Large Vocabulary Continuous Speech Recognition using Associative Memory and Hidden Markov Model

ZÖHRE KARA KAYIKCI
Institute of Neural Information Processing
Ulm University
89069 Ulm
GERMANY

GÜNTER PALM
Institute of Neural Information Processing
Ulm University
89069 Ulm
GERMANY

Abstract: We attempted to improve recognition accuracy, avoiding extensive retraining when the vocabulary is changed or extended, by applying a hidden Markov model and neural associative memory based hybrid approach to continuous speech recognition. In this approach hidden Markov models are used for subword-unit recognition such as syllables. For a given subword-unit sequence a network of neural associative memories generates first spoken single words and then the whole sentence. The fault-tolerance property of neural associative memory enables the system to correctly recognize words although they are not perfectly pronounced or run into each other. The approach are evaluated for TIMIT, and for WSJ1 5k and 20k test sets. The obtained results are encouraging.

Key–Words: Continuous Speech Recognition, Hidden Markov Models, Neural Associative Memory

1 Introduction

State-of-the-art continuous speech recognition systems are usually based on the use of Hidden Markov models (HMMs). However, HMMs suffer from several difficulties, concerning increasing dictionary size, different speaking styles of speakers, and weakness to environmental conditions. In recent years a variety of hybrid approaches based on HMMs and artifical neural networks (ANNs) have been introduced to augment the performance of speech recognizers. Some of these works involved the attempt of ANN architectures to emulate HMMs [1], and ANNs were used to estimate the HMM state-posterior probabilities from the acoustic observations [2]. In another approach, the ANN is used to extract observation feature vectors for a HMM [3].

In this paper, we introduce a novel approach based on HMMs on the elementary subword unit level and neural associative memories (NAMs) on a higher level, such as word and sentence levels. A NAM is a realization of an associative memory in a single layer artificial neural network. For large vocabulary continuous speech recognition (LVCSR), context-dependent phonemes are usually used to model the elementary acoustic units of speech due to the insufficient amount of training data. In our approach, a context-dependent phoneme recognizer is used to find the best subword unit sequence for a given speech utterance. These subword units are longer than the context dependent phonemes like syllables. First, HMMs generate a sequence of subword units and provide it to a network of NAMs on a higher level. At the second stage of recognition, single words are first recognized from the HMM output stream and the whole sentence is retrieved according to the recognized single words. The memory usage of the associative memories is proportional to the number of distinct subword units and the number of words required for a given recognition task. This number is a function of the vocabulary size and increases in general with the vocabulary size. Thanks to the advantages of pattern completion and fault tolerance of NAM, the network of NAMs is able to handle ambiguities on different levels that occur due to the spurious subword-units (incorrectly recognized by HMMs) in the input stream. The goal of the presented approach is to take advantage of both HMMs and NAMs in order to improve recognition performance for large vocabularies and to generate more flexible recognition systems. This paper first describes the hybrid system and evaluates it for TIMIT [6], WSJ1 5k and 20k “hub” test sets. The results are then compared with other studies in the literature.

2 Speech Material

2.1 TIMIT

TIMIT [6] is manually labelled and includes time-aligned, manually verified phone and word segmentations. For this study, the original set of phonemes was reduced to a set of 45 phonemes. The speech data is composed of three sets: a set for training the acoustic models, a development set for optimising language model scaling factor, and word insertion penalty, and a test set for evaluating the acoustic models. Table 1 shows details of the data.

Table 1: *TIMIT data sets.*

	Train	Test	Devel.	Total
Word tokens	30132	9455	1570	41157
Speakers	462	144	24	630

2.2 Wall Street Journal

The Wall Street Journal (WSJ) corpus consists of two parts, WSJ0 and WSJ1. The corpus covers 284 different speakers. The training data is formed by combining training data from both WSJO and WSJ1.

We have worked on two test sets: the 5k (4986) word closed vocabulary and 20k (19979) word open vocabulary non-verbalized pronunciation WSJ tasks. The WSJ1 5k development test has 2076 distinct words and a total of 13866 words. For the 5k word closed vocabulary task the si_dt_05.odd set is used, which is a subset of the WSJ1 5k development test data formed by deleting sentences with out-of-vocabulary (OOV) words, choosing every other remaining sentence, and thus is comprised of 248 sentences from 10 speakers.

The WSJ1 20k development test has 2464 unique words with the total count of 8227 words and contains 187 out of vocabulary words. 2.27 % of the word occurences in the development set are not included in the standard 20k-word vocabulary. The WSJ1 20k development test data consists of 503 sentences from 10 speakers.

