RC22185 (W0109-057) September 25, 2001
Computer Science

IBM Research Report

Current State of the Art in Large Vocabulary Automatic Speech Recognition Algorithms

Mukund Padmanabhan, Michael A. Picheny
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Delhi - Haifa - India - T. J. Watson - Tokyo - Zurich

Current State of the Art in Large Vocabulary Automatic Speech Recognition Algorithms

Mukund Padmanabhan and Michael Picheny
mukund@us.ibm.com, picheny@us.ibm.com

Abstract—
Large vocabulary continuous speech recognition has progressed enormously over the last three decades. There are now practical commercial systems available capable of accurately recognizing vocabularies of thousands of words from cooperative talkers. However, there are still many applications of speech recognition beyond the capabilities of today's recognizers, including, for example, recognition of speech in conversations and meetings. This article describes some of the recent advances in speech recognition technology that are making substantial inroads in handling such challenging problems.

I. Introduction

Natural interfaces to computers have been one of the main focus areas of research in computer science for almost forty years. One of the key enablers of such interfaces would be to give the computer the ability to understand human speech. Efforts in speech recognition research started more than three decades ago with an attempt to decode ***isolated*** words from a small *vocabulary*. When performance on this task improved to the point of being usable, the bar was raised and the research community started to work on large vocabulary and continuous speech tasks. However, the development of robust ***automatic speech recognition*** (ASR) systems has proven to be one of the most difficult tasks in the computer science field, and it is only over the last few years that practical large vocabulary continuous speech recognition systems have progressed to the stage of being moderately usable and commercially successful. However, these commercial applications are still restricted to very limited domains, and either restrict the vocabulary to a few thousand words (in the case of name dialing or airline reservation systems), or require high-bandwidth high-feedback situations, such as dictation, in which the user modifies his/her own speech to minimize recognition errors effectively.

The main source of advancements in speech recognition over the years has been the development of a set of statistical modeling techniques that can be trained from hundreds of hours of speech, combined with a massive increase of available computing power allowing speech researchers to explore complex models with hundreds of thousands of parameters. Early attempts at speech recognition tried to apply expert knowledge about speech production and perception processes, but such knowledge was sadly found to be inadequate to capture the complexities of continuous speech. Researchers discovered that it was more effective to apply non-parametric models with trainable parameters and refine these parameters by expanding their complexity and exposing them to progressively more and more data. Since computing speed more than kept pace with the ability to intelligently add more parameters to these models, the state of the art in speech recognition rapidly progressed.

Another reason for advancement has been multi-site speech recognition research cooperation and competition supported through government agencies such as DARPA. By funding yearly competitions across sites and requiring industrial labs (IBM, AT&T, BBN, SRI, etc.) universities (Cambridge University, Johns Hopkins, etc.) and other companies (Dragon Systems, etc.) to share data and algorithms, rapid advances in the state of the art of speech recognition have been maintained [1]. For better or for worse, a by-product has been that almost all successful systems have roughly the same architecture and algorithms, since each site would immediately copy other site's successful algorithms.

In order to enable the next generation of applications, such as speech recognition over cellular phones, transcription of conversations over telephones between people, and recognition of broadcast news, researchers continue to work on the ASR problem, and several recent advances have taken place in the field motivated by these new applications. The rest of this paper first describes the basic architecture of most ASR systems in use today, and some of these recent algorithmic advances.

II. Basic Architecture of Today's Recognizer

The process of speech recognition starts with a sampled speech signal (sampled at 16KHz, for example). There is a good deal of redundancy in this speech signal because the articulators that produce the speech (glottis, tongue, lips, etc.) are physcially constrained and cannot move very quickly. Consequently, the first operation that is carried out on the signal is to extract a smaller number of ***acoustic feature vectors*** from the signal (typically, a single multi-dimensional feature vector is extracted every 10 ms). These feature vectors contain information about the local frequency content in the speech signal, and are referred to as the acoustic observations because they represent the quantities that are actually observed by the ASR system. The goal of the ASR system is now to infer the word sequence that was spoken, that could have produced the observed acoustic sequence.

A simplifying assumption that is made in ASR systems is that the vocabulary of the speaker is known. Consequently, the search for possible word sequences is restricted to the words that are in the vocabulary (referred to as the *lexicon*). The lexicon lists all the words in the vocabulary,

as well as the pronunication of each word in terms of a set of basic units, usually individual speech sounds, called *phonemes*. Typical lexicons of commercial systems today include tens of thousands of words. Further, the length of the word sequence is not known. Let us assume for a moment that this length is N – if V represents the size of the lexicon, then the ASR system can hypothesize V^N possible word sequences. Just based on the constraints of the language, not all of these word sequences are equally likely in the first place (for instance, the word sequence "give me a call" is much more likely than "give call a me"). Further, we also have the benefit of having observed the acoustic feature vectors corresponding to the speech that was produced by the word sequence. The utility of these acoustic observations lies in the fact that they give a clue as to the phoneme that produced them. If we know the sequence of phonemes corresponding to the acoustic observations, then it is possible to infer the word sequence that could have produced the sequence of sounds. Consequently, the acoustic observations are an important source of information that can help to further narrow the space of possible word sequences. Armed with these pieces of information, we can now assign a probability that the observed acoustic feature vectors were produced by a particular word sequence. The job of the ASR system is essentially to compute these probabilities in an efficient manner, and then to output the most probable sequence as the *decoded hypothesis*.

