Rapid Acoustic Model Development using Gaussian Mixture Clustering and Language Adaptation

Nikos Chatzichrisafis¹, Vasilios Digalakis², Vasilios Diakoloukas², Costas Harizakis¹

(1) Dialogos Communications SA
Kissamou 178
Chania, 73100, Greece
{nikos, harizak}@speech.gr

(2) Dept. of Electronics and Computer Engineering
Technical University of Crete
Chania, 73100, Greece
{vas, vdiak}@telecom.tuc.gr

Abstract
This work presents techniques for improved cross-language transfer of speech recognition systems to new, previously undeveloped, languages. Such techniques are particularly useful for target languages where minimal amounts of training data are available. We describe a novel method to produce a language-independent system by combining acoustic models from a number of source languages. This intermediate language-independent acoustic model is used to bootstrap a target-language system by applying language adaptation. For our experiments we use acoustic models of seven source languages to develop a target Greek acoustic model. We show that our technique significantly outperforms a system trained from scratch when less than 8 hours of read speech is available.

1. Introduction
Developing acoustic models for a new language requires large amounts of speech samples that need to be collected, transcribed, and processed to efficiently train the parameters of the acoustic model. Such speech databases have been created for major languages, speaking conditions and tasks. For new languages, the collection, transcription and processing of such amounts of training data accounts for the largest portion of the time needed to develop the new acoustic model and represents an important cost factor.

To address this problem, some researchers tempted to use resources of already developed acoustic models. For example, in [1] cross-language mappings of phones are created by acoustic segment labeling on a held-out training set. The labels are used to acquire frequency counts of overlapping instances of source- and target-language phones. The best performing hidden Markov model (HMM) states are subsequently combined to yield a new target-language phone model, which subsequently is used for adaptation. In [6], it is shown that a multilingual speech recognition system built using speech data from diverse source languages, can serve as a better seed model for language MAP adaptation in an isolated word task. Similar findings are also reported in [8] and [9].

In this paper, we present an alternative algorithm to effectively build a new language-specific speech recognition system using directly acoustic models from other languages. Our method combines HMM transition probabilities, Gaussians and mixture weights of several already developed mono-lingual genonic [3] acoustic models. We employ Gaussian-mixture clustering based on combinations of Gaussian-mixture distances and acoustic similarity to yield language-independent (LI) acoustic models, which are then subjected to adaptation. We evaluate our technique on live field applications with continuous speech data collected over a cell-phone network.

The remainder of this paper is structured as follows: in Section 2 we describe our algorithm for building language-independent systems based on a number of diverse source-language acoustic models. Section 3 describes implementation issues and presents the experimental setup and recognition performance of the various systems. We also present the language adaptation results of the language-independent system. Finally, Section 4 concludes on our work.

2. Acoustic Model Combination

2.1. Genonic Acoustic Models
A typical mixture observation distribution in a genonic HMM-based speech recognition system has the form:

\[ p(x_i|l_s_i) = \sum_{q \in \Omega} p(q|l_s_i) N_g(x_i; \mu_{qg}, \Sigma_{qg}) \]  (1)

where \( s_i \) is the HMM finite-state process which is modeled as a first-order Markov chain with transition probabilities \( a_{ij} = p(s_j = j | s_{i-1} = i) \). \( x_i \) is the observed feature vector at time \( t \). \( \Omega(g) \) is the pool of Gaussian components \( N_g(x_i; \mu_{qg}, \Sigma_{qg}) \) for the \( g \)-th Gaussian codebook (called genotype) and \( p(q|l_s_i) \) is the mixture weight associated to the Gaussian component \( q \). Each Gaussian component is characterized by a mean vector \( \mu_{qg} \) and a diagonal covariance matrix \( \Sigma_{qg} \). For all the acoustic models that we consider in this work there is a total of \( N_g \) genomes and each genome consists of \( N_q \) Gaussians.

2.2. Phone and State Level Mappings
In the first step of the algorithm we establish phone level mappings for the acoustically related phones among the source languages. The mappings, which are based on the Computer Phonetic Alphabet (CPA) [8] symbol set include all HMM states of all allophones and produce language-independent phones. We define several methods for the construction of the language-independent state clusters.

