Improved Modeling and Efficiency for Automatic Transcription of Broadcast News

Ananth Sankar[1]   Venkata Ramana Rao Gadde
Andreas Stolcke   Fuliang Weng[2]
Speech Technology and Research Laboratory
SRI International, Menlo Park, CA, U.S.A.

To appear in *Speech Communication*

[1] Now with Nuance Communications, Menlo Park, CA 94025, USA
[2] Now with Intel Corporation, Beijing, China

Abstract

Over the last few years, the DARPA-sponsored Hub-4 continuous speech recognition evaluations have advanced speech recognition technology for automatic transcription of broadcast news. In this paper, we report on our research and progress in this domain, with an emphasis on efficient modeling with significantly fewer parameters for faster and more accurate recognition. In the acoustic modeling area, this was achieved through new parameter tying, Gaussian clustering, and mixture weight thresholding schemes. The effectiveness of acoustic adaptation is greatly increased through unsupervised clustering of test data. In language modeling, we explored the use of non-broadcast-news training data, as well as adaptation to topic and speaking styles. We developed an effective and efficient parameter pruning technique for backoff language models that allowed us to cope with ever increasing amounts of training data and expanded N-gram scopes. Finally, we improved our progressive search architecture with more efficient algorithms for lattice generation, compaction, and incorporation of higher-order language models.

Zusammenfassung

In jüngster Zeit wurde die automatische Transkription von Rundfunknachrichten durch die von der amerikanischen DARPA geförderten Hub-4-Spracherkennungswettbewerbe vorangetrieben. In diesem Artikel berichten wir über Fortschritte auf diesem Gebiet, mit einem Schwerpunkt auf effizienter Modellierung mit weniger Parametern zwecks beschleunigter und genauerer Spracherkennung. In der akkustischen Modellierung wurde dies erreicht durch neue Verfahren zur Parameterbindung, zum Clustern von gaußschen Verteilungen und zum Komprimieren von Gewichten in Mischverteilungen. Die Effektivität von akkustischer Adaptierung wurde durch automatisches Clustern der Testdaten erheblich verbessert. In der Sprachmodellierung untersuchten wir die Benutzung von Trainingsdaten außerhalb der Rundfunknachrichten-Domäne sowie die Anpassung an Themen und Sprechstil. Wir haben ein effektives und effizientes Verfahren zum Parameter-Pruning in Backoff-Sprachmodellen entwickelt, das es uns ermöglicht, stetig wachsende Trainingskorpora und längere N-gramme zu verwenden. Abschließend beschreiben wir Verbesserungen in der progressiven Sucharchitektur unseres Erkenners, mit effizienteren Algorithmen zur Erzeugung, Komprimierung und Expandierung von Wortgraphen.

Résumé

Durant ces dernières années, les évaluations Hub-4 des systèmes de reconnaissance de la parole continue sponsorisées par DARPA ont fait progresser les techniques de reconnaissance de la parole pour la transcription de nouvelles audio-diffusées (“broadcast news”). Dans cet article, nous présentons notre recherche et nos progrès dans ce domaine, en nous concentrant plus particulièrement sur une modélisation efficace utilisant sensiblement moins de paramètres, permettant ainsi d’améliorer la vitesse et les performances de la reconnaissance vocale. En termes de modélisation acoustique, les améliorations ont été obtenues en utilisant une nouvelle méthode pour l’ajustement des paramètres, l’agrégation des Gaussiennes, et le seuillage des poids des mélanges de Gaussiennes. L’efficacité de l’adaptation acoustique est grandement améliorée par l’agrégation non supervisée des données de test. En modélisation du langage, nous avons étudié le résultat de l’utilisation de données d’entraînement ne provenant pas de “broadcast news”, ainsi que de l’effet de l’adaptation au sujet et au style de parole. Nous avons développé une technique efficace d’élagage des paramètres pour des modèles de langage avec repli (“backoff”) ce qui nous a permis de supporter l’accroissement continuel de la quantité de données d’entraînement et l’extension de la portée des N-grammes. Enfin, nous avons amélioré notre architecture de recherche progressive en utilisant des algorithmes plus efficaces pour la génération et la compaction de treillis, ainsi qu’en y incorporant des modèles de langage d’ordre supérieur.

1 Introduction

In recent years there has been increasing interest in developing large-vocabulary continuous speech recognition (LVCSR) systems for speech found in real sources. Broadcast news, in particular, has been the testbed for the DARPA-sponsored Hub-4 continuous speech recognition (CSR) evaluations over the last few years, and represents a significant challenge to speech recognition researchers.