This sequence of steps is encapsulated in a probabilistic model that is used by all of today's most successful speech recognition systems. In this model, the goal of the recognizer is to find the word sequence $\hat{w}_1^N$ that maximizes the probability of the word sequence given some observed acoustic sequence y_1^T. This can be interpreted as trying to maximize the product of two terms, the probability of the acoustic observations given the word sequence, and the probability of the word sequence itself:

$$\hat{w_1^N} = \arg\max_{w_1^N} p(w_1^N|y_1^T) \equiv \arg\max_{w_1^N} p(y_1^T|w_1^N)p(w_1^N) \quad (1)$$

This formulation suggests a basic structure to speech recognition systems, depicted in the block diagram of Fig. 1. In Block 1, multi-dimensional features y_1^T are extracted from the sampled speech signal; in Block 5, a search procedure hypothesizes a word sequence w_1^N that maximizes the probability $p(w_1^N|y_1^T)$. This step is driven by several components (i) a lexicon (Block 2) that defines the possible words that can be hypothesized in the search, where each word is represented as a linear sequence of phonemes, (ii) a language model (Block 3) that models the linguistic structure but does not contain any knowledge about the relationship between the feature vectors and the words, (iii) an acoustic model (Block 4) that models the relationship between the feature vectors and the phonemes. In the following subsections, we will describe these building blocks in more detail.

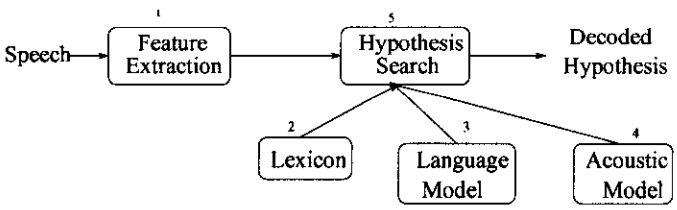

Fig. 1. Speech Recognition System Block Diagram

A. Feature Extraction

The most commonly used feature extraction schemes extract a multi-dimensional feature vector from the sampled speech signal at a uniform frame rate (typically every 10 ms). These feature vectors are supposed to provide clues about the phonemes that produced them - consequently, the procedure for extracting them is motivated by the workings of the human auditory system, which is able to infer the phoneme after listening to the speech signal. As the human auditory system simulates a constant-Q filterbank [1] with the sensitivity to the energy in each channel following a logarithmic relationship, most feature extraction schemes are based on mimicking these steps. Often, the temporal derivatives and second derivatives, of these features are also computed, and appended to form the final feature vector. The reasoning here is that the temporal variation of the local spectral content of the speech signal also contain a good deal of information to help infer the spoken phonemes.

B. Lexicon

A typical lexicon is shown in Table I. The lexicon lists all the words in the vocabulary, as well as the pronunication of each word in terms of a set of basic units called phonemes. There are approximately 50 of these phonemes that are used in English. Note that an individual word may have multiple pronunciations. The choice of lexicon is usually made on a task-dependent basis by choosing a set of words that result in fewer than 3% out-of-vocabulary words. The basic representation units can also be automatically derived when either the vocabulary is small or there is a substantial amount of training data available. Phonetic representations for words in common usage are easy to find in various sources; task dependent jargon often requires pronunciations to be written out by hand.

C. Language Model

The search for the most likely word sequence in (1) requires the computation of two terms, $p(y_1^T|w_1^N)$ and $p(w_1^N)$. The second of these computations is carried out by a *Language Model* (Block 3 of Fig. 1). The function of the language model is to assign a probability to a sequence of words w_1^N. The simplest way to estimate such a probability would be to compute the relative frequencies of different sequences of words; however, the number of different

[1] In a constant-Q filter-bank, the ratio of the center frequencies of adjacent filters in the filter-bank is constant, and the bandwidths of the filter are proportional to the center frequencies.

