language module for semantical interpretation. In this paper, however, we will focus on the word recognition architecture.

The word recognition architecture can handle ambiguities that occur because of subword-units incorrectly recognized by HMMs. In the case that it is not able to decide on a unique word, the set of all alternative words is held to be forwarded to the architecture on language level to resolve the ambiguity on word level with respect to the context information [7].

The architecture based on NAMs has also the important advantage over standard HMMs that it is able to learn new words encountered during performance, which is computationally expensive for HMMs due to the fact that the dictionary, language model and many parameter files required for HMMs need to be rebuilt and, if it is necessary, new subword-unit HMMs must also be created and trained for new words. The learning process is initiated by a special sentence "this is X" where "X" is the novel word. While learning novel words, the HMMs generate a subword-unit transcription for the unknown word based on the available subword-units, which is used to create a new cell assembly in the heteroassociative memories involved in the learning process. After learning, the new word can be recognized as previously stored words.

## II. THE WORD RECOGNITION SYSTEM

### A. Neural Associative Memories

The memories in the architecture are implemented based on the binary Willshaw model of neural associative memories which uses binary neurons and synapses [3][8][9][10][11]. The binary Willshaw model is efficient with respect to storage capacity, fault tolerance and retrieval efficiency [8][9][10].

Using sparsely coded patterns, which mean low information

content per pattern, it is possible to store large pattern sets in associative memories and retrieve the stored patterns with low error probability, thus reaching high storage capacity values.

If the stored patterns are sparse (i.e. have a low density of ones) and the density of ones in the memory matrix is equal to 0.5, $P(w_{ij} = 1) = 0.5$, the binary Willshaw model of associative memory has a maximal asymptotic storage capacity of $\ln 2 \approx 0.7$ bits per binary synapse [3].

Therefore, the patterns are represented in the memories as binary sparse vectors of length n containing k active entities where k is usually much smaller than n. This allows us to effectively adapt the word recognition system for large vocabulary speech recognition tasks.

The patterns are stored in the architecture by "Hebbian" learning rule [12]:

$$w_{ij} = \bigvee_{k=1}^{M} X_i^k \cdot Y_j^k \tag{1}$$

where $M$ is the number of stored patterns, $X^k$ is the input pattern and $Y^k$ is the address pattern.

In the architecture, different retrieval strategies are employed in different memories.

One of these strategies is one step retrieval with threshold, where the threshold is set to a global value.

$$Y_j^k = 1 \iff (X^k W)_j \geq \theta \tag{2}$$

where $\theta$ is the global threshold.

A special case of this strategy is Willshaw's strategy, where the threshold is set to the number of ones in the binary input vector $X$.

### B. Hidden Markov Models

The HMMs are used to provide the input stream of subword-units to the word recognition network, e.g. word-internal triphones composed of diphones, defined as $p+p_R$ (first phoneme with right context phoneme) or $p_L$-$p$ (last phoneme with left context phoneme), and triphones, defined as $p_L$-$p$+$p_R$ (central phoneme with left and right context phonemes), where $p_L$ is the phoneme preceding $p$ and $p_R$ is the phoneme following $p$.

The topology of HMMs are three-state continuous 8-Gaussian triphone models [1][2]. The design of the triphone models follows the standard flat start Baum-Welch reestimation strategy with decision tree based triphone creation and clustering [2]. The models are trained with the training set of TIMIT speech corpus [13] and our own speech data composed of 70 different simple command sentences like "bot show plum", "but put red apple (to) plum", each of which is spoken by 4 different speakers. For the implementation presented here, word-internal triphones are used as subword-units. Therefore, in order to get a word-internal triphone-level transcription of the auditory input, a word-internal triphone-level bigram language model is created with respect to the

speech data.

### C. Word Recognition Architecture

Fig. 1 shows an overview of the architecture of the word recognition network. Each box in Fig. 1 corresponds to a heteroassociative memory. The word recognition network consists of 5 heteroassociative memories HM1-5 and an area HMO to represent the global output of the memories HM1-3.
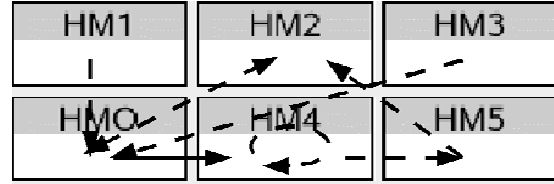


Fig. 1 Word recognition architecture based on heteroassociative memories

The memories are interconnected with each other via hetero- and autoassociative connections. In Fig. 1, the dashed arrows denote autoassociative and solid arrows denote heteroassociative connections.