Many interesting problems are associated with the automatic recognition of broadcast news. One problem is that the speech is in the form of a single long stream, whereas typical automatic speech recognition (ASR) systems are designed to process sentence-length units of speech. ASR systems work best when the segment to be recognized is homogeneous with respect to speaker and acoustic condition. It is also desirable, both for ASR and for speech understanding, that the segments correspond to linguistic units such as sentences or phrases. An interesting challenge, therefore, is to develop algorithms that can automatically segment a long stream of speech according to such criteria. Another problem with broadcast news is the many different variations of speech, such as conversational speech, noisy speech, speech in the presence of music, non-native speech, or a combination of these variations. It is necessary to develop techniques that are robust to these variations. Finally, it is important to focus attention on real-time recognition to transfer this technology to applications such as information archival and retrieval.

In this paper, we review our work on broadcast news transcription over the last three years. The paper is divided into sections detailing work in acoustic processing and modeling, language modeling, and lattice-based decoding. Comparative experimental results showing the performance of the techniques are given in each section.[1]

2 Broadcast News Task Description

The data used for the Hub-4 evaluation task consists of recordings of broadcast news shows from various television and radio sources. In contrast to previous test sets used in DARPA evaluations, the Hub-4 data is naturally occurring data, collected under realistic conditions. It is a mixture of various different speech styles, acoustic conditions, and background non-speech segments, making the recognition task difficult. The Hub-4 data has been divided into seven different acoustic focus conditions to facilitate the study of different speech recognition problems in this data. These conditions are:

F0 : Clean read speech (e.g., broadcast news anchor)

F1 : Conversational speech (e.g., interviews in the news studio)

F2 : Telephone speech (e.g., telephone interviews)

F3 : Speech with background music (e.g., introduction of stories)

F4 : Noisy speech (e.g., field interviews; noisy channels)

F5 : Non-native speech (e.g., non-native reporters)

FX : Anything that could not be classified into the previous categories

Two types of Hub-4 test data have been used by the community: (1) the partitioned evaluation (PE) data and (2) the unpartitioned evaluation (UE) data. The PE data consists of manually-created acoustically homogeneous segments. The UE data, on the other hand, is simply the original recording from the broadcast news show. All recent DARPA evaluations have used only UE test data. A more detailed description of the Hub-4 data is given in Stern (1997).

Until the 1998 Hub-4 evaluation, the basic evaluation metric was the system word error rate. In 1998, a 10 times real-time spoke evaluation was introduced to focus attention on real applications.

3 Acoustic Processing and Modeling

3.1 Acoustic Segmentation and Clustering of UE data

Processing a single long UE segment containing both speech and significantly long non-speech segments is difficult. Thus, we first chop the UE data into segments of manageable length that contain only speech. For segmentation, we first run a fast recognition

[1] However, since this work was done over the course of the last few years, the reader should be careful to note that the same test sets and baseline systems are not used across different sections.

step on the UE segment using a parallel male/female context-dependent (CD) phonetically tied mixture (PTM) hidden Markov model (HMM) set to produce a hypothesized sequence of gender-tagged words and background segments (Sankar, Weng, Rivlin, Stolcke, and Gadde, 1998). Both the words and background segments are time-tagged. The models used for this recognition step are trained using the Hub-4 training data, and 5 minutes of non-speech segments (silence, background noise and music) are used to train the background model.

The segmentation algorithm processes the output from the recognition step. It first removes any non-speech segments longer than one second, and then chops at the remaining non-speech regions to create nominally 10-second segments. In addition, a new segment is created whenever a gender change occurs. The resulting segments are thus nominally 10 seconds long, and are labeled by gender. This algorithm does not attempt to make sure that the resulting segments are acoustically homogeneous. A segment may contain multiple speakers or speech from different acoustic focus conditions.

To facilitate adaptation to the test data, the segments were clustered using bottom-up agglomerative clustering (Sankar et al., 1998; Sankar, Heck, and Stolcke, 1997; Heck and Sankar, 1997). The distance measure used for clustering is derived as follows: First we train a Gaussian mixture model (GMM) using all the test segments, and a separate mixture weight distribution for each segment to these shared Gaussians. The mixture weight distribution for a cluster of segments is simply the weighted-by-counts average of the individual distributions. The distance between two segment clusters is then defined as the weighted-by-counts increase in entropy of the mixture weight distribution due to merging the two clusters (Sankar et al., 1998). This is similar to the approach we use for HMM state clustering (Digalakis, Monaco, and Murveit, 1996a).