Word	Phonetic Representation
the	DH AH
the	DH IY
cat	K AE T
pig	P IH G
two	T UW

TABLE I
TYPICAL LEXICON. NOTE THE WORD **the** HAS TWO PRONUNCIATIONS

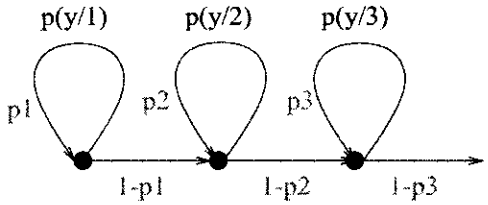

Fig. 2. Hidden Markov Model for a Phoneme

sequences of words grows exponentially with the length of the sequence, making this infeasible. A typical approximation is to assume the probability of the current word only depends on the previous two words, so that the probability of the word sequence can be approximated as:

$$p(w_1^n) \approx p(w_1)p(w_2|w_1) \prod_{i=3}^{i=N} p(w_i|w_{i-1}, w_{i-2}) \quad (2)$$

$p(w_i|w_{i-1}, w_{i-2})$ can be estimated by counting relative frequencies of trigrams (i.e., triplets) of words:

$$p(w_i|w_{i-1}, w_{i-2}) \approx N(w_i, w_{i-1}, w_{i-2})/N(w_{i-1}, w_{i-2}) \quad (3)$$

where N refers to the relative frequency of the associated event. Typically to train such a language model, hundreds of millions of words are used to estimate $p(w_i|w_{i-1}, w_{i-2})$ and even then, many trigrams do not occur in the training text. Hence, the probability estimates must be smoothed to avoid zeroes in the probability assignment. A good review of estimation techniques for the language model parameters is given in [2].

D. Acoustic Model

The search for the most likely word sequence in (1) requires the computation of two terms, $p(y_1^T|w_1^N)$ and $p(w_1^N)$. The first of these computations is carried out by an *Acoustic Model* (Block 4 of Fig. 1). The function of the acoustic model is to compute the probability of sequences of feature vectors under the assumption that they were produced by a particular word sequence. As speech is an inherently stochastic process, no word is ever uttered the same way twice. The variation in the pronunciation of a word or phoneme manifests itself in two ways: the duration, and the spectral content (acoustic observations). Further, the spectral content variation for a particular phoneme may sometimes be systematically attributed to the effect of the phonemes in the surrounding context. This phenomenon is called *coarticulation*. The acoustic model needs to be able to compute a probability for an observed sequence of feature vectors while taking all of these inherent stochastic variations into account.

D.1 Hidden Markov Models

A natural choice of models that can model the stochastic aspects of speech is a *Hidden Markov Model* (HMM). Fig. 2 is an example of a HMM for a phoneme. HMM's are probabilistic finite state machines - the model comprises of a set of states, and its topology specifies the allowed transitions between states. At every time frame, a transition is made probabilistically from one state to another and the model emits a feature vector every time a transition is made. The transitions that can be made between the states are governed by a set of *state transition probabilities* p_1, p_2, p_3, which specify the probability of going from one state at time t to another state at time $t+1$. The feature vectors emitted while making a particular transition represent the spectral characteristics of the speech at that time frame, which vary corresponding to different ways of pronuncing the phoneme. This variation is modeled by a *probability distribution* or *probability density function* (pdf). The pdf's ($p(y/1), p(y/2)$ and $p(y/3)$) could be different for different transitions, as in Fig. 2. These pdf's are often represented as parametric distributions (for example, a mixture of multi-dimensional gaussians).

The HMM shown in Fig. 2 comprises of 3 states, and the pronunciation of the phoneme corresponds to starting from the first state, 1, and making a sequence of transitions to eventually arrive at the last state, 3. The duration of the phoneme is the number of time frames required to make the sequence of transitions. This duration is governed by a probability distribution which is implicitly specified by the transition probabilities p_1, p_2, p_3. If p_1, p_2, or p_3, the *self-loop* probabilities, are high, then the model spends more time in the same state - consequently, it takes longer to go from the state 1 to state 3. The sequence of output feature vectors that are produced in making this sequence of transitions are governed by the pdf's associated with the transitions, $p(y/1), p(y/2)$ and $p(y/3)$.

A fundamental operation that is repeated over and over in the process of building and using ASR systems is the computation of the likelihood of a sequence of acoustic feature vectors using an HMM. For instance, assume that T feature vectors were extracted from speech corresponding to the pronunciation of a single phoneme, and the objective is to infer which one of the set of 50 phonemes was spoken. The procedure for inferring the phoneme assumes that the i^{th} phoneme was spoken, and finds the likelihood that the HMM model for this phoneme produced the observed feature vectors; subsequently, the phoneme model that gives the highest likelihood to the sequence of feature vectors is hypothesized to have been spoken.

The probability of a sequence of feature vectors may be easily computed *if* the sequence of states through the

HMM, that produced these feature vectors, is known. In this case, the likelihood of the t^{th} feature vector, y_t, is computed using the probability density function for the HMM state at time t. The likelihood of the complete set of T feature vectors would simply be the product of all these individual likelihoods. However, in general the actual sequence of transitions (states) is not known. Consequently, the likelihood computation process sums over all possible state sequences. It turns out that as all dependencies in the HMM are local, it is possible to derive computationally efficient formulae to perform these summations recursively [2].

