Comparison of Hidden Markov Model and Recurrent Neural Network in Automatic Speech Recognition

Akshay Madhav Deshmukh

Abstract—Understanding human speech precisely by a machine has been a major challenge for many years. With Automatic Speech Recognition (ASR) being decades old and considering the advancement of the technology, where the current state of the machines still does not recognize all speech, are still used in many applications and services regularly. Hence, to facilitate and strengthen the research, it is very important to recognize the significant research directions, especially those that lacked persuasion and funding in the past. Due to the application of Deep Neural Networks (DNNs), the ASR systems which are built on Hidden Markov Model (HMM) has shown a significant improvement in its performance. Despite the progress in the performance of ASR systems, building one has always remained a huge challenge necessitating the requirement of many resources and training stages. The idea of using DNNs for Automatic Speech Recognition has gone further from being a single component in a pipeline to building a system mainly based on such a network.

This paper provides a literature survey on state of the art researches on major models, namely Deep Neural Network - Hidden Markov Model (DNN-HMM) and Recurrent Neural Networks trained with Connectionist Temporal Classification (RNN-CTC). It also provides the differences between these two models at the architectural level.

Index Terms—Recurrent Neural Network, Deep Neural Network, Automatic Speech Recognition, Hidden Markov Model, Gaussian Mixture Model

I. INTRODUCTION

The technology of Automatic Speech recognition (ASR) concedes a system to recognize human speech and produce the output. The process commences when a sentence is spoken by the speaker i.e. sequence of words including the vocalized pauses (ah, umm, oh). The system produces a speech waveform which epitomizes the words of the sentence as well as the vocalized pauses in the input in the form as speech. Later, the system decodes the speech providing the best fit of the sentence. Primarily, it converts the speech signal into a sequence of vectors that is measured at the duration of the speech signal. Later, with the assistance of the decoder, it generates a valid sequence of sentences [27].

Today, the ASR systems that are available do not need a long period of speech training and can successfully recognize uninterrupted speech with large set of vocabulary with high accuracy rate. Most existing speech recognition softwares have an accuracy rate of 98% to 99% if performed under favorable conditions, i.e. having similar speech characteristics and training data, achieving significant speaker adaptation and performing under a noise-free environment.

Since the birth of ASR in late 1970’s, it has witnessed a drastic change in its field and has eventually evolved from its infancy to its coming of age and into an instantaneous growth in applications and commercial markets. In spite of its drastic achievements in such a short span, ASR still remains an unsolved problem.

This section provides an a rundown of the major developments in ASR in the field of infrastructure, knowledge representation, models and algorithm, search, and metadata. Section [II] gives an overview of Hidden Markov Model based hybrid systems and their architectural progress. Section [III] presents the Recurrent Neural Network based ASR systems and the pipeline required to build an end-to-end recognizer. In section [IV] we discuss various experiments to juxtapose the performance of both systems, and finally conclude in section [V].

A. Infrastructure

Moore’s Law states, "that we can anticipate doubling the amount of computation achievable for a given cost every 12 to 18 months, alongside with a comparable narrowing cost of memory” [22]. Considering this, researchers have been able to run complex algorithms in sufficiently shorter time frames to make a remarkable significant progress.

B. Models and Algorithms

In the early 1970’s there was a huge transition from speech recognition towards the statistical methods, specifically stochastic processing with Hidden Markov Models (HMMs) [3]. Statistical methods are required in recognizing the continuous speech which involves modeling extraction of the models’ statistical parameters, hypothesis search procedures and other linguistic decoding methods [21]. Even after three decades this methodology is still in the act and predominates. In iterative computation of maximum-likelihood estimates the Expectation Maximization (EM) algorithm is used from incomplete data set. Since each iteration has to undergo two steps: Expectation step and the Maximization step, hence the name derived as Expectation Maximization Algorithm [9] and forwardbackward algorithm which is an inference algorithm which computes the values in two iterations which are required...
for achieving marginal distributions. The first and the second iterations goes forward and backward in time respectively. Thus the name Forward-Backward Algorithm [5].