We tested the segmentation algorithm with the 1996 Hub-4 development test data. For four of the test shows in this data, manually created PE segments were available. We could thus compare our automatically created segments against these. We ran recognition on the manually created and automatically created PE segments for these shows using a 20,000-word bigram language model (LM), and non-crossword gender-dependent HMMs. Both speaker-independent (SI) models and models adapted to the segment clusters using maximum likelihood (ML) transformation-based adaptation (Sankar and Lee, 1994, 1996; Digalakis, Rtischev, and Neumeyer, 1995; Legetter and Woodland, 1995) were used. Table 1 gives word error rates for the manually created and automatically created segments. In the case of the SI models, the word error rate was 1.5% (absolute) worse for the automatically created segments. However, for the adapted models this difference increased to 2%. This is probably because our segmentation algorithm does not guarantee acoustically homogeneous segments, thus not deriving maximum benefit from acoustic adaptation.

PE Segment Type	Models	
	SI	Cluster-Adapted
Manually created	37.9	35.5
Automatically created	39.4	37.6

Table 1: Word error rates (%) for the PE and UE segments

3.2 Modeling the Acoustic Focus Conditions

SRI's acoustic modeling technology is based on state-clustered HMMs. Acoustically similar HMM states are clustered together, each cluster sharing a set of Gaussian distributions. Each HMM state in a cluster has a different set of mixture weights associated with the shared Gaussians (Digalakis et al., 1996a). In SRI's system, the shared Gaussians are called "Genones", and the models "Genonic HMMs".The observation density for state i is given by

$$p_i(x) = \sum_{m=1}^{M} w_{i,m} N_{g,m}(x), \qquad (1)$$

where state i belongs to state cluster g, $N_{g,m}$ is the mth Gaussian distribution corresponding to state cluster g, M is the number of shared Gaussians, and $w_{i,m}$ is state i's mixture weight for the mth Gaussian. The observation density parameters and other HMM parameters are estimated using ML training. The expectation-maximization (EM) algorithm (Dempster, Laird, and Rubin, 1977; Juang, 1985) is used to iteratively increase the likelihood of the models using transcribed training data.

Since the acoustic focus conditions in the Hub-4 data are so different from each other, we decided, in our initial work, to train a separate gender-specific Genonic HMM for each focus condition (Sankar et al., 1997). However, the first release of the Hub-4 training data contained only 50 hours of data, 35 hours of which were speech segments and the rest non-speech segments not usable for training. This data accounted for all the conditions; thus there was not enough data to train good condition-specific models. We addressed this problem by using ML transformation-based adaptation (Sankar and Lee, 1994, 1996; Digalakis et al., 1995; Legetter and Woodland, 1995). We adapted a model trained on the Wall Street Journal (WSJ) database (Doddington, 1992) to each of the focus conditions except for the F1 condition. For the F1 condition, we used the Switchboard corpus (Godfrey, Holliman, and McDaniel, 1992) to train the seed models. We did this because both Switchboard and F1 contain conversational speech. While the idea of condition-specific models made intuitive sense, work done by BBN (Kubala et al., 1997) showed that a single model trained on all the Hub-4 data performed better. Motivated by their results, we also compared the performance of our condition-specific models to a single gender-specific Hub-4 model trained on the 50 hours of Hub-4 training data (Sankar et al., 1998).

Table 2 gives recognition word error rates for the male subset of the 1996 Hub-4 development test set. Recognition was run from bigram lattices (Murveit, Butzberger, Digalakis, and Weintraub, 1993) generated using the 20,000 word bigram language model (LM) we used for the 1996 evaluations. The single Hub-4 model gave a relative 6.1% lower word error rate than the condition-specific models. Since training a single Hub-4 model is easier, we have since been using this approach to train broadcast news models.

Models	Word Error (%)
Condition-Specific	41.12
Single Hub-4 Model	38.61

Table 2: Comparison of condition-specific models vs. a single Hub-4 model

It is possible that with a large amount of training data, it would be possible to train good condition-specific models. Currently there are 200 hours of Hub-4 training data available; however, we have not repeated condition-specific model experiments with this data. One disadvantage of condition-specific models is that it is necessary to automatically determine the acoustic condition of the test data, which is not an easy task in itself. Most broadcast news systems currently use one or at most two, models. In the latter case, a separate model is used for broadband speech and for telephone speech (for example, see Woodland et al. (1998)).