D.2 Estimation of Parameters

The HMM described in the previous subsection is parametrized by the *state transition probabilities* and the *means, covariances* and *priors* of the gaussians that model the pdf's of the feature vectors for the different states of the HMM. Before using the HMM to compute the likelihoods of sequences of feature vectors, it is necessary to estimate the parameters of the model. This process is called *training* of the HMM models, and it assumes that a large amount of training data is available. This training data comprises of the feature vectors extracted from the speech signal, *along with* the word sequence that the speaker utters.

A commonly used training paradigm is *Maximum Likelihood* (ML) estimation. As the correct word sequence corresponding to the sequence of feature vectors is known, the ML estimation process tries to choose the parmeters of the HMM to maximize the likelihood of the training feature vectors, computed using the HMM model for the correct word sequence. If y_1^T represents the stream of T acoustic observations, and w_1^N represents the correct word sequence, the ML estimate for the parameters θ is

$$\hat{\theta}_{ML} = \arg\max_{\theta} \log\left[p_\theta(y_1^T/w_1^N)\right] \tag{4}$$

The first step in the training process is to construct a HMM model for the correct word sequence. First the HMM models for each word are constructed by concatenating the HMM models for the phonemes comprising the word. Subsequently, the word HMM models are concatenated to form the HMM model for the complete utterance. For instance, the HMM model for the utterance "The two" would be the concatenation of the HMM models for the four phonemes "DH AX T UW".

The training process assumes that the acoustic observations y_1^T are obtained by traversing the states of this HMM from the initial state of the HMM to the final state in T time frames (in the previous example, the initial state is the first state of the HMM for DH and the last state is the final state of the HMM for UW). However, as mentioned earlier, the actual state sequence is not observable, so the ML estimation process assumes that this state sequence is *hidden* and averages over all possible values of the state sequence.

Using s_t to denote the hidden state at time t, and after making various assumptions, the maximization (4) may now be expressed in terms of the hidden states of the HMM as

$$\arg\max_{\hat{\theta}} \sum_{t=1}^{T} \sum_{s_t} p_\theta(s_t/y_1^T) \log\left[p_{\hat{\theta}}(y_t/s_t)\right] \tag{5}$$

The solution of (5) is an iterative process, with each iteration involving two steps, an (E)xpectation step, and a (M)aximization step [2]. The E-step involves the computation of $p_\theta(s_t/y_1^T)$, the posterior probability, or count of a state, conditioned on all the acoustic observations. This is done using an algorithm called the *Forward-Backward* algorithm [2], which uses the current estimates of the HMM parameters. The M-step involves choosing the parameters $\hat{\theta}$ to maximize (5). For the case where the pdf's are gaussians, closed form expressions can be derived for this M-step.

D.3 Coarticulation

In earlier sections, we assumed that the fundamental acoustic units were phonemes, and HMM's were used to model the duration and acoustic variation associated with the pronunciation of the phonemes. However, the acoustic variation for a particular phoneme can sometimes be systematically analyzed as being due to the effect of the phonemes in the surrounding context. This phenomenon is called coarticulation, and is particularly prevalent in spontaneous speech where the speaker is not careful about the enunciation of his/her speech [2]. Coarticulation is modeled assuming the density of the observations depends not only on the current phoneme, but also the surrounding ones. However, modeling every phone in every possible context would lead to a prohibitively large number of densities that need to be modeled. For instance, if the phoneme alphabet comprises of 50 phonemes, and we model every phoneme in the context of its immediately surrounding neighbours, that would require the modeling of 125,000 densities. Consequently, a commonly adopted approach clusters the surrounding phonemes into a small number of equivalence classes to reduce the total number of densities that need to be modeled [2]. Typical large vocabulary speech recognition systems contain thousands of context-dependent densities capturing different coarticulatory behaviors.

E. Hypothesis Search

The purpose of the hypothesis search is to determine the word sequence with the highest likelihood given the input features and the parameters of the model [2]. Since the number of word sequences increase exponentially with the length of the word sequence, this might at first seem to be an intractable problem for anything other than short utterances from a small vocabulary. However, since all the probabilistic dependencies of the model are local, it is possible to incrementally search through the hypotheses in a left-to-right fashion discarding most of the candidates without any loss in optimality [2].

[2] An example of this phenomenon is the pronunciation of "could you" as "couldju".

The recognition search is depicted in Fig. 3 using a di-

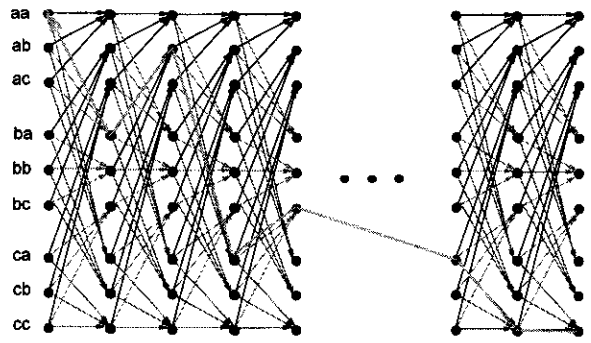

Fig. 3. Hypothesis Search for a Trigram Language Model. Language model states indicated on the left for $w_{n-1}w_{n-2}$ pairs. w_n transitions indicated by black, blue, and red for words **a**, **b** and **c**, respectively. A typical traceback is shown in green.