In statistical discriminative techniques and corrective training the prominent roles are played by Maximum Mutual Information (MMI) and minimum-error model parameters. The values of maximum likelihood estimation through forward-backward algorithm does not provide a maximum recognition accuracy. Hence to overcome this problem an estimation procedure called corrective training is used which results in minimizing the number of recognition errors [2]. Deterministic approaches involve some neural network techniques such as Hopfield net and Hamming net [23].

With a broad array of versatile conditions with respect to vocabulary set, speaker, environment, channel and so on, getting adapted to these versatile conditions plays a prominent role. The most approved techniques include maximum a posteriori probability (MAP) estimations. The prior distribution family, the specification of the parameters of prior densities and the MAP estimates evaluation are the main issues of MAP estimates [11].

C. Search

Search strategies and key decoding were traditionally developed in non-speech applications, specifically dedicated on stack decoding. Stack decoding uses stack storage at the receiver side and compared to Fano algorithm it is much simpler to analyze and six times faster than the other [22]. The viterbi search algorithm which is derived from dynamic programming, whose role is to find the most likely sequence of hidden states (viterbi path) which leads to a series of events. It is usually applied to search alternative hypotheses. Beam search algorithm is a type of heuristic search algorithm that traverses a graph by expanding the most promising node (i.e. non-promising nodes can be pruned at any step in the search). The set of most promising search nodes are called a "beam". In majority of the ASR systems beam search is the most widely used search algorithm.

D. Metadata

In some processing systems the significant roles are played by automatic determination for sentence, punctuation and speaker segmentation. Audio indexing and mining have enabled tracking and language identification and topic detection [12].

II. HIDDEN MARKOV MODEL SYSTEMS

Automatic Speech Recognition has traditionally leveraged the Gaussian Mixture Model-Hidden Markov Model (GMM-HMM) principle for acoustic modeling. The normalization of temporal variability is enacted by HMMs where as the emission probabilities of HMM states are computed by GMMs. With the help of Deep Neural Network (DNNs) as acoustic models, there was an improvement in the overall performance [19], where the network is used to distinguish speech frames into clustered context-dependent states. Despite these advances, building an ASR system based upon HMMs require various resources and training stages. Furthermore, to obtain the initial frame level models, training of Deep Neural Networks has a certain amount of reliability on GMM hybrid systems.

A. Hidden Markov Model

The root of all the existing speech recognition system mainly consists of group of statistical models exhibiting the different sounds of the language that has to be identified. Hidden Markov modeling is one of the methods for automatic recognition of spoken utterances. Speech has temporal structure and can be concealed as a chain of spectral vectors covering a wide range of audio frequency. Hence Hidden Markov Model (HMM) provides a natural framework for building such models [30].

The primary components of such a Automatic Speech Recognition system are illustrated in Figure 1. The main goal of the feature extraction process is to convert the speech (input) into a sequence of fixed size acoustic vectors $Y = y_1, \ldots, y_T$ [10]. The decoder then attempts to find the sequence of words $w = w_1,\ldots, w_L$ which is most likely to generate $Y$ [10].

![Fig. 1: Architecture of HMM-based Recognizer](image)

Amongst the acoustic models, HMMs are the most common one among many acoustic models (segmental models, super-segmental models, maximum entropy models) [10]. Acoustic modeling of speech establishes statistical representations for the feature vector sequences computed from speech waveform. Acoustic modeling also encircles "pronunciation modeling" i.e. the representation of larger speech units (words or phrases) which are also the object of speech recognition are used by sequence of fundamental speech units (phonetic features).

B. Gaussian Mixture Model Hybrids

An HMM is a statistical modeling technique which contains doubly stochastic process that cannot be observed (hidden) [1]. It is only observable via another set of stochastic processes that creates the sequence of observed symbols. Hence, GMM and HMM captures spatial variations and temporal variations respectively [1].