The first method clusters all the states having the same allophone representation in the CPA symbol set. We call this method the acoustically motivated clustering (AMC).
The second method is inspired from the Phonetically Tied Mixture (PTM) acoustic model tying scheme and clusters all states which share the same language-independent central phone and the same state. We will refer to this method as Phone-State Clustering (PSC).

The clustering of phone states in all techniques is accomplished by combining the individual HMM transition probabilities and Gaussian mixtures using the algorithms presented in the following sections.

2.3. Combining the HMM Transition Probabilities

In order to create the target language state transition probabilities we average the transition probabilities of the language-independent state cluster. Although averaging is a simple method, our experiments proved that it is efficient enough. The acoustic likelihoods are dominated by the Gaussians scores rather than the transition probabilities.

2.4. Combining the Gaussian Mixtures

The Gaussian mixtures associated to each state in a language-independent state cluster are combined through concatenation. This process increases dramatically the components in the Gaussian pools as well as the mixture weights (total of $N_{LS} \times N_{q_i}$, where $N_{LS}$ is the number of states in the state cluster).

2.4.1 Gaussian and Mixture Weight Merging

To reduce the size of the Gaussian components in each genone, we perform Gaussian clustering with an agglomerative hierarchical clustering process [5]. We use the weighted-by-counts entropy of the mixture-weight of the nearest Gaussians are replaced by the Gaussian distribution for the languages with indices (1) and (2) respectively. Based on the distance defined above, the two nearest Gaussians are replaced by the Gaussian

$$D_{opt} = \frac{n^{(1)} + n^{(2)}}{2} H(s^{(1)}_{g} \cup s^{(2)}_{g}) - \frac{n^{(1)}}{2} H(s^{(1)}_{g}) - \frac{n^{(2)}}{2} H(s^{(2)}_{g})$$

where $n^{(1)}$ and $n^{(2)}$ are the number of observations used for the maximum-likelihood estimation of the mixture-weight distribution for the languages with indices (1) and (2) respectively. Based on the distance defined above, the two nearest Gaussians are replaced by the Gaussian

$$N_{gs}(x_t; \mu_{\frac{\mu_{gs(1)}}{2} + \frac{\mu_{gs(2)}}{2}}, \sigma_{\frac{\sigma_{gs(1)}}{2} + \frac{\sigma_{gs(2)}}{2}})$$

The mixture-weights of the nearest Gaussians are then merged in a similar way. The process is repeated until the desired number of Gaussians in the pool is reached.

2.4.2 Mixture Weight Smoothing

The merging process introduces zero-valued mixture-weight elements that correspond to Gaussians with no training data. We normalize mixture-weight components by removing an empirically estimated fraction from each non-zero mixture-weight component and redistributing it equally to the zero-valued ones.

2.4.3 Genome Merging

The combination of Gaussian mixtures for the development of the language-independent system increases the number of genones, resulting to reduced robustness and slow adaptation characteristics. To reduce the number of parameters, we repeatedly cluster and concatenate the two “closest” genones until a predefined global distance threshold is reached. The reduction of the Gaussians in the clustered genome is performed through the same merging process described in section 2.4.1.

The computation of the distance between two genones is based on the entropy distance of their Gaussian components. One possible distance is the sum of the pair-wise distances of the individual Gaussians of a pairing between the Gaussians of the two genones. Since exhaustive search for the optimum pairing can be computationally very intensive, we experimented with two sub-optimal genome-distance metrics.

The first greedy sub-optimal distance metric (SODG) is computed by summing the individual distances of the two nearest Gaussians in the two genones, excluding these two Gaussians and repeating the procedure until all Gaussians are exhausted. The second sub-optimal distance metric (SODN) between the genones is defined as the “optimal” pair-wise distance between the N Gaussians in each genome that had the largest amounts of training data.

2.5. Distance Based Clustering

We applied the technique of distance-based clustering (DBC) to develop a class of models that are constructed by repeatedly merging the 2 nearest genones amongst all the source languages that are used by the same language-independent center phone, until a predefined maximum distance is reached. The genones are then clustered and their size is reduced as described in section 2.4.1.

Compared to AMC, where all source-language genones of a particular phone-state are clustered, this technique provides a certain degree of freedom by clustering only those genones which are sufficiently “close”.