agram commonly known as a recognition *trellis*. In this simple example, there are three words in the vocabulary - **a**, **b**, and **c**. Each small circle represents a state of the language model. For this example, in a trigram language model there would be nine states, representing each possible combination of the two possible preceding words of the vocabulary. For each language model state, it is possible to output any of the three words in the vocabulary. Depending upon which word is output, a different set of language model states will be activated. For example, if the word **a** is assumed to be output, the states labelled *aa*, *ab* and *ac* will be activated at the next interval. It is possible at a given point in time to make a hard decision about the best path leading into a particular language model state. For example, into state *aa* are three paths, representing the sequences **aaa**, **baa** and **caa**. The local dependency of the language model allows us, to discard the two lowest scoring paths and just keep track, using a back-pointer, to the path that had the highest score. The decision about the overall best path must wait until the end of the sentence is reached and then to recursively follow the set of back pointers from the highest scoring path at the end of the sentence to determine the overall best path through the sentence. Although the number of states in the language model can theoretically grow as the square of the vocabulary, since many trigrams never actually occur in the training data, and the smoothing operation backs off to bigram and unigram estimators, substantial reduction in trellis size can result. It is also possible to speed up the search further by making additional approximations, for instance, only retaining hypotheses at any time frame that fall within a threshold of the maximum score at that time - this is referred to as a *beam search*.

F. Benchmarking the performance of ASR Systems

The performance of ASR systems is quantified by a *word eror rate (WER)* that expresses the ratio of the number of word errors to the number of words in the reference transcription as a percentage [3]. The WER of ASR systems can vary by a couple of orders of magnitude depending on the task. The main characteristics that effect the WER are

- size of vocabulary – small vocabulary tasks (eg. digits) have lower WER
- language model *perplexity* – the perplexity indicates the branching power of the language, and low perplexity tasks generally have lower WER
- background noise – the cleaner the background in which the test speech is recorded, the lower the WER
- spontaneous nature of the speech – the more spontaneous the speech, the higher the WER [4]
- sampling rate of the speech signal – the higher the sampling rate, the better the performance
- amount of training data available – the more the amount of training data, the better the performance

In order to get the best performance for a task, it is necessary to customize the ASR system to the task. The aspects of a task that affect the various components of the ASR system in Fig. 1 are summarized below:

- *feature extraction* (Block 1) – depends on the sampling rate of the speech signal
- *lexicon* (Block 2) – depends on the vocabulary that the speaker is likely to use [5]
- *language model* (Block 3) – depends on the topic and linguistic characteristic of the user's speech [6]
- *acoustic model* (Block 4) – depends on the nature of the speech, and the sampling rate [7]

In this section, we examined the basic components of an ASR system. In the last few years, several advances have been made in improving the performance of individual components. In the following section, we describe a number of these recent algorithmic advances, and quantify the improvement due to these algorithms, by reporting the improvement in WER due to each of these algorithms. We chose to report these WER results on a VOICEMAIL TRANSCRIPTION task [1] – as the name suggests, this task uses ASR systems to transcribe voicemail messages. This task represents spontaneous telephone speech (sampled at 8KHz) with a fairly large vocabulary and consequently represents one of the more challenging problems in ASR research. Though the relative improvements that we report are quantified on this task, the methods are quite

[3] The number of errors is computed by aligning the decoded word sequence with the reference word sequence, and counting all instances where the reference word is ***substituted*** by another, or is ***deleted***, or where the decoded sequence contains an extra word that is not in the reference ***(insertion)***.

[4] This is because spontaneous speech contains several characteristics such as prosodic variation, false starts, hesitations, etc. which are not adequately modeled by current ASR systems.

[5] For recognizing digit strings, the lexicon need comprise of only the digits, for recognizing radio broadcast news, the lexicon generally comprises of several tens of thousands of words.

[6] For instance, radio broadcast news and voicemail messages represent different domains, and the word sequences that the speaker is likely to utter differ drastically for the two domains. Consequently, the language model needs to be cutomized to each of these domains.

[7] For instance, the characteristics of the acoustic feature vectors obtained from speech recorded at 8KHz with noise in the background is markedly different from that of speech recorded at 16KHz in a quiet environment. Consequently, the acoustic models need to be customized depending on the background conditions, sampling rate, type of speech (spontaneous or isolated), etc.

general and applicable to any speech recognition task. The test data on which all WER numbers are reported comprises of 105 voicemail messages (52 minutes of speech), and the training data comprises of 4700 voicemail messages, (53 hours of speech). The *speaker-independent* (SI) WER on this task with a baseline system, as described in this Section, is 40.5%.