3.3 New Training Algorithms for Acoustic Models

The EM algorithm is an iterative algorithm that recomputes the model parameters, based on their current values, so as to increase the model likelihood with each iteration (Dempster et al., 1977). The algorithm performance is dependent on the initial parameter values, and can, at best, arrive at a locally optimal solution. To investigate this issue, we developed and studied several techniques for HMM parameter initialization and training (Sankar, 1998a).

Another important problem in training acoustic models for speech recognition is that of robustly estimating a large number of parameters with limited data. We addressed this problem by developing new HMM training algorithms based on previously developed acoustic adaptation techniques (Sankar, 1998c).

3.3.1 Parameter Initialization and Training

3.3.1.1 SRI's Previous Training Algorithm. We start with a brief description of SRI's previous HMM training algorithm. By way of example, consider the problem of training an HMM with 1000 state clusters and 32 Gaussians in each of the corresponding Genones. This is done by first training a PTM system, where all states in a phone share the same set of 100 Gaussians. The states in this phone are then clustered using bottom-up agglomerative clustering (Digalakis et al., 1996a), and a cut in the cluster tree is chosen so as to give 1000 state clusters.

The 32 Gaussians in each state cluster are initialized using the corresponding 100 PTM Gaussians. The 100 Gaussians in each phone are clustered down to 32 for each state cluster through a series of steps involving the selection of the most likely Gaussians for each state cluster, and also Gaussian merging. Details of the algorithm can be found in Digalakis et al. (1996a).

This approach poses the following potential problem for the initial values of the Gaussians in the state clusters and hence the final models: The 100 PTM Gaussians cover the entire acoustic space for a particular phone; however, each state cluster for this phone covers only a small part of this large acoustic space. Thus, the PTM Gaussians may not be appropriate for initializing the Gaussians in the individual state clusters, and may result in inefficient use of the parameters.

3.3.1.2 Gaussian Splitting and Merging. We implemented a Gaussian initialization scheme based on the splitting strategy commonly used in vector quantization (Linde, Buzo, and Gray, 1980; Gersho and Gray, 1991). In this approach, we first estimate a single Gaussian model for each Genone. We then split the Gaussian for each Genone into two by slightly perturbing the mean of the Gaussian along the direction of the standard-deviation vector, and reestimate the model by further EM training. This process of splitting and retraining is repeated until the required number of Gaussians is achieved. At each stage, we can choose how many Gaussians to split. Thus, if there are currently n Gaussians which we want to increase to m Gaussians, then we split the $m - n$ Gaussians which have the largest average sample variance (Sankar, 1998a). A similar Gaussian splitting algorithm is used in the Cambridge University HTK system, though a different criterion is used to select which Gaussian to split (Young and Woodland, 1993).

The Gaussian splitting approach can be configured in a variety of ways. For example, we may split all Gaussians at each stage, or may split only the single largest variance Gaussian, or may do something in between these extremes. We experimented with many of these approaches and found that there was not a very significant difference in performance. Thus, we decided on a simple strategy that splits all Gaussians at each stage until we have the desired number of Gaussians per Genone.

The Gaussian splitting algorithm tends to uniformly distribute the training data among the Gaussians (Sankar, 1998a). Thus, if a Genone has very little data, then all its Gaussians may be poorly estimated. To ensure robust Gaussian estimation, we used a Gaussian merging algorithm. In this method, the Gaussians in a Genone are iteratively merged using bottom-up agglomerative clustering until all Gaussians have at least a threshold amount of data (Sankar, 1998a). This threshold is specified by the user, and its optimum value is experimentally determined. For clustering, the distance between two Gaussians is given by the weighted-by-counts increase in entropy due to merging the Gaussians. Combining Gaussian merging and splitting by doing a merge operation before every split operation gives the GMS algorithm (Sankar, 1998a).

3.3.1.3 Experimental Results. For these experiments, we used the Wall Street Journal (WSJ) corpus (Doddington, 1992). We trained HMMs using a small subset of the WSJ SI-284 male training data. We used 71 of the 142 male training speakers and about 50 sentences from each for a total of about 3500 training sentences. We created three different WSJ test sets, denoted as WSJ1, WSJ2, and WSJ3, each with 10 male speakers and about 3600 words, for a total of about 10,900 words in all. For speed of experimentation, recognition was run from bigram lattices (Murveit et al., 1993).