GMM-HMM based systems can be ascribed to the following: (a) Acoustic variations of the speech can be taken care of its stochastic modeling, (b) It is a system with a high computational efficiency and (c) Effective handling of time sequences. But, there are number of limitations in HMM
speech modeling such as (a) Maximum likelihood criterion lacks discriminative power, (b) The prevention of HMM from using it to its fullest advantage of the correlation that exists amidst the phone segment frames can be prevented by making the conditional independence assumptions, (c) The incorporation of contextual information into HMM systems and (d) Due to their Markovian nature, they do not take into account the sequence of states leading into any given state [4].

Temporal data as a sequence of states are modelled by HMM. These states are generally defined as separate GMMs and with the help of transition matrix, their usage of time is well directed. With the help of training data, the transition matrices are learned and it defines the probability of moving from one state to another. Conclusively, what the HMM does is to create a sequence of GMM models to explain input data (sensitive to temporal changes) [7]. By Maximum Likelihood Estimation (MLE) the parameters of acoustic model in HMM are estimated. However, the main drawback of MLE is that it cannot directly optimize word or phone recognition error rates due to its training data and model correctness [25].

1) Gaussian Mixture Models: A Gaussian Mixture Model is a probabilistic model that infers that the generation of all data points are from the combination of a finite number of Gaussian distributions with unknown parameters. According to Parham Zolfaghari and Tony Robinson [34], "parametric probability density function is defined as a weighted sum of Gaussian component densities" [25]. The parameters of GMM are iteratively approximated using the EM algorithm. The target mixtures are estimated through posterior probabilities and with the help of resulting values the calculation of the Gaussian component parameters for HMM based classifiers are derived.

While working with mixture model, the type of function that provides a better adaptation to the data field is the most prominent task to be chosen. Among all, the commonly used mixture based clustering model is the Gaussian Mixture models. GMMs are widely used in ASR systems due to its flexibility and potential of characterizing a large class of sample distributions. The main advantage of using this model is (a) It provides a combination of (flexible) non-parametric methods with (robust) parametric Gaussian model and (b) Its potential to yield smooth approximations to arbitrarily shaped densities. Techniques which involve GMM are used widely in many different tasks such as image segmentation, identification of speaker, speech recognition and image color and texture verification through biometric systems.

The computational overhead is increased by GMMs especially when making use of log-arithmetic, a series of log-additions which are needed for the GMM likelihood computation. Only those components are included which provides some improvement to the overall likelihood. Furthermore, to decrease a larger computational load, GMM likelihood can be approximated simply by maximum over all the components (weighted by the priors) [10].

C. Deep Neural Network Hybrids

Speech recognition system traditionally uses HMM to handle temporal variability of speech. GMMs are used to make sure that how good each state of each HMM fits a frame of coefficients that represent acoustic input. Alternatively, by using feed-forward neural network we can evaluate several frames of coefficients as input and deliver posterior probabilities over HMM states as output [18].

DNNs are more robust to the variations of the speaker than other models. But to small perturbations in the input features they are less sensitive in nature. These properties enable DNNs to generalize better than shallow networks and enable DNN-HMM hybrid systems to perform speech recognition in a manner that is more robust to mismatches in environment, speaker or bandwidth [33].

1) Deep Neural Networks: A Deep Neural Network is a feedforward, artificial neural network where between its inputs and outputs there exists one or more hidden units. With the help of logistic function, each hidden unit maps its inputs from the layer below, $x_j$, to the scalar state, $y_j$ sends to the layer above [18].

$$y_j = \text{logistic}(x_j) = \frac{1}{1 + e^{-x_j}}$$

$$x_j = b_j + \sum_i y_i w_{ij}$$

where $b_j$ is the bias of unit $j$, $i$ is an index over units in the layer below, and $w_{ij}$ is the weight on a connection to unit $j$ from unit $i$ in the layer below [18].

DNNs can be discriminatively trained by back propagating derivatives of a cost function that measures the discrepancy between the target outputs and the actual outputs produced for each training case [18].

D. Training of a DNN-HMM Based System

The neural network in a hybrid ASR system is trained with a two-stage training procedure.

1) DBN Pre-Training: In the first stage (Deep Belief Network Pre-Training ), layers of feature detectors are initialized by fitting a heap of generative models.