III. Recent Advances in ASR Systems

In this section, we will describe some recent advances in ASR systems that have led to significant improvements in accuracy. These improvements are not specific to the IBM ASR system, but have been incorporated in almost all commercial and laboratory systems. These advances may be broadly classified as (i) novel methods of extracting acoustic observations from the speech signal, (ii) alternatives to ML estimation of the HMM parameters, (iii) post-processing methods for hypothesizing a better sequence of words, (iv) adaptation of the acoustic models based on limited amounts of test data from the speakers, and finally (v) methods to combine the outputs of several ASR systems.

A. Feature Extraction

As described in Section II-A, temporal trajectory information is often encoded in the feature vector by augmenting it with first and second derivatives. If the extracted cepstra are d-dimensional, this leads to an acoustic feature that has dimensionality $3d$. A more sophisticated approach is to incorporate such information by means of a linear discriminant transformation. In this approach, the d-dimensional cepstra from several adjacent frames (typically 9) are concatenated together to form a $9d$-dimensional feature vector. The dimensionality of the feature vector is then reduced by computing a linear projection that finds the directions that maximally separate the phonetic classes out. The pdf of the projected feature vectors for each HMM state is then modeled by a mixture of gaussians. Further, due to computational considerations, most ASR systems assume that the covariance matrices of these gaussians are diagonal, which in effect assumes that the dimensions of the projected feature vectors are independent.

One possible method of computing the projection is the *Maximum Likelihood Discrimimant* (MLD) projection describe in [3], that simultaneously (i) maximizes the log-likelihood of the data in the projected space (ii) maximizes the separation between the class means in the projected space (iii) minimzes the correlation between the dimensions of the projected feature vectors. For further details, we refer the reader to [3]. The use of this projection reduces the WER of the baseline system in Section II-F from 40.5% to 39.1%, a relative improvement of 3.5%.

B. Acoustic Model - Estimation of Parameters

In Section II-D.2, we studied the commonly used ML approach to estimating the parameters of the models in an ASR system. ML estimation makes several assumptions which are known to be incorrect, consequently, a *Maximum Mutual Information* (MMI) estimation procedure was

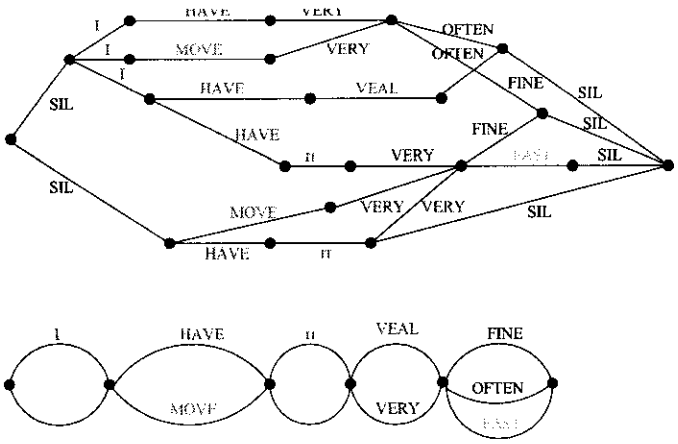

Fig. 4. Converting a word lattice to a confusion network

suggested in [4]. This estimation strategy tries to obtain estimates of the model parameters that maximize the mutual information between the correct word sequence and the observed acoustic feature vectors, i.e.,

$$\hat{\theta}_{MMI} = \arg\max_{\theta} \int_{w,y} p(w,y) \log \left[\frac{p_\theta(w,y)}{p_\theta(w) p_\theta(y)} \right] \quad (6)$$

This objective function, in addition to considering the correct word sequence for the training data, also considers all possible transcriptions of the training data, and attempts to maximize the discrimination between the correct transcription and the incorrect transcriptions. A good review of issues related to MMI estimation is given in [5].

C. Hypothesis Search

The most commonly used decoding paradigm for speech recognition is the *maximum-a-posteriori* (MAP) rule given by (1). However, an alternative scoring procedure has been proposed recently to guide the hypothesis search. If w_1^N represents the decoded word sequence, and $w_1^{'N}$ represents the correct word sequence, we may define a loss function $l(w_1^N, w_1^{'N})$ that quantifies the difference between the two word sequences, and define the objective of the decoding procedure as the minimization of the average expected loss. This may be written

$$w_1^{N*} = \overset{argmin}{w_1^{'N}} \sum_{w_1^N} l(w_1^N, w_1^{N'}) p(w_1^N | y_1^T) \quad (7)$$

If $l(w_1^N, w_1^{'N})$ is taken to be a delta function, which represents the sentence error rate, then (7) reduces to the commonly used MAP decoding rule in (1). Hence a hypothesis search guided by the the MAP decoding rule is equivalent to minimizing the sentence error rate. However, as the quantity of interest in most speech recognition applications is the word error rate, it makes more sense to define $l(w_1^N, w_1^{'N})$ as the word error rate between the two word sequences.