We trained two sets of acoustic models: the first had 991 state clusters and the second had 2027. Both models had 32 Gaussians per Genone. We trained the models using both our previous training algorithm and the GMS algorithm. Table 3 shows that the GMS algorithm performs similarly to the old method for the 991 Genone model, but is significantly better for the 2027 Genone model, where the number of parameters is very large relative to the amount of training data. This shows the robustness of the GMS algorithm relative to our previous approach.

Since speaker adaptation (Sankar and Lee, 1994, 1996; Digalakis et al., 1995; Legetter and Woodland, 1995) is a common technique used in most state-of-the-art systems, we studied the interaction between the training algorithm for the original speaker-

	Word Error Rate (%)			
	Old algorithm		GMS algorithm	
	Number of Genones			
Database	991	2027	991	2027
WSJ1	23.7	25.3	23.5	23.9
WSJ2	13.7	15.5	13.5	14.1
WSJ3	24.3	26.0	23.9	25.1

Table 3: Comparison of word error rates (%) for systems with different numbers of parameters

	Training algorithm for SI models			
Models	Old		GMS	
	Number of Genones			
	991	2027	991	2027
Speaker-independent	23.7	25.3	23.5	23.9
Adapted	20.5	21.1	19.9	20.3

Table 4: Comparison of word error rates for WSJ1 before and after adaptation using different approaches to train the SI models

independent (SI) models and speaker adaptation. Table 4, shows the word error rate before and after speaker adaptation for WSJ1 for the 991 and 2027 Genone systems when the SI models were trained with the old algorithm and the GMS algorithm. The results show that the performance both before and after adaptation was superior with the GMS algorithm algorithm.

3.3.2 Robust Parameter Estimation

While the GMS algorithm gives robust parameter estimates, it does so indirectly by merging Gaussians that have too little data. Thus, if there is less training data, fewer Gaussians will be trained, giving less acoustic resolution. To train a large number of Gaussians, we developed the tied-transform HMM (T^2-HMM), which uses ML acoustic adaptation methods to robustly estimate HMMs with a large number of parameters (Sankar, 1998c).

3.3.2.1 Tied-Transform HMM. We explain the concept of T^2-HMMs using the state-cluster tree in Figure 1. The leaf nodes in the tree correspond to individual HMM states. The other nodes in the tree correspond to state clusters that are created using bottom-up agglomerative clustering (Digalakis et al., 1996a). State clusters can be generated by cutting the tree at intermediate levels. The figure shows state clusters at two different levels with N and M state clusters, where $N > M$.

Suppose our goal is to train an HMM for the larger number of state clusters N. However, we do not have enough data to robustly estimate the large number of Gaussians in this system. In the T^2-HMM, we solve this problem by training an HMM for the smaller number of state clusters M, for which we assume we have enough data to robustly estimate each Gaussian. We can always select a small enough M so that robust Gaussian estimates are possible. Each state cluster in the larger HMM is a descendent of an ancestor state cluster in the smaller HMM as shown in the figure. The Gaussians in the state clusters of the larger HMM are transformed versions of the ancestor Gaussians in the smaller HMM. In the figure, the transformations $T(1), \ldots, T(m)$ are used to map the Gaussians in the Genone $GEN(0)$ to the Gaussians in the Genones $GEN(1), \ldots, GEN(m)$. $T(i)$ can also be a set of transforms, each tied to a cluster of acoustically similar Gaussians in the Genone GEN(i). Since the transforms are tied to a set of Gaussians in the N-state-cluster HMM, they can be estimated with the pooled data from all those Gaussians. This results in robust estimates of the transforms.

The estimation problem is now that of computing the parameters of the smaller HMM and the parameters of the transformations. We can use different types of transformations as have been described in the acoustic adaptation literature (Sankar and Lee, 1994, 1996; Digalakis et al., 1995; Legetter and Woodland, 1995; Neumeyer, Sankar, and Digalakis, 1995). In this paper, we chose to use the block-diagonal affine matrix transform of the Gaussian means as this has given us good performance in the past for speaker adaptation (Neumeyer et al., 1995). We solve the ML estimation problem iteratively. First, we assume identity transforms and estimate the parameters of the smaller HMM. Then we keep the parameters of the small HMM fixed, and estimate the transformations.