The training of the models takes place without any prior information in regards to the HMM states that the acoustic model will need to discriminate. Hence to make the rapid progress, generative pre-training searches a region or an area of the white-space that grants the discriminative fine-tuning to make an instant advancement [18].

2) DNN Fine-tuning: After the pre-training, each generative model in the stack is used in the second stage to introduce one layer of hidden units in DNN and the entire network is later discriminatively fine-tuned to forecast the HMM states tagets. To establish a forceful alignment, with the help of using a baseline GMM-HMM system the required target can be accomplished.

III. RECURRENT NEURAL NETWORK SYSTEMS

Recurrent neural networks enabled to build end-to-end ASR systems, i.e. without any intermediate components mapping can be modelled between speech and labels (phonemes, words, etc) [25] [16]. Apart from the knowledge of data, they require
no previous knowledge which is beyond input and output representation. Training of network can take place discriminatively with the internal state providing a efficient, general mechanism for modeling of time series \( \text{[25]} \text{[16]} \). Additionally, they are usually robust to spatial and temporal noise.

A. Recurrent Neural Networks

In a class of artificial neural network, Recurrent Neural Network (RNN) is a process where connections between units results in a directed cycle. This leads to the creation of internal state of the network that allows it to demonstrate dynamic temporal behaviour. RNNs, with the help of internal memory are capable of processing arbitrary sequences of inputs.

Given an input sequence \( x = (x_1, ..., x_T) \), a standard RNN computes the hidden vector sequence \( h = (h_1, ..., h_T) \) and output vector sequence \( y = (y_1, ..., y_T) \).

Training of RNNs can be done using Back-Propagation Through Time (BPTT). In practice, training RNNs to learn long-term temporal dependency can be difficult due to the vanishing gradient problem \( \text{[6]} \). This issue can be solved by using so called Long Short-Term Memory (LSTM) networks \( \text{[20]} \).

RNNs training can be achieved through Back-Propagation Through Time (BPTT). Due to vanishing gradient problem training of RNNs to learn long-term temporal dependency can be very difficult \( \text{[6]} \). However this problem can be solved by using Long Short-Term Memory (LSTM) networks \( \text{[20]} \). LSTMs employ purpose-built memory cells with self connections to keep the temporal states of the network. LSTM RNNs avoid back dissemination errors from disappearing and breaking out. Instead, errors can run backwards across infinite number of virtual layers, providing the network to learn deep learning tasks that needs memories of events that took place many time steps ago. Hence when compared to traditional Recurrent Neural Networks, LSTM architecture is always better in searching and exploiting long range context.

Another limitation of conversational RNNs is that they can use only the previous context but in ASR related tasks where the transcription of the whole utterances takes place by which the future context as well can be exploited. Thus the processing of the data bidirectionally can be possible with the help of Bidirectional Recurrent Neural Networks which inturn are processed forward to the same output. Long range context can be achieved for both the input directions through the combination of BRNNs and LSTM (BD-LSTM) \( \text{[13]} \text{[15]} \).

The usage of deep architectures which helps in constructing the higher level representations of acoustic data is the key point in the success of hybrid HMM-DNN. RNNs architecture can be achieved by building many RNN hidden layer by building one on top of the other where in the sequence of output of one layer acts as the input sequence for the next layer.

Hence by stacking up many layers of LSTMs we can create a Deep Bidirectional Short-Term Memory Network by reinstating every hidden sequence by sequences of forward and backward layers. Also, it make sure that all the hidden layers receives the input from both the layers (forward and backward) at the below level \( \text{[15]} \).

1) Long Short-Term Memory: LSTMs are a unique RNN architecture that are constructed to model temporal sequences. LSTMs, when compared to traditional RNNs their accuracy for long range dependencies are high \( \text{[20]} \). They are designed specifically to evade the long-term dependency problem and have gating mechanism instead of single neural network layer associating in a special way. Figure \( \text{[2]} \) illustrates a single LSTM memory cell.