In [6] this idea was used to obtain a "consensus" hypothesis that minimized the loss. The procedure for finding this hypothesis first computes a word lattice (graph)

using a MAP decoder, and then converts the word lattice into a chain like structure by merging different paths in the graph. This is shown in Fig 4. The components of the chain represent parallel sequences of words. Minimizing the loss is now equivalent to just picking the most probable word in each component, and the concatenation of these words represents the consensus hypothesis. The use of the consensus search technique reduces the WER from 39.1% (see Section III-A) to 38.0%, a relative improvement of 2.8%.

D. Acoustic Model - Adaptation

It is well known that the best performance is generally obtained for *speaker dependent* (SD) ASR systems, in which the system is trained on speech from a single speaker. In order to build such a model, a large amount of training data is required from the specific speaker, which is usually impractical to obtain. It often is possible to collect a small amount of speech data (few minutes) which can be used to either adapt the parameters of a SI system to the speaker, or adapt the feature extraction process for the speaker. For the purpose of adapting the model, the transcription corresponding to this speech data is also needed. If the correct transcription is available, the procedure is referred to as *supervised adaptation.* Adaptation methods such as *Maximum-A-Posteriori* (MAP) estimation [7] and *Maximum Likelihood Linear Regression* (MLLR) [8] are the most commonly used adaptation methods for this situation. If the transcription of the speech data is not available, a SI system is used to transcribe the data and the resulting (erroneous) transcriptions are used in the adaptation procedure. This is referred to as *unsupervised adaptation.* One of the more remarkable findings in recent years is that unsupervised adaptation can improve recognition performance even when the baseline error rates are as high as 80%, if the adaptation parameters are carefully constrained. In the following sections, we discuss two popular unsupervised adaptation methods.

D.1 Vocal Tract Length Normalization

The pdf's of the acoustic model in ASR systems are meant to model the variation in the feature vectors that are due to statistical variation in the speech. As these pdf's are estimated from the speech of a number of speakers, they model not only the inherent statistical variation within a phoneme, but also the variation due to differing vocal tract characteristics of different speakers. The VTLN [9] adaptation method attempts to remove the inter-speaker component of the variation in the models, and is applied during the feature extraction stage. The idea is to extract acoustic observations in such a way that the feature vectors for a particular phonetic class look similar for different speakers. This is equivalent to mapping the speech from a given speaker to a *canonical space* that characterizes a *canonical speaker.* The acoustic model parameters are then estimated from the canonical feature vectors. The method is based on the observation that a dominant source of difference between speakers is their pitch and formant ranges. Consequently, if the power spectrum for a speaker is scaled in frequency so that the formant frequencies for the speaker take on a target value (the target being the value of the formant frequency for the *canonical* speaker), then it would eliminate the inter-speaker source of variability in the acoustic feature vectors. For further details, the reader is referred to [9]. The use of VTLN to normalize the feature vectors reduces the WER from 38.0% (see Section III-C) to 35.0%, a relative improvement of 7.9%.

D.2 Linear Transform Adaptation

One of the major strengths of SI systems is that the acoustic models are generic enough to match the feature vectors for any speaker reasonably well. However, this characteristic is also a limitation because the acoustic model never matches the feature vectors for a particular speaker as well as an SD system that is built from a large amount of data from the same speaker. Consequently, speaker adaptation method tries to use a limited amount of adaptation data from the test speaker, and change the SI models to bring them closer to the SD models. A commonly used method [8], [10] focuses only on the parameters of the gaussian pdf's that make up the acoustic model (and ignore the transition probabilities). This method assumes that the adapted pdf's are obtained by applying an affine transformation to the means and covariances of the SI pdf's, and uses the adaptation data to estimate the parameters of these affine transformations. The adapted pdf's are then used in the ASR system to transcribe speech from that speaker. The use of these linear transform adaptation methods reduces the WER from 35.0% (see Section III-D.1) to 34.3% a relative reduction of 2%.

E. System Combination

In the earlier sections, we presented results using a number of different systems. Though the average performance (WER) of most of these systems is similar, there are significant differences between their outputs. For instance the difference in the hypotheses produced by an ASR system that uses Mel cepstra and PLP cepstra could could be as much as 20%. In [11], a method was proposed whereby the output of multiple ASR systems was treated as independent sources of knowledge, and the outputs were combined to produce a composite output that had a lower WER than any of the individual ASR outputs. The method is based on taking multiple ASR outputs and aligning them to form a word transition network (WTN) for the utterance. This is done by an iterative process whereby the first two ASR system hypotheses are aligned to form a WTN; subsequently the next ASR hypothesis is aligned to the WTN, and so on. Finally, the composite WTN is searched by a voting or scoring module to select the best scoring word sequence. Applying this technique to combine the hypotheses of four systems, with WER of 39.0%, 34.3%, 38.7% and 34.4% reduces the final WER to 32.7%, a relative improvement of 4.7%, with respect to the WER of Section III-D.2.