3 System

Fig. 1 shows the block diagram of the system based on the presented approach. The first block is a set of HMMs that transforms the speech utterance into a sequence of syllables. The reason for the use of a syllable as an output subword unit is that the subword unit accuracy for syllables is higher than that for context-dependent phonemes. The resulting syllables are then sent to the second block which is a sentence recognition module consisting of a word recognition network and a sentence recognition network. The word recognition network extracts single words from this syllable stream and the sentence recognition network finds the output sentence containing most of the recognized words.

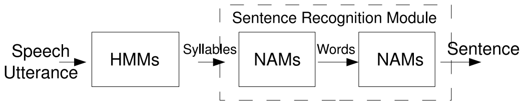

Fig. 1: The block diagram of the system.

3.1 Phoneme-based HMM

For TIMIT and WSJ the HMM systems are separately developed using Sphinx-4 speech recognition system [7]. The acoustic waveforms from TIMIT and WSJ are parameterized into 13-dimensional cepstrum along with computed delta and delta-deltas. While a set of 45 phonemes and a silence model is used for TIMIT, the system uses 50 phonemes and a silence model for WSJ.

The context-dependent phoneme-based systems follow a general strategy for acoustic model training. All phone models are three-state left-to-right HMMs without skip states. The training procedure essentially involves the four steps, as follows:

- ○ Single Gaussian monophone models are created and initialized with the global mean and variance of the training data and trained using reference transcription derived from the pronunciation dictionary.
- ○ All cross-word triphones that occur in the training corpus are created by copying the monophone models for each required triphone context and the transition matrices across all the triphones of each base phone are tied. Then, the models are retrained.
- ○ For each group of triphones sharing the same base phone, a decision tree is computed to cluster the states into equivalence classes ensuring that enough data to train will be associated with each cluster. The distributions of all the states in each equivalence cluster are tied. The state-clustered triphones are then retrained.
- ○ The number of mixture components in each state is successively incremented by splitting single Gaussian distributions into mixture distributions.

Further details of the training procedure are given in [7]. The experiments are run using syllable level trigram language models.

3.2 Sentence recognition module

3.2.1 Neural associative memories

A NAM realizes a mapping between an input space and an output space, which can be specified by learning from a finite set of patterns. There are two types of associative memory, namely *hetero-associative* and *auto-associative* memory. In heteroassociative memories, a mapping $x \mapsto y$ is stored, a content pattern y is addressed by its input pattern x. In auto-associative memories, the content pattern y is equal to the corresponding input pattern x. We have chosen Willshaw's simple binary model of associative memory [8, 9]. The typical representation of a NAM is a matrix. The binary patterns are stored by a "Hebbian" learning rule [10]:

$$w_{ij} = \bigvee_{k=1}^{M} x_i^k y_j^k, \tag{1}$$

where M is the number of patterns, x_k is the input pattern, y_k is the output pattern and w_{ij} corresponds to the synaptic connectivity weight between neuron i in the input population to neuron j in the address population.

Retrieving is performed by a one-step retrieval strategy with threshold:

$$y_j^t = 1 \Leftarrow (Wx^t)_j \geq \Theta, \tag{2}$$

where the threshold Θ is set to a global value and y is the content pattern.

3.2.2 Architecture

Fig. 2 shows an overview of the sentence recognition module which consists of two parts: word recognition network (left of Fig. 2) and the sentence recognition network (right of Fig. 2). Each box in Fig. 2 corresponds to an associative memory.

The word recognition network consists of 5 interconnected associative memories and a representation area SWU, where the memories M1 and M3 are autoassociative memories, while M2, WRD and M4 are heteroassociative memories. The basic idea in this approach is that the word recognition network generates a list of word hypotheses in terms of the syllables processed each time a new syllable is read from the HMM output sequence. The number of neurons used in all the associative memories, except for WRD, depends on the number of distinct subword units required for the recognition task and in the case of the memory WRD, the dependence is on the size of the task vocabulary.

For continuous speech recognition tasks based on subword units, it is usually difficult to determine word boundaries because there is no boundary between words such as a small pause. Therefore, in our approach the word boundary is detected when there is no transition between the current and the subword units previously recognized by the network during recognition of the current word. However, in this way the word recognition network always searches for a long word. If two short words come subsequently in a sentence and a long word consisting of these two words exists in the vocabulary, it is not possible to correctly recognize these two adjacent short words at this level of the architecture. But this problem can be solved on the upper (sentence) level of architecture using additional information such as syntax.