The functionality of Gates are to let information pass through. They are formed out of a sigmoid neural net layer and a point-wise multiplication operation. The output of the sigmoid layer is always numbered between zero and one, specifying the amount of each component that needs to be let through \( \text{[15]} \). Preventing the information flow is associated with the value ‘zero’ and letting all the information through is associated with a ‘one’. There are three gates in a traditional LSTM; namely “input gate”, “output gate” and “forget gate”. The flow of the values into the next gate is blocked when the output of the input gate is close to zero. If the forget gate outputs the value close to zero, then the LSTM will forget whatever value it was remembering. And, finally when the LSTM unit should output the value in its memory is determined by the output gate \( \text{[15]} \).

B. Network training

In ASR, Neural Networks are usually trained as frame-level classifiers. Thus it necessitates that every frame has to be trained specifically by which the synchronisation between the audio and transcription sequences can be achieved by the HMM. However the high error susceptible synchronisation is irrelevant for the majority of the tasks related to ASR where the only thing that matters is the word-level transcriptions. The main focus is on end-to-end training with RNNs for the mapping of acoustic to phonetic sequences \( \text{[15]} \). Thus it helps in elimination for the requirement of predefined alignment for the creation of training targets where the network is trained without the requirement of phonetic representation but directly on the text transcripts.
1) **Connectionist Temporal Classification (CTC)** [16] is an objective function that permits an RNN to be trained for sequence transcription tasks without the need of any prior alignment among the input and target sequences. It adds as a catalyst in maximizing the log probability of getting the sequence transcription completely right. With the help of softmax layer, CTC defines a separate output distribution at every time step ‘t’ along the input sequence. This distribution covers ‘k’ targets (e.g., phonemes or characters) plus an extra blank symbol which represents a non-output [16]. Furthermore, with the help of forward-backward algorithm it sums up over all possible alignments and determine the normalized probability of the target sequence given the input sequence [16].

RNNs trained with CTC are build with bi-LSTMs as they yield better performance compared to using plain LSTMs. This is due the fact that the output of the network is then dependent on past and future observations at any time point and every probability depends on the entire input sequence. This constrains the usage of the recognition system to a so called ‘offline speech recognition’, where the whole sequence is available in advance. Real time - or online - speech recognition on the other hand is dealing with a stream of audio input where no future observations are available.

C. Decoding

Decoding a CTC network can be done by choosing the most likely output at every time step and giving back the corresponding transcription. This approach is called an acoustic only model, as CTC is distributed over the phoneme or character sequence that relies only on the input sequence of an acoustic. However, with the help of beam search algorithm an accurate decoding can be achieved which in turn helps in integrating into a language model. The algorithm differs at a marginal level because of its network output interpretation. In hybrid system there are combination of different interpretations of network outputs, i.e the posterior probabilities of state occupancies and the transitional probabilities (provided by language model and HMM) which are then combined together. However, in CTC the transition probabilities are the representations of the network outputs [14].

D. Decoding with a Language Model

The integration of a language model during decoding is further investigated in [14], [24], [25]. The researchers achieved better word error rates by constraining the search paths with various models.

Maas et al. used a neural network character language model, and trained and decoded their system by analysing the characters. This eliminated the necessity of having a lexicon, and facilitated transcription of new words and fragments [24].

Miao et al. on the other hand used a decoding approach which is generic and based on Weighted finite-state transducers (WFST) [25]. This approach facilitated the researchers to assimilate into the CTC decoding of lexions and the language models.

### IV. EXPERIMENTS

This section presents multiple experiments comparing HMM based and RNN based ASR systems conducted on two speech corpora: The Wall Street Journal and the Switchboard corpus. Apart from the different architectures and setups used by the researchers for their recognition systems, the varying sizes and qualities of these corpora led to different word error rates.

#### A. Wall Street Journal 1 (Graves et al.)

Following are the results of the experiments conducted by Graves et al. [14]. The experiments were performed based on the Wall Street Journal corpus "LDC93S6B" "LDC94S13B". The 14 hour subset ‘train-si84’ and the full 81 hour set were used by RNN, the development set used for the validation is ‘testdev93’ [14]. 43 characters including the uppercase, punctuation and space character trained with CTC were the targets for the training. The spectograms were used as the form of inputs which are acquired by the raw audio files [14].

