TOWARDS KNOWLEDGE-BASED FEATURES FOR HMM BASED LARGE VOCABULARY AUTOMATIC SPEECH RECOGNITION

Benoît Launay *Olivier Siohan* *Arun Surendran* *Chin-Hui Lee*[†]

Multimedia Communications Research Lab
Bell Laboratories – Lucent Technologies
600 Mountain Ave., Murray Hill, NJ 07974, USA

[†] Department of Computer Science
National University of Singapore, Singapore

{siohan,acs}@research.bell-labs.com benoit_launay@hotmail.com chl@comp.nus.edu.sg

ABSTRACT

This paper describes an attempt to design a knowledge-based large vocabulary speech recognition system. Our motivation is to replace features based on the short-term spectra, such as Mel-frequency cepstral coefficients (MFCC), by features that explicitly represent some of the distinctive features of the speech signal. However, rather than attempting to compute acoustic correlates of these distinctive features, we have engineered an approach where neural networks are trained to map short-term spectral features to the posterior probability of some distinctive features. These probabilities are then used as features in a large vocabulary tied-state HMM-based recognizer. Experimental results on the Wall Street Journal Task show that such a system, while not outperforming a MFCC-based system, generates very different error patterns. After combining the results of a baseline MFCC system with the results of several systems based on the proposed approach, we were able to obtain reductions in word error rates of 19% and 10 % on the 5K and 20K tasks respectively over our best MFCC-based systems.

1. INTRODUCTION

Most state-of-the-art automatic speech recognition (ASR) systems characterize the speech signal using features like Mel-Frequency Cepstrum Coefficients (MFCC) which are based on the short-term spectral properties. These features are then used to train hidden Markov models (HMM) in a purely data-driven fashion. Such an approach does not incorporate our understanding of acoustic-phonetics and contextual variability of the speech signal, and does not allow easy integration of any speech knowledge into the recognition process.

Alternatively, linguists have proposed the use of "distinctive features": a compact set of articulatory and acoustic "gestures" whose combinations can codify meaningful similarities and differences between all sounds [1, 2]. Each phoneme can then be represented by its values in the distinctive feature space, and the difference between phones can be described simply and succinctly in the same space. For example, even though the phones "p" and "b" differ significantly in the spectral space, they differ only by the presence and absence of voicing in the distinctive feature space. Similarly, a change of context may only affect a few distinctive features while causing significant changes in the short-term spectra. Hence it is attractive to use distinctive features in ASR systems.

This is however a difficult task. In order to use distinctive features to build an ASR system, it is first required to define some acoustic correlates of each distinctive feature [3], which describe how the acoustic signal relates to the feature. For example, an acoustic correlate of the sonorant feature is a strong low frequency energy. But identifying acoustic correlates for most distinctive features is a difficult task. The features might have more than one acoustic correlate, some more reliable than others. The next step requires defining and extracting a set of parameters characterizing the acoustic correlates. For example, voice onset time can help in voiced-unvoiced distinction for stop consonants [6]. One of the problem in this step is that, due to the tremendous variability of the speech signal, it is not always possible to automatically and consistently measure such correlates even while using signal processing algorithms. For example, voice onset time relies on the detection of the closure-burst transition which degrades heavily in the presence of noise[6]. Another well known example is the problem of tracking formants [4]. Finally, modeling the features using traditional statistical models like HMM can be a problem since (1) many identified acoustic correlates are not modeled well by Markov processes, and (2) the set of acoustic correlates may consist of heterogenous features that cannot be treated together in HMMs (for example in [5]). An example of a system which uses acoustic correlates followed by HMMs is [5]. That system computed acoustic parameters which corresponded only to the manner of articulation, and was used for classifying speech into phonemic classes only. Overcoming all the above problems in building a distinctive feature based large vocabulary system can be challenging.