All the heteroassociative memories except for HM5 consist of n neurons, where n is the number of word-internal triphones and the memory HM5 contains 200 neurons. The network receives the sequence of triphones recognized by HMMs and retrieves the word that matches the best. The memory HM1 serves as an input area and presents the triphone received from the HMMs to the network. HM2 represents the triphone expected in the next step and predicts it with respect to the word hypothesis (or hypotheses) activated in the memory HM5 and the triphone(s) represented in the area HMO in the current step. The memory HM3 stores which triphones follow each other with respect to the words in our own training data.

During retrieval, the outputs of the memories HM1-3 are summed up and a common threshold is applied. This helps the network to correct the spurious triphones, which may cause ambiguities on the word level.

The global output of the memories HM1-3 is then represented in the area HMO and is forwarded to HM4. The memory HM4 activates the triphones that have been processed by the network up to the current step. HM5 stores the words in the vocabulary using their word-interval triphone level transcriptions and is responsible for generating word hypotheses with respect to the triphones activated in HM4.

It will be demonstrated how the word recognition network processes an example command sentence "bot lift red ball". The corresponding word-internal triphone-level transcription generated by the HMMs is given "b+ow b-ow+t ow-t sp l+ih l-ih+f ih-f+t f-t sp b+r b-r+eh r-eh+d eh-d sp ao+l ao-l sp" where "sp", which is used to determine the word boundaries, denotes "small pause" between words. In the example transcription, the phonetic transcriptions for the words "red" and "ball" were incorrectly recognized by HMMs. The correct transcriptions should have been "sp r+eh r-eh+d eh-d sp b+ao b-ao+l ao-l sp".

| HM1 | HM2 | HM3 |
|-----|-----|-----|
| b+r |     |     |

| HMO | HM4 | HM5 |
|-----|-----|-----|
| b+r | b+r | brown |

Fig. 2 The processing of the first triphone "b+r" in the network

Fig. 2 shows the word recognition system which processes the first word-internal triphone in the transcription part "b+r b-r+eh r-eh+d eh-d". As shown in Fig. 2, the pattern "b+r" is activated in the memory HM1 and the memories HM2 and HM3 do not receive any input at the beginning of the word. Therefore, they do not activate any output neurons, consistent with the fact that no expectation can be generated in the beginning of the word recognition. After the first triphone has been stored in HM4, the word pattern "brown" is activated in HM5 with respect to the activated triphone in HM4.

| HM1 | HM2 | HM3 |
|-----|-----|-----|
| b-r+eh | b-r+aw | b-r+aw |

| HMO | HM4 | HM5 |
|-----|-----|-----|
| b-r+eh b-r+aw | b+r b-r+eh b-r+aw | brown |

Fig. 3 The word recognition network after processing the second triphone "b-r+eh"

In the next step (see Fig. 3), the second triphone "b-r+eh" enters the input area HM1 and the memory HM2 activates the triphone "b-r+aw" expected in the next step according to the activated word in HM5 and the triphones represented in HMO. HM2 and HM3 represent the triphones expected in the current step. After applying a global threshold, the resulting triphones are represented in HMO and are also stored in HM4. Then, the word pattern "brown" is activated with respect to word-interval triphones activated in HM4.

The word recognition network processes the third word-internal triphone "r-eh+d" in the same way (see Fig. 4) as in the previous step and the words "brown" and "red" are activated.

| HM1 | HM2 | HM3 |
|-----|-----|-----|
| r-eh+d | r-aw+n | r-aw+n |

| HMO | HM4 | HM5 |
|-----|-----|-----|
| r-eh+d r-aw+n | b+r b-r+eh b-r+aw r-eh+d r-aw+n | brown red |

Fig. 4 The word recognition network after processing the third triphone "r-eh+d"

| HM1 | HM2 | HM3 |
|-----|-----|-----|
| eh-d | aw-n eh-d | aw-n eh-d |

| HMO | HM4 | HM5 |
|-----|-----|-----|
| eh-d | b+r b-r+eh b-r+aw eh-d r-eh+d r-aw+n | red |