<table>
<thead>
<tr>
<th>Network LM</th>
<th>TL 14 Hr</th>
<th>TL 81 Hr</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNN-CTC None</td>
<td>74.2</td>
<td>30.1</td>
</tr>
<tr>
<td>RNN-CTC Dictionary</td>
<td>69.2</td>
<td>24.0</td>
</tr>
<tr>
<td>RNN-CTC Dictionary</td>
<td>25.8</td>
<td>15.8</td>
</tr>
<tr>
<td>RNN-CTC Bigram</td>
<td>15.5</td>
<td>10.4</td>
</tr>
<tr>
<td>RNN-CTC Bigram</td>
<td>13.5</td>
<td>8.7</td>
</tr>
<tr>
<td>Baseline None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Baseline Dictionary</td>
<td>56.1</td>
<td>51.1</td>
</tr>
<tr>
<td>Baseline Monogram</td>
<td>23.4</td>
<td>19.9</td>
</tr>
<tr>
<td>Baseline Bigram</td>
<td>11.6</td>
<td>9.4</td>
</tr>
<tr>
<td>Baseline Trigram</td>
<td>9.4</td>
<td>7.8</td>
</tr>
</tbody>
</table>

#### B. Wall Street Journal 2 (Maas et al.)

For the experiments conducted by Maas et al. [16], the research team used a neural network character language model, and trained and decoded their system by analysing the characters. This eliminated the necessity of having a lexicon, and facilitated transcription of new words and fragments [24].

<table>
<thead>
<tr>
<th>Network LM</th>
<th>TL 14 Hr</th>
<th>TL 81 Hr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Baseline Dictionary</td>
<td>56.1</td>
<td>51.1</td>
</tr>
<tr>
<td>Baseline Monogram</td>
<td>23.4</td>
<td>19.9</td>
</tr>
<tr>
<td>Baseline Bigram</td>
<td>11.6</td>
<td>9.4</td>
</tr>
<tr>
<td>Baseline Trigram</td>
<td>9.4</td>
<td>7.8</td>
</tr>
</tbody>
</table>
B. Wall Street Journal 2 (Hannun et al.)

Hannun et al. [17] performed their experiments on the WSJ corpus (LDC93S6B and LDC94S13B) too, but achieved error rates not as good as the ones of Graves et al. The ‘dev93’ evaluation subset was used as a development set and final test set performance was reported on the ‘eval92’ evaluation subset.

1) RNN System: The underlying network was a Bidirectional Recursive Deep Neural Network (BRDNN) without LSTM cells. The network had five levels of hidden layers, with 1824 cells in each layer, giving 20.9M free parameters. The third hidden layer had recurrent connections.

The authors used the Nesterov accelerated gradient optimization algorithm [31] for training with an initial learning rate of 0.00001 and a maximum momentum of 0.95. After each epoch the learning rate was divided by 1.2.

2) Baseline System: No HMM baseline system was used for comparison.

3) Results: Table II provides the results of the experiment conducted by Hannun et al. on The Wall Street Journal corpus. The Character Error Rate and Word Error Rate results from a BDRNN (Bi-Directional Recurrent Deep Neural Networks) systems are depicted.

Furthermore, Table II shows word and character error rates for approaches with different language models. Just like Graves et al. reported already, the error rates are getting better as the language model is strengthened. One thing to note here is that the system already performs good on character recognition without any language model at all.

C. Wall Street Journal 3 (Miao et al.)

In context of the project of Miao et al. called Eesen, the authors conducted also experiments on the WSJ corpus (LDC93S6B and LDC94S13B) [25]. Results were reported on the ‘eval92’ set.

1) RNN System: The underlying recurrent network had 4 BD-LSTM hidden layers, with 320 cells in each layer. The RNNs input were 40-dimensional filterbank features along with their first and second-order derivatives [25].