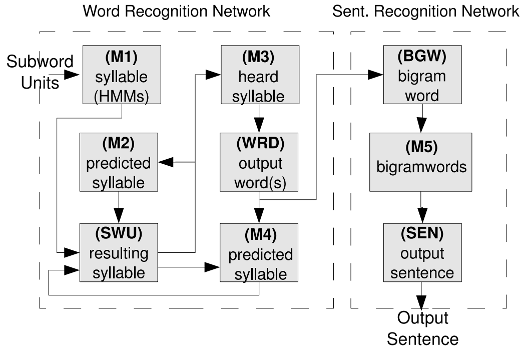

Fig. 2: Overview of the sentence recognition module and its internal connectivity.

The memories M1 and M3, each of which is a memory matrix of dimension $n \times n$ (n is the number of distinct syllables in the task vocabulary), store syllables in columns using 1 out of n sparse binary code vectors as input and output patterns. The memory M2, a memory matrix of dimension $n \times n$, stores the syllable transitions within the words in the vocabulary using 1 out of n sparse code vectors. The memory WRD is a memory matrix of dimension $n \times r$ (r is a specific number, 5000), and M4 is a memory matrix of dimension $r \times n$. They store each word in the vocabulary using two representations, i.e. the syllablic transcription of the word as k out of n code vector (k is the number of syllables involved in the word) and a randomly generated 2 out of r sparse binary code vector. For each word, the input and output patterns in WRD are given as syllable-level transcription and as the randomly generated code vector of length r, respectively, while the input and output patterns in M4 are used inversely.

In order to simplify the explanition of the retrieval a "global time step" is introduced. In one global time step, each memory performs one pattern retrieval, and the results are forwarded to subsequent memories. All memories work in parallel.

M1 serves as an input module and presents the HMM output syllable to the network. M2 represents the possible syllables which follow the resulting syllable(s) (in SWU) in the previous retrieval step. M4 represents the expected syllable in the current global time step for the word hypothesis generated (in WRD) in the previous global time step. But, at the beginning of each word the memories M2 and M4 do not represent any syllable due to the fact that no expectation can be generated in the beginning of the word recognition process. The outputs of the memories M1, M2 and M4 are summed up and a common threshold is then applied. In this way, the spurious syllables, which can cause ambiguities on the word level, may be corrected by the network. The resulting syllable is represented in the area SWU.

The memory M3 stores the processed syllables up to the current step. Each time a new syllable has been recognized and stored in M3, WRD is responsible for generating a word hypothesis or superposition of word hypotheses with respect to the syllables activated in M3. When a word boundary is detected, the iterations for the current word end. If the word recognition network can not decide on a unique word representation for a given syllable sequence, a superposition of word hypotheses matching the input sequence is generated by the network. After recognition of each word hypothesis (or superposition of word hypotheses), it is forwarded to the network that is responsible to recognize the sentence.

The second part in the architecture is the sentence recognition network which consists of one autoassociative memory M5 and two heteroassociative memories BGW and SEN. Given a sequence of words (or superpositions of words), it recognizes the output sequence of word trigrams. The memory BGW is a memory matrix of dimension $V \times L$, where V is the number of words in the vocabulary and L is the number of word bigrams in the test set, and transforms two sequential output words into a binary bigram representation. The memory M5 is a memory matrix of dimension $(L) \times (L)$. It stores the bigram representations of the output words. The last memory SEN is a memory matrix of dimension $K \times K$, where K is the number of word trigrams in the test set. After recognition of all words, all the bigram representations are sent to SEN as input and the output sequence of word trigrams are recognized.

4 Experiments

The presented hybrid system was evaluated on TIMIT test set, the 5k (4986) word closed vocabulary and 20k (19979) word open vocabulary nonverbalised pronunciation WSJ tasks. TIMIT vocabulary contains 6218 distinct words, 17983 word bigrams and 20075 word trigrams.

For 20k WSJ open test, over 2% of the word occurrences are not included in the standard 20k-word vocabulary. Naturally, words that are not in the vocabulary can not be recognized accurately. The 20k-word open vocabulary contains 5965 syllables, 6543 word bigrams and 7342 word trigrams. The 5k-word closed vocabulary contains 2682 syllables, 6241 word bigrams and 7514 word trigrams.