The approach presented in this paper is quite different in several regards. Given the difficulty with estimating acoustic correlates, we propose a system which does the following: (1) builds data-driven detectors that can detect the presence or absence of distinctive features directly from the short-term spectral representation and limited temporal information, (2) generates outputs that are homogeneous, and (3) thus can be used directly in a statistical model like the HMM for large vocabulary applications. Similar work was done in [8] and [9]. There are many important differences between this paper and the above works and these will be highlighted as we elaborate on our proposed approach. To briefly mention the unique highlights of our proposed approach, we use non-linear detectors based on both spectral and temporal information, to individually detect (in a probabilistic sense) articulatory and phonetic distinctive features. We use these as features to build a context-dependent HMMs for large-vocabulary speech recognition. Further we combine multiple systems (which have different error patterns) at the word level to improve perfor-

0-7803-7402-9/02/$17.00 ©2002 IEEE

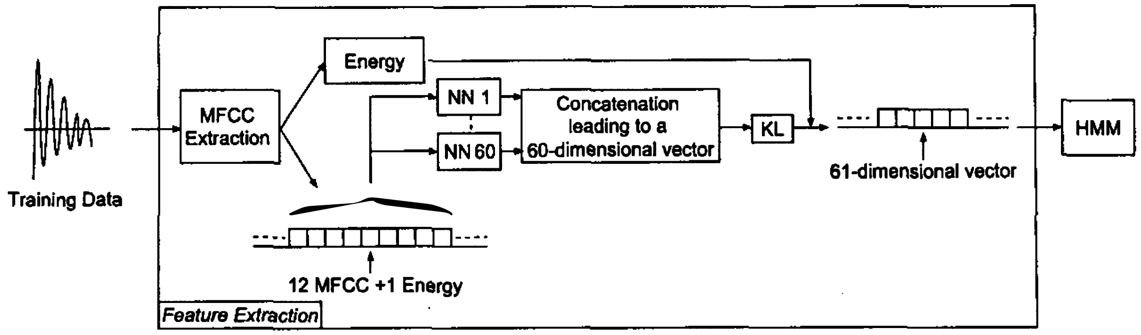

Figure 1: Block diagram of the proposed approach.

mance.

We describe the proposed approach in in Section 2. We present experimental results from the WSJ November-92 task in Section 3, and we summarize the paper in Section 4.

2. PROPOSED APPROACH

In this work, we avoid the problem of trying to compute acoustic correlates of distinctive features. Instead, we use a data-driven approach to learn the mapping from the spectral-based feature space (such as MFCC) to the distinctive feature space. We propose that the feature vector represents directly the posterior probability of the presence of these distinctive features given the data, thus making sure that each feature is comparable to the rest. Given these feature vectors, we can then build a large-vocabulary ASR system and carry out recognition.

This mapping can be learned by training a neural network (NN) on a large database of phonetically segmented speech. The input to the NN at each time instance t is a sequence of $(2N+1)$ consecutive frames $X_{t-N} \ldots X_t \ldots X_{t+N}$, where X_t is a vector of MFCC coefficients computed over a 30-second frame of the signal. The number N is chosen so that the segment of $(2N+1)$ frames adequately characterizes an articulatory event. The corresponding output, Y_t, is a K-dimensional vector where each component represents a specific distinctive feature, set to 1 (0) depending whether the feature is present (absent) in the input, respectively. These non-linear mapping functions can be interpreted as detectors since a high probability indicates the possible presence of a feature, and a low probability indicates an absence. The way our feature vector space is generated bears some similarity with Ellis' [7] and Eide's [8] work. The main difference from [7] is that that our feature space represents the posterior probability of distinctive features instead of single phonemes. In [8], instead of NNs, Gaussian mixture models are used to produce a discriminative score for each distinctive feature, and no information about temporal context is used.

Once such a mapping has been learned by training the neural network, the training data is mapped from the MFCC domain to the distinctive feature domain. A large-vocabulary triphone-based ASR system is then built on these features, using exactly the same approach as if it was built directly on the MFCC feature vectors. In that regard, the phone models defined by the ASR represent a bundle of distinctive features. In [9], the features are all articulatory, and are divided into groups based on manner, place of articulation, etc., and one neural network is built for each group. Each neural network has outputs that are equal in number to the features within each group.