2) Baseline System: The baseline system was a hybrid DNN-HMM system, built with the Kaldi recipe ‘s5’. DNN models input were 11 neighboring frames of filterbank features. The network consists 6 hidden layers and 1024 units at every layer. Table III shows the results of the phoneme-based Eesen system compared to the hybrid DNN-HMM system [25].

3) Results: Table III lists the two systems trained with a trigram language model, while the results of the RNN trained with only a lexicon was also reported. The WER of Eesen’s RNN with a trigram language model is nearly on par with the error rates of the baseline state-of-the-art hybrid HMM system [25]. The network is even better performing compared to the two networks in the previous experiments.

As a major advantage of their system Miao et al. point out the decoding speed, which is 3.2x faster than that of DNN-HMM systems. The decoding times, together with the decoding graph size is listed in Table IV. It shows the decoding speed comparison among the phoneme-based Essen system and the DNN-HMM system. real-time factor in decoding and the size of the decoding graph (in MegaBytes) is represented by ‘RTF’ and ‘Graph Size’ respectively [25].

D. Switchboard (Maas et al.)

Maas et al. [24] conducted another experiment with recurrent neural networks trained with CTC by integrating a character based language model in the decoding process. The experiment was performed on a telephone speech corpus with 300 hours of data (LDC97S62). The utterances were taken from 2,400 conversations among 543 speakers. The WER and CER were reported on the HUB5 Eval2000 (LDC2002S09). Table V depicts the results for Dependent Bidirectional Recurrent Neural Networks and baseline systems.

1) RNN System: The system considered for the experiment allows training of utterances that are tagged by word-level transcriptions and consists of absolutely no information regarding when the words could appear within the utterances. During the beam search decoding policy the two neural networks are being integrated into this approach. At each time stamp the probability distribution over characters are being mapped by the features of an acoustic inputs in the first neural network. The component of the second neural network is the neural network character language model [24].

The Dependent Bidirectional Recurrent Neural Network consists of 5 hidden layers out of which the third layer comprises of recurrent connections. There are 1824 units that exists in all layers providing around 20 million parameters for training [24].

### Table II: Experiment Results on the Wall Street Journal Corpus of Hannun et al. [17]

<table>
<thead>
<tr>
<th>Model</th>
<th>CER</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>No LM</td>
<td>10.0</td>
<td>35.8</td>
</tr>
<tr>
<td>Dictionary LM</td>
<td>8.5</td>
<td>24.4</td>
</tr>
<tr>
<td>Bigram LM</td>
<td>5.7</td>
<td>14.1</td>
</tr>
</tbody>
</table>

### Table III: Performance of the Phoneme-Based Eesen System Compared with the Hybrid DNN-HMM Baseline System. ‘#Param’ Means the Number of Parameters [25].

<table>
<thead>
<tr>
<th>Model</th>
<th>LM</th>
<th>#Param</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eesen RNN</td>
<td>Lexicon</td>
<td>8.5M</td>
<td>26.92</td>
</tr>
<tr>
<td>Eesen RNN</td>
<td>Trigram</td>
<td>8.5M</td>
<td>7.87</td>
</tr>
<tr>
<td>Hybrid DNN-HMM</td>
<td>Trigram</td>
<td>9.2M</td>
<td>7.14</td>
</tr>
</tbody>
</table>

### Table IV: Comparison of Decoding Speed Between the Phoneme-Based Eesen System and the DNN-HMM System [25].

<table>
<thead>
<tr>
<th>Model</th>
<th>RTF</th>
<th>Graph Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eesen RNN</td>
<td>0.64</td>
<td>263</td>
</tr>
<tr>
<td>Hybrid DNN-HMM</td>
<td>2.06</td>
<td>480</td>
</tr>
</tbody>
</table>
TABLE V: BASED ON THE EVAL2000 TEST SET THE RESULTS OF CHARACTER ERROR RATE (CER) AND WORD ERROR RATE FROM MAAS ET AL. [24]