A speech utterance such as “japan plays by different rules ones rigged for the producer” is first processed by a phoneme-based HMM and a syllable sequence is then generated, e.g. “START jh_ah p_ae_n p_l_ey_z b_ay d_ih f_er *** r_uw_l_d w_ah_n_z r_ih_g_d f_ao_r dh_ah p_r_ah d_uw s_er END”, where the last syllable “***” of the word “different” can not be recognized (it should have been “ah_n_t”) and the single syllable word “rules” is also incorrectly recognized as “r_uw_l_d”, which should have been “r_uw_l_z”. “START” and “END” denote the beginning and end of the sentence, respectively.

In Fig. 3, the state of the word recognition network is shown after the first syllable “jh_ah” in the HMM output sequence has been processed. M1 shows the first syllable received from the HMM output at the current global time step, while M2 and M4 do not represent any syllable due to the beginning of the word recognition process. Therefore, SWU represents the same syllable and it is forwarded to M3. The syllable in M3 does not allow for a unique word interpretation because there are many words in the vocabulary which contain the syllable “jh_ah” and thus a list (superposition) of all matching word patterns (with the highest activation) is finally displayed in WRD. Note that this additional calculation of overlaps with word patterns is only holded for display and only in the WRD memory.

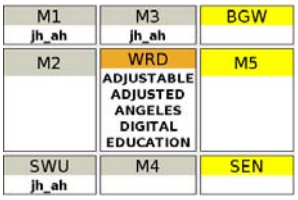

Fig. 3: The word recognition module after the first syllable “jh_ah” has been processed. Because of the limited display area of WRD, only the first 5 matching words are displayed in WRD.

Fig. 4 shows the sentence recognition module after the second syllable belonging to the word “japan” has been recognized. M1 represents the HMM output, the memories M2 and M4 represent the expected syllable at the current step with respect to the word hypotheses represented in WRD and the syllable represented in SWU in Fig. 3. The word recognition network generates a unique decision for “JAPAN” in WRD, after processing both syllables belonging to the word. Fig. 5 shows the sentence recognition module after the first word “JAPAN” has been recognized. After recognition, the generated word hypothesis is forwarded to the memory BGW to generate the bigram word representation. Since the word “JAPAN” is the first word in the sentence, the first bigram representation is given as “START+JAPAN”and stored in M5.

Fig. 6 shows the sentence recognition module after the syllables “d_ih” and “f_er” belonging to the word “DIFFERENT” have been processed. The word recognition network produces a superposition of word hypotheses in WRD containing the syllables in M3 and. The superposition of word hypotheses is then sent to BGW to generate bigram word representations.

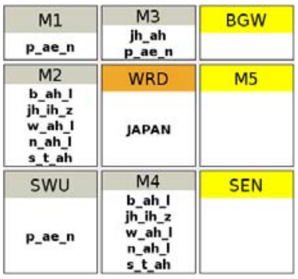

Fig. 4: The word recognition module after both syllables belonging to the word “JAPAN” have been processed.

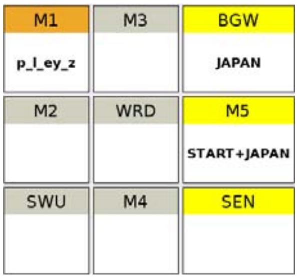

Fig. 5: The word recognition module after the first word “JAPAN” has been recognized.

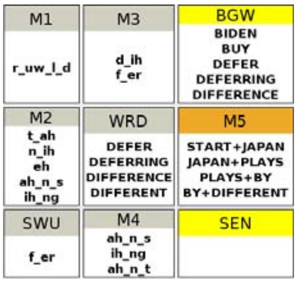

Fig. 6: The word recognition module after the incomplete set of syllables for the word “DIFFERENT” has been processed.

Fig. 7 shows the sentence recognition module after all words have been recognized. M5 stores all bigram representations of the output words generated by BGW module. These bigram representations will be used as input in SEN in order to recognize the spoken sentence. The output of SEN is a sequence of word-level trigrams of the spoken sentence and, these trigrams are used to detect the syntax of the sentence. The sentence is then extracted from this output sequence using a dynamic algorithm, e.g. “start-japan+plays japan-plays+by plays-by+different by-different+rules different-rules+ones rules-ones+rigged ones-rigged+for rigged-for+the for-the+producer the-producer+end” is transformed into “japan plays by different rules ones rigged for the producer”.

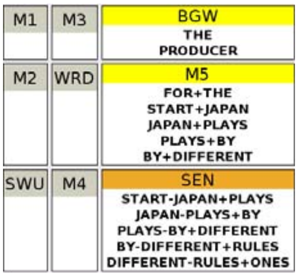

Fig. 7: The word recognition module after all words have been recognized.