The proposed approach is described in details in the following sections and is outlined in Figure 1.

2.1. Neural Network Training

We used multi-layer perceptrons (MLP) to learn the mapping between the acoustic space and the distinctive feature space. Such a non-linear mapping tool can have significant advantages over the Gaussian mixture models in [8]. The distinctive feature space consists mostly of commonly used articulatory features [11], plus some broad phonetic classes, such as I-Vowel (/ih/, /ix/, /iy/), that represent the questions that are typically used in the decision tree state tying algorithm described in [10]. These 60 "distinctive" features are enumerated in Table 1. Rather than training a single neural network to learn the whole mapping, an MLP is trained individually on each feature, leading to a set of 60 distinct NNs. This is in contrast to the grouping of features in [9]. Each NN consists of 3 layers: one input layer, one single hidden layer and one output layer. The input layer consists of 117 nodes, corresponding to a sequence of 9 consecutive frames of MFCC feature vectors of dimension 13 (including energy coefficient). The input data used to train each NN is obtained by moving a sliding window of 9 frames over the available training data set. The central frame of the sliding window is used to determine which output feature should be active. The output layer consists of 2 nodes, called positive and negative nodes, set to complementary values 1 and 0, depending whether the distinctive feature associated to each NN is present in the input [1]. The number of nodes in the hidden layer varies according to the experimental conditions (*c.f.* Section 3).

In accordance with common practice, we have found it very useful to balance the number of positive/negative training patterns when training each neural network. This typically involves decimating/repeating some of the training patterns to reach the required balance.

Each NN is trained using the standard back-propagation algorithm and its performance is evaluated using cross-validation. Because of the non-linear activation function (sigmoid) used in every node during the NN training, the output values of the NN are either very close to 1 or 0, and can be interpreted as the posterior probability of each distinctive feature. Consequently, the distribution of the output values of each NN are strongly bimodal, with very sharp modes located near 0 and 1. Such distributions are indeed difficult to model accurately, even when using mixtures of Gaussian densities. As a result, we have found it very important to replace the non-linear activation function of each output node by a linear function, which in effect spreads out the output values over the real axis [7].

Once all the neural networks have been trained, the whole training corpus is presented to each neural network. At each time t, 60

[1] The choice of using 2 output nodes was dictated by the software package that we used, since a single binary output node could be used instead.

General	Vowels		Consonants		
Stop	Front Vowel	Rounded	Unvoiced	Non Anterior	Central Stop
Nasal	Central Vowel	Unrounded	Voiced	Continuant	Back Stop
Fricative	Back Vowel	Reduced	Front Consonant	Non Continuant	Voiced Fricative
Liquid	Long	IVowel	Central Consonant	Positve Strident	Unvoiced Fricative
Vowel	Short	EVowel	Back Consonant	Negative Strident	Front Fricative
Front	Dipthong	AVowel	Fortis	Neutral Strident	Central Fricative
Central	Front Start	OVowel	Lenis	Syllabic	Back Fricative
Back	Fronting	UVowel	Neither Fortis or Lenis	Voiced	Affricate
	High		Coronal	Unvoiced	Not Affricate
Noise	Medium		Non Coronal	Stop	
Silence	Low		Anterior	Front Stop	

Table 1: List of the 60 phonetic features (after [10])

2-dimensional outputs are produced (forward propagation of the input through the NN). Only the positive output nodes only are kept and concatenated to create a 60-dimensional feature vector, while the output of the negative nodes are simply discarded.

2.2. Karhunen-Loeve Transform

Because of the nature of the set of 60 distinctive features that have been chosen, some correlation is likely to exist between the outputs of each neural network, violating the diagonal assumption of the Gaussian mixtures used in the HMM training. A Karhunen-Loeve (KL) transform can be applied on the 60-dimensional data, effectively decorrelating the feature components. Optional dimensionality reduction can be applied also at this stage to remove eigenvalues close to 0, but we have typically observed that reducing the dimension hurts the final performance.