<table>
<thead>
<tr>
<th>Network</th>
<th>CER</th>
<th>EV</th>
<th>CH</th>
<th>SWBD</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM-GMM</td>
<td>23.0</td>
<td>29.0</td>
<td>36.1</td>
<td>21.7</td>
</tr>
<tr>
<td>HMM-DNN</td>
<td>17.6</td>
<td>21.2</td>
<td>27.1</td>
<td>15.1</td>
</tr>
<tr>
<td>CTC no LM</td>
<td>27.7</td>
<td>47.1</td>
<td>56.1</td>
<td>38.0</td>
</tr>
<tr>
<td>CTC+5-gram</td>
<td>25.7</td>
<td>39.0</td>
<td>47.0</td>
<td>30.8</td>
</tr>
<tr>
<td>CTC+7-gram</td>
<td>24.7</td>
<td>35.9</td>
<td>43.8</td>
<td>27.8</td>
</tr>
<tr>
<td>CTC+NN-1</td>
<td>24.5</td>
<td>32.3</td>
<td>41.1</td>
<td>23.4</td>
</tr>
<tr>
<td>CTC+NN-3</td>
<td>24.0</td>
<td>30.9</td>
<td>39.9</td>
<td>21.8</td>
</tr>
<tr>
<td>CTC+RNN</td>
<td>24.9</td>
<td>33.0</td>
<td>41.7</td>
<td>24.2</td>
</tr>
<tr>
<td>CTC+RNN-3</td>
<td>24.7</td>
<td>30.8</td>
<td>40.2</td>
<td>21.4</td>
</tr>
</tbody>
</table>

2) Baseline System: The authors used two HMM hybrid systems to compare their approach to a traditional ASR system. The first system is basically the HMM-GMM system which uses the open-source toolkit “Kaldi” [8]. By using maximum likelihood there are 8986 sub-phone states and 200K Gaussians in the baseline recognizer. The second system is basically the HMM-DNN system which is built based on the HMM-GMM system using maximum likelihood for training a DNN acoustic model. Like the DBRNN system, the DNN system as well consists of 5 hidden layers. Each layer in turn consists of 2048 hidden units which sums up to 36 million parameters which are free in the acoustic model [24].

3) Results: With the help of the 3 hidden layer DNN CLM the performance of Dependent Bidirectional Recurrent Neural Network is always at the optimal best. Although there is not much of a huge advantage of RNN CLM when compared to the best of DNN CLM, however when compared to a traditional n-gram model for DBRNN decoding, Neural network is the winner. Table V shows all results of the different networks with different language models.

Based on the full test set (EV) that contains switchboard (SWBD) and CallHome (CH) subsets, the Word Error Rates were reported [24]. The authors used an HMM-GMM and an HMM-DNN system as base-line systems. Against various character-level language models (such as 5-gram, 7-gram, densely connected neural networks with 1 and 3 hidden layers (NN-1 and NN-3) and Recurrent Neural Networks with 1 and 3 hidden layers) they evaluated their Dependent Bidirectional Recurrent Neural Network [24].

It is understood that the major Word Error Rate differences are translated from minor Character Error Rate differences. This model provides a complete LVCSR system with roughly around a 1000 lines of code which is of a lesser magnitude in comparison of of HMM-GMM systems yielding a better and high performance [24].

V. CONCLUSION

In this paper, we discuss the two main approaches of Automatic Speech Recognition systems. We start with the traditional Hidden Markov Model based systems which were the state-of-the-art for many years. The progress from Gaussian Mixture Model hybrids to Deep Neural Network hybrids is explained. Even with this progress, building an ASR system based on HMMs remained a challenging task requiring multiple resources and training stages. We proceed with explaining the new end-to-end Recurrent Neural Network based recognition systems and their details.

The survey shows, that RNN-CTC systems give better results when compared with DNN-HMM ones. As always, having more training data can make the difference. The RNN-CTC acoustic models are much faster at decoding and much simpler to train - no more bootstrapping of traditional DNN-HMM models. Especially faster decoding time can be very appealing in practical settings, even if there is a decrease in the accuracy.

REFERENCES