5 Results

The WER results for TIMIT test set are shown in Table 2 and the system based on the proposed approach achieved a lower WER than a HMM based triphone recognizer. The WER results for the 5k and 20k development test sets of WSJ1 are given in Tables 3 and 4. It is shown that the system based on the proposed approach has decreased the word error rates substantially, compared to WERs in [4] which uses a cross-word triphone based system and in [5] which is based on language model training.

Table 2: Word error rates (WER) on TIMIT.

Recognizer Type	WER (%)
Context Dep. Phoneme [11]	8.1 ± 0.6
Our Hybrid Approach	7.03

Table 3: WER on WSJ1 5k (si_dt_5k.odd).

Recognizer Type	WER (%)
Cross-word Triphone [4]	6.09
Our Hybrid Approach	4.91

Table 4: WER on WSJ1 20k (si_dt_20k).

Recognizer Type	WER (%)
Language Training [5]	16.4
Our Hybrid Approach	13.21

6 Conclusion

In this paper, a new hybrid HMM/NAM approach to LVCSR is represented, where HMM is used on a subword-unit level and NAM is used on a higher level, such as word and sentence levels. The output of HMMs can be various types of subword units, such as context-dependent phonemes, demi-syllables or syllables. The subword unit type is chosen in terms of the highest subword unit accuracy. If the ambiguity on the subword unit level can not be solved, the system then represents the ambiguity on the word level as a superposition of all possible words and resolves the ambiguity on the word level in the syntax of the whole sentence.

The system was evaluated on TIMIT, 5k closed and 20k open vocabulary tasks of WSJ1 and considerable improvements over the performance of the HMM based recognizers were obtained. The implemented system takes advantage of NAMs, such as flexibility and fault tolerance. Thus, the network of NAMs is able to solve ambiguities that occur due to incorrectly recognized subword units or words, or pronounciation variation. On the other hand, in terms of computational complexity, the presented system has an advantage over pure HMM based recognition systems. The system utilizes a task vocabulary of syllables and the number of syllables in the vocabulary is less than that of words. Therefore, on the HMM level, it takes less time to search for the most appropirate syllable sequence for a given speech utterance. Because of the sparse representation of syllables and words in NAMs, the computational cost in NAMs is only limited for active input units. Due to the high storage capacities of the the sparse binary associative memories [9], the presented system scales well with large vocabularies.

Compared to HMMs, another advantage of NAMs is its more flexible functionality in terms of the lexicon generation. In order to enlarge the vocabulary, the modifications to the lexicon, the language model and training of new subword-unit models are necessary for HMMs, while word recognition network in the presented system needs only a sequence of subword-units from HMMs for the novel word without further training of HMMs [12].

References:

[1] J. S. Bridle, Alphanets: a Recurent Neural Network Architecture with a Hidden Markov Model Interpretation, *Speech Communication* 9(1), 1990, pp. 83–92.

[2] H. Bourlard and N. Morgan, Connectionist Speech Recognition. a Hybrid Approach, *Kluwer Academic Publisher*, 1994.

[3] Y. Bengio, A Connectionist Approach to Speech Recognition, *International J. Pattern Recognition Artificial Intelligence* 7(4), 1993, pp. 647–667.

[4] P.C. Woodland, J.J. Odell, V. Valtchev and S.J. Young, Large Vocabulary Continuous Speech Recognition using HTK, *in Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing*, 1994, pp. 125–128.

[5] R. Schwartz, L. Nguyen, F. Kubala, G. Chou, G. Zavaliagkos and J. Makhoul, On Using Written Training Data for Spoken Language Modeling, *Proceedings of the workshop on Human Language Technology*, 1994, pp. 94–98.

[6] TIMIT Acoustic-Phonetic Continuous Speech Corpus. National Institute of Standards and Technology Speech Disc 1-1.1, NTIS Order No. PB91-505065, 1990.

[7] Robust Group Tutorial, http://www.speech.cs.cmu.edu/sphinx/tutorial.html.

[8] D. Willshaw, O. Buneman and H. Longuet-Higgins, Non-holographic Associative Memory”, *Nature* 222, 1969, 960–962.

[9] G. Palm, On Associative Memory, *Biological Cybernetics* 36, 1980, pp. 19–31.

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

[11] A. Hämäläinen, J. de Veth and L. Boves, Longer-Length Acoustic Units for Continuous Speech Recognition’, *Proceedings EUSIPCO* , 2005.

[12] Z. Kara Kayikci and G. Palm, Word Recognition and Incremental Learning Based on Neural Associative Memories and Hidden Markov Models, *Proceedings of 16th ESANN*, 2008, pp. 119–124.