After KL, the energy component of the original MFCC vector is then added to the KL-transformed feature vector, leading to a 61-dimensional vector (in case of no dimensionally reduction). This final feature vector is then used instead of the MFCC to represent the speech signal. Compared to MFCC, such a feature vector has now a more explicit physical interpretation since it relates directly to the distinctive features of phonemes [2]. Such features can then be used in place of MFCC to build an HMM-based recognition system, as described in the next section.

3. EXPERIMENTS AND RESULTS

Experiments are performed on the Wall Street Journal (WSJ) task. The 61-dimensional feature vectors (1 energy coeff + 60 KL-transformed features) are used as features to build triphone HMM models[3] on the WSJ SI-84 training set using the decision-tree state tying algorithm described in [12]. We point out that the first and second order derivative of this feature set are not used. A total of 4164 tied-states with an average of about 10.5 Gaussian mixture components per state is obtained, which is very similar to the size of HMMs that we typically build on MFCC features. Experiments have been carried out on the 5K and 20K Nov-92 tests[4] using a 5K and 20K word pronunciation lexicon, respectively, that were generated automatically using a general English text-to-speech system [13]. The language model used in all experiments is the trigram language model provided by NIST for the WSJ task. For comparison purposes, we have also built a MFCC-based system, leading to the word error rates (WER) reported on the first line of Table 2.

The second line of Table 2 reports the WER for the system built using the 61-dimensional feature vectors, with 1000 nodes in the hidden layer of each neural network. While the performance is significantly worse than the MFCC-based system, we observed that the error patterns are very different, suggesting that the results of both system could be combined.

We then decided to build an additional system using a different set of features, but following the same principle used to build the 60+1 distinctive features. Instead of using distinctive features, we used the 44 phone labels used in our HMM-based system as features. The neural network was trained in a slightly different way. Instead of training 44 separate neural networks, we built a single NN with 44 output nodes. During the training of the NN, at each time t, the output of the neural network activates the phone corresponding to the input. This is in spirit similar to Ellis' tandem architecture [7]. An HMM-based system is again built on this new feature set (which also includes the energy coefficient), tuned to generate a number of Gaussian densities similar to the one built on the MFCC and the 60+1 distinctive features. The obtained WER are given on the third line of Table 2, indicating that such a system outperforms the 60+1 distinctive features, but is not as good as the baseline MFCC-based system. Again, we observed that the errors generated by this system are quite different from the errors of both the MFCC and the distinctive features.

Based on this observation, we decided to combine the results of some of our systems using ROVER[5] [14], which essentially corresponds to a majority vote decision. (Kirchhoff's work [9] also uses a combination scheme, but she chooses to combine the phone output probabilities using the sum and product rules). We first combined different systems (with different number of nodes in the hidden layer of the neural networks) built on the 60+1 distinctive features and on the 44+1 phone-based features. The obtained results are given in the fourth line of Table 2, labelled "Combination without baseline". Such a system is marginally better than our baseline MFCC, but is significantly better than our best system built either on the 60+1 or 44+1 features.

Again, we observed that many errors corresponding to the ROVERed NNs systems seem to be not correlated with errors corresponding to the baseline MFCC system. We then included our baseline MFCC system into the ROVER combination, leading to the results given in the last line of Table 2. This corresponds respectively to about 20%

[2] up to the KL transform

[3] These models are cross-word triphone models, position-independent, gender independent.

[4] si_et_05 and si_et_20

[5] The so-called "method 1" was used, without any confidence measures

Features	Test 5K	Test 20K
Baseline (MFCC)	4.6	11.8
60+1 distinctive features	7.6	16.8
44 phone-based features	5.6	13.0
Combination without baseline	4.4	11.8
Combination with baseline	3.7	10.6

Table 2: Word error rates (%) for various feature sets and combinations on WSJ Nov-92, 5K and 20K.

and 10% relative improvement over our best MFCC system, on the 5K and 20K task, respectively, our best results ever on these tasks.