Fig. 5 The final word is activated after processing the last triphone of the word "eh-d" in the network

As shown in Fig. 5, the last word-internal triphone of the word "eh-d" is activated in the area HM1 and the expected triphones are activated in the memories HM2 and HM3. The triphone having the highest activation is represented to HMO and then forwarded to HM4. Finally, the word "red" is retrieved with respect to the word-internal triphones in HM4 by applying a threshold.

| HM1 | HM2 | HM3 |
|-----|-----|-----|
| ao+l |     |     |

| HMO | HM4 | HM5 |
|-----|-----|-----|
| ao+l | ao+l |     |

Fig. 6 The network receives the first triphone of the last word in the sentences

Fig. 6 shows the processing of the last word transcription part "ao+l ao-l" in the HMM output. After processing the first word-internal triphone "ao+l", the network can not generate a word hypothesis due to the fact that the activated assembly in HM4 can not activate any neuron in the memory HM5. Therefore, no word hypothesis is generated in the memory HM5.

| HM1 | HM2 | HM3 |
|-----|-----|-----|
| ao-l |     |     |

| HMO | HM4 | HM5 |
|-----|-----|-----|
| ao-l | ao-l ao+l | ball wall |

Fig. 7 The state of the network after processing the last triphone

In the next step (see Fig. 7), the second word-internal triphone has been processed and a superposition of the assemblies "ball" and "wall" is activated with respect to the triphones in HM4. Since the network can not solve the ambiguity on subword-unit level, it generates a superposition of the word patterns which will be forwarded to the language processing system to resolve the ambiguity on sentence level [14]. By using a bidirectional connection which supports matching pairs of verbs and objects, in this case "lift" and

"ball", the language system is able to successfully resolve the ambiguity.

### III. Incremental Learning

In many speech recognition applications such as HMMs, it is usually impossible to increase the vocabulary size during performance, and if it is possible, many parameter files required for the application have to be changed and HMMs have to be re-trained. The network architecture presented here allows for a relatively easy enlargement of the vocabulary by learning of new words during performance.

In the implementation, learning is triggered by a special command "this is X" where "X" is the novel word. First, HMMs preprocess the auditory input "this is X" to generate a plausible word-internal triphone sequence for the novel word. To this purpose, we have chosen a large bigram language model composed of TIMIT sentences and our simple command sentences. The word recognition network uses the command "this is" to start the learning process. During this process, the learning takes place in the heteroassociative memories HM2, HM3 and HM5. The memory HM2 stores the novel word using a randomly generated 5 out of 200 binary code vector (as input pattern) and a binary code vector (as output pattern) created with respect to the word-internal triphone sequence generated by HMMs for the new word. In the memory HM3, new associations are stored with respect to the information which triphones follow each other in the new word transcription. The memory HM5 stores the new word using the same binary code vectors in HM2. In HM5, the binary code vector with respect to the new word-interval sequence from HMMs is used as input pattern and the randomly generated 5 out of 200 binary code vector is used as output pattern. After learning, the new word can be used and processed exactly as the previously stored words. Thus the system can correctly recognize the word "apricot" in a sentence like "bot show apricot" after "apricot" has been learnt.

### IV. Discussion

A word recognition architecture based on neural associative memories and HMMs for a language processing system [14] is presented. The model deals with finding out the sequence of words from an input stream of subword-units (e.g. word-internal triphones) generated by HMMs.

It is also able to solve and represent the ambiguties that occur due to the fact that the HMMs can not correctly generate a subword-unit transcription of the spoken words. In this case, if it is not possible to make a unique decision on word level, then the ambiguity is kept on word level by creating a superposition of several alternative words and forwarded to a higher (sentence) level in the language processing system to be resolved using contextual information [7].

The word recognition system is constructed in such a way that it has the ability to enlarge its vocabulary by learning new words during performance. Compared to subword-units based standard hidden Markov models for word recognition, where adding a new word to the vocabulary involves the modification to the pronouncing dictionary and the language model, the presented architecture only requires a subword-unit-level representation from the HMMs (the number of the new representations for the novel word can be more than one to increase the performance of the system) which is used to generate new patterns in the corresponding associative memories for the novel word.