IV. Comparisons with Human Performance

In the previous sections, we described the basic components of an ASR system, and recent advances that resulted in improving the WER from 40.5% (for a baseline system, Section II-F) to 32.7% (see Section III-E), on a VOICEMAIL TRANSCRIPTION task. This is one of the more challenging tasks that ASR researchers are working on. To place the WER performance on this task in context, we also describe the level of performance of ASR systems on several other speech recognition tasks.

- *Broadcast News Task* - transcription of radio broadcast news shows [8]
  - speech sampled at 16KHz
  - perplexity of 150, **WER = 13%**
- *SWITHCBOARD Task* - transcription of spontaneous telephone conversations between two strangers
  - speech sampled at 8KHz
  - perplexity of 80, **WER = 25%**
- *VOICEMAIL Task* - transcription of voicemail
  - speech sampled at 8KHz
  - perplexity of 80, **WER = 28%**
- Digits Task - transcription of digit strings
  - speech sampled at 16KHz or 8 KHz
  - low perplexity, **WER $\leq$ 1%**

From the above, we can see that the WER of ASR systems varies from less than 1% to around 30% depending on the complexity of the task. Th fact that ASR systems today are able to transcribe difficult sources of speech such as telephone conversations and voicemail at only a 30% WER is a testimony to the progress that has been made in these systems over the years. However, in spite of all this progress, recent studies still suggest that machine performance is quite far from human performance across a wide variety of tasks, ranging from high-bandwidth digit recognition to large vocabulary telephony speech [12] (see Table II). Human performance still typically exceeds machine performance by *at least* an order of magnitude across a wide variety of tasks (Table 1). In addition, human recognition performance across *signal-to-noise* (SNR) ratios is highly robust - less than 20% degradation from quiet to SNRs of 10 dB - while even highly compensated machine performance typically degrades by at least a factor of two over such a range.

Task	Machine Performance	Human Performance
Connected Digits[8]	.72%	.009%
Letters[9]	5%	1.6%
Resource Management	3.6%	.1%
WSJ	7.2%	.9%
Switchboard	43%	4%

TABLE II

Human vs. machine recognition word error rates across a variety of tasks (from Lippmann)

V. Conclusion

Large vocabulary continuous speech recognition has progressed enormously over the last three decades, with the word error rate on several tasks improving by several factors. This progress has been slow but consistent (the results reported in the DARPA yearly evaluations show the word error rate reducing by 10-15% relative every year), and has been the outcome of improvements in all the basic components that make up a speech recognition system. However, despite the progress that has been made, the word error rate is still relatively high for spontaneous speech sources, and the systems are also very sensitive to background noise. This is in contrast to human performance, which is often an order of magnitude better, and also much more robust to background noise. Though it may appear that there is a long way still to go to match human performance, if the current rate of yearly progress in the word error rate is maintained, this objective should be within reach in less than a decade, thus enabling those many applications of speech recognition (such as transcription of multi-person conversations and meetings) that are still beyond the capabilities of today's recognizers.

Acknowledgement

We would like to thank the members of the Telephony Speech Algorithms Group for their comments.

References

[1] Proceedings of the NIST Speech Transcription Workshop, *http://www.nist.gov/speech/publications/tw00*, May 2000.

[2] F. Jelinek, "Statistical Methods for Speech Recognition", *ISBN 0-262-10066-5*, MIT Press, Cambridge, Massachusetts, 1999.

[3] G. Saon, M. Padmanabhan, R. Gopinath, S. Chen, "Maximum likelihood discriminant feature spaces", Proceedings of ICASSP, 2000.

[4] L. R. Bahl, P. F. Brown, P. V. deSouza, R. L. Mercer, "Maximum mutual information estimation of HMM parameters for speech recognition", Proceedings of the ICASSP, pp 49-52, 1986.

[5] P. C. Woodland and D. Povey, "Large Scale Discriminative Training for Speech Recognition", Proceedings of ASR Workshop, Sep 2000, Paris.

[6] L. Mangu, E. Brill and A. Stolcke, "Finding consensus among words: lattice-based word error minimization", Proceedings of Eurospeech, 1999.

[7] J. L. Gauvain and C. H. Lee, "Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains, IEEE Trans. Speech and Audio Processing, Apr 1994.

[8] C. J. Legetter and P. C. Woodland, "Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models", Computer, Speech and Language, vol. 9, pp 171, 1995.

[9] S. Wegman, D. McAllaster, J. Orloff, and B. Peskin, "Speaker normalization on conversational telephone speech", Proceedings of ICASSP, 1997.

[10] M. J. F. Gales. Maximum Likelihood Linear Transforms for HMM-based Speech Recognition *Computer, Speech and Language*, vol. 12, 1998.

[11] J. G. Fiscus, "A Post-Processing System to Yield Reduced Word Error Rates: Recogniser Output Voting Error Reduction (ROVER)", Proceedings of IEEE ASRU Workshop, pp. 347-352, Santa Barbara, 1997.

[12] R. Lippmann, *Speech recognition by machines and humans*, Speech Communication, Vol. 22 No. 1, 1997

[8] String Error Rates

[9] Humans presented with continuous letters, machine with discrete