2.6. Adaptation

All resulting language-independent systems are adapted using the adaptation technique presented in [4]. It combines maximum likelihood (ML) transformation and maximum a-posteriori (MAP) based estimated models to build the final language-adapted system. This method has also been successfully applied to a dialect-porting task [2].

3. Experiments

3.1. Target and Source Languages

We selected Greek as the target language on which we evaluate our techniques, since large amounts of training data and accurate pronunciation corpora were available. Acoustic models of the source languages listed in Table 1 were used for the development. All source acoustic models were state-of-the-art models targeted at real live-telephony applications.

All languages use a CPA representation for phones, an encoding of the IPA phone-set using ASCII characters, thus allowing easy one-to-one mappings of phones from language
to language. All languages also share special phonemes for silence and noise.

<table>
<thead>
<tr>
<th>Language</th>
<th>Number of phonemes</th>
</tr>
</thead>
<tbody>
<tr>
<td>American English</td>
<td>46</td>
</tr>
<tr>
<td>American Spanish</td>
<td>26</td>
</tr>
<tr>
<td>Czech</td>
<td>40</td>
</tr>
<tr>
<td>French</td>
<td>38</td>
</tr>
<tr>
<td>German</td>
<td>41</td>
</tr>
<tr>
<td>Greek</td>
<td>31</td>
</tr>
<tr>
<td>Italian</td>
<td>33</td>
</tr>
<tr>
<td>Norwegian</td>
<td>40</td>
</tr>
</tbody>
</table>

Table 1: Languages

3.2. System Configuration

The techniques presented in section 2.4 were applied to the set of source-language acoustic models with several configurations. The various language-independent systems that were constructed for our experiments are summarized in Table 2.

<table>
<thead>
<tr>
<th>System Type</th>
<th>Genone Distance</th>
<th>Genomes</th>
<th>Gaussians</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional ML-training</td>
<td>N/A</td>
<td>986</td>
<td>31552</td>
</tr>
<tr>
<td>AMC</td>
<td>N/A</td>
<td>15230</td>
<td>487360</td>
</tr>
<tr>
<td>AMC_SOD1_100</td>
<td>100K</td>
<td>4425</td>
<td>141600</td>
</tr>
<tr>
<td>AMC_SOD1_500</td>
<td>500K</td>
<td>1936</td>
<td>61952</td>
</tr>
<tr>
<td>PSC</td>
<td>N/A</td>
<td>79</td>
<td>17696</td>
</tr>
<tr>
<td>DBC_SOD1_40</td>
<td>40K</td>
<td>1888</td>
<td>60416</td>
</tr>
<tr>
<td>DBC_SOD4_70</td>
<td>70K</td>
<td>1934</td>
<td>61888</td>
</tr>
<tr>
<td>DBC_SODG_40</td>
<td>40K</td>
<td>2110</td>
<td>67520</td>
</tr>
<tr>
<td>DBC_SODG_230</td>
<td>230K</td>
<td>1015</td>
<td>32480</td>
</tr>
</tbody>
</table>

Table 2: Comparison of LI system parameters

The last column shows the amount of Gaussians each system incorporates, defined as the product of the total number of genones with the number of Gaussians per genome.

The first row shows the parameters of a traditionally ML system trained on a large amount of training data from over 2000 speakers. The system in the second row is built using the AMC method. To reduce the amount of parameters we applied the SODN technique and built the AMC_SOD1_100 and AMC_SOD1_500 system.

As a reference PSC system we used a system with a genone size of 224 Gaussians.

The last four systems were built using the DBC techniques described in section 2.5.

3.3. Cheating Reference System

A cheating language-independent system using the mapping information of allophone states to genones, phonemic inventory and HMM transitions of the target-language ML-trained system was built to indicate the upper performance bound of a language-adapted system. This system perfectly mimics the ML-trained system. The only difference is that the system uses Gaussian-mixture components produced from the equivalent source-language models.

3.4. Training and Evaluation Data Sets

For the production of the target-language acoustic models we used the training sets in Table 3. Data in the training sets stems from data collected over land-line telephone network consisting of read speech and were used to adapt the LI systems.

<table>
<thead>
<tr>
<th>Speakers</th>
<th>Utterances</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>475</td>
<td>40min</td>
</tr>
<tr>
<td