The type of the subword-units, the bigram language model used in the HMMs, and the size of the vocabulary have a large effect on the computational complexity, the memory requirements and the speed of the system. If the words have many overlaps in their subword-unit transcriptions, the computation time of the system is also increased. The speed of the system was analyzed in terms of short and long sentences on a standard computer (Intel Pentium 4 2.66 MHz). An application toolkit for HTK (HMM Toolkit) is used for the implementation of HMMs[15]. It takes 21 seconds for short sentences like "bot show plum", whereas it is measured as around 38 seconds for long sentences like "bot put red apple (to) blue plum" and it takes 26 seconds to learn a new word in a sentence like "this is apricot", most of which is taken for the HMMs to generate the subword-unit transcriptions.

The recognition performance of the system has been evaluated on a small vocabulary of 43 different words. The test data is composed of 35 simple command sentences from 4 speakers and there are totally 504 word tokens in the test set. On the test data the presented system recognized 98% of the word tokens, whereas HMMs achieved 96% [14].

Due to the large storage capacity of NAMs, this system can be extended to larger vocabularies. A larger speech corpus of 279 German bus stop names has also been used. The training set consists of 14 speakers, whereas test set contains 5 speakers, and each bus stop name is spoken by every speaker. The number of German triphones used to create the heteroassociative memories in the architecture is 1284. The architecture yielded a word recognition accuracy of 98%. Compared to a HMM based recognizer, which achieves 99% word recognition accuracy, there is a slight difference between the presented architecture and the pure HMMs [16]. This is due to the word-interval triphone HMMs and language model used. The performance of the presented model can be increased by using demisyllable or syllable HMMs as subword-units and a more efficient language model.

### References

[1] L. Rabiner and B. H. Juang, *Fundemantal of Speech Recognition*, Prentice-Hall, Inc., Upper Saddle River,, 1993.

[2] S. Young, et al., *The HTK book for HTK version 3.2.1.*, Cambrige University, Engineering Department, 2002.

[3] G. Palm, "On associative memory," *Biological Cybernetics*, vol. 36, pp. 19-31, 1980.

[4] F. Schwenker, F. T. Sommer and G. Palm, "Iterative retrieval of sparsely coded associative memory patterns," *Neural Networks*, vol. 9(3), pp. 445-455, 1996.

[5] G. Palm, F. Kurfess, F. Schwenker and A. Strey, *Neural Associative Memories. Technical Report*, Universitat Ulm, Germany, 1993.

[6] A. Knoblauch and G. Palm, "Pattern separation and synchronization in spiking associative memories and visual areas," *Neural Networks*, vol. 14, pp. 763-780, 2001.

[7]   H. Markert, A. Knoblauch and G. Palm, "Modeling of syntactical processing in the cortex," *BioSystems*, vol. 89, pp. 300-315, 2007.

[8]   D. Willshaw, O. Buneman and H. Longuet-Higgins, "Non-holographic associative memory," *Nature*, vol. 222, pp. 960-962, 1969.

[9]   G. Palm, "Memory capacities of local rules for synaptic modification. A comparative review," *Concepts in Neuroscience*, vol. 2, pp. 97-128, 1991.

[10]  J. Buckingham and D. Willshaw, "Performance characteristics of associative nets," *Network:Computation in Neural Systems*, vol. 3, pp. 407-414, 1992.

[11]  F. T. Sommer and G. Palm, "Improved bidirectional retrieval of sparse patterns stored by hebbian learning," *Neural Networks*, vol. 12, pp. 281-297, 1999.

[12]  D. O. Hebb, *The Organization of Behaviour*, John Wiley, Newyork, 1949.

[13]  TIMIT Acoustic-Phonetic Continuous Speech Corpus, National Institute of Standartsand Technology Speech Dics 1-1.1, NTIS Order No. PB91-505065, 1990.

[14]  Z. Kara Kayikci, H. Markert and G. Palm, "Neural associative memories and hidden Markov models for speech recognition," presented at 2007 Int. Joint Conf. on Neural Networks, Orlando, Florida (USA).

[15]  S. Young*, ATK An Application Toolkit for HTK Version 1.6*, Machine Intelligence Laboratory, Cambrige University, Engineering Department, 2007.

[16]  Z. Kara Kayikci, D. Zaykovskiy, H. Markert, W. Minker and G. Palm *Distributed Architecture for Speech Controlled Systems Based on Associative Memories* Chapter in Mathematical Analysis of Evolution, Information and Complexity, Wiley-VCH, Weinheim (Germany), unpublished.