4. CONCLUSION AND DISCUSSION

This paper describes our preliminary attempt at building a large vocabulary ASR system using distinctive features. Rather than trying to derive the acoustic correlates of distinctive features, we have used a data driven approach to learn a mapping between the MFCC domain and the distinctive features associated to the corresponding phone segment. While this work is still at an early stage, we have shown the following:

- It is possible to build a large triphone-based system with acceptable performance using the proposed set of distinctive features;
- The error patterns corresponding to the distinctive feature-based system are very different from the errors of the MFCC-based system;
- Significant improvement in performance can be obtained by combining the results of the standard MFCC-based system and systems built using the proposed approach.

Indeed, this suggests that the features we built carry information not available in the MFCC domain.

Many issues have not been addressed in this paper. For example, we always assumed that when training the neural networks, the distinctive features that should be activated at each time t are the ones given by the identity of the central frame of the sliding window. As a result, when the center of the sliding window passes over a phone boundary, the output of the NN will suddenly deactivate some features and activate others. We believe that a smoother transition is required. In some unreported experiments, to alleviate some of the boundary effects between phone segments, slowly decaying output values have been used, leading to smoother transition from active to non-active features between consecutive frames. These preliminary experiments have not led to any improvement so far, but more work is required.

Another major issue is that the HMM does not properly integrate the feature information. Suppose that the KL is not used, and that the NN operates with non-linear (sigmoid) activation in the output nodes (leading to a 0/1 output for each feature). For a given HMM, if a single component of the distinctive feature vector is incorrectly predicted by the NN, the likelihood of that frame will become almost equal to 0 because there will be a very strong mismatch for that specific feature component between the observation and the model. This illustrates that standard HMM may not well suited to handle distinctive features since they are too sensitive to errors affecting one specific NN predictor for example. Solutions to this problem could involve using a multi-band approach where each feature would correspond to a single band, or replacing the HMM formalism with a different pattern matching paradigm.

Finally, since the 60 distinctive features used here have been extracted using separate neural networks, it is possible to replace their outputs with explicit acoustic correlates like the ones proposed in [5]. We believe this can be a very promising way to incorporate speech knowledge into large scale ASR systems.

5. ACKNOWLEDGEMENT

We thank the International Computer Science Institute (ICSI) for their neural network training software.

6. REFERENCES

[1] N. Chomsky and M. Halle, *The sound pattern of English*, MIT, 1968.

[2] G. Fant, *Speech sounds and features*, MIT, 1973.

[3] K. Stevens, "Acoustic correlates of some phonetic categories", *JASA*, vol. 68, pp. 836-842, 1980.

[4] J. L. Flanagan, *Speech analysis, synthesis and perception*, Springer, 2nd edition, 1972.

[5] N. N. Bitar and C.Y. Espy-Wilson, "Knowledge-based parameters for HMM speech recognition", Proc. ICASSP, pp. 29-32, Atlanta, 1996.

[6] P. Niyogi and M. M. Sondhi, "Detecting stop consonants in continuous speech", submitted to JASA.

[7] D. Ellis, R. Singh and S. Sivadas, "Tandem acoustic modeling in large vocabulary recognition", Eurospeech '01, Denmark.

[8] E. Eide, "Distinctive features for use in an Automatic Speech Recognition System", Eurospeech 2001, pp. 1613-1619, Denmark.

[9] K. Kirchhoff, "Robust Speech Recognition Using Articulatory Information", TR-98-037, ICSI, 1998.

[10] J. Odell, *The use of context in large vocabulary speech recognition*, Pd. D. thesis, University of Cambridge, 1995.

[11] N. Clements, "The geometry of phonological features", *Phonological Yearbook*, vol. 2, pp. 25-252, 1995.

[12] W. Reichl and W. Chou, "A decision tree state tying based on segmental clustering for acoustic modeling", Proc. ICASSP, pp. 801-804, Seattle, 1998.

[13] R. W. Sproat and J. P. Olive, "Text-to-speech synthesis", *AT&T Technical Journal*, vol. 74, pp.35-44, 1995.

[14] J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)", Proc. ASRU, pp. 347-352, Santa Barbara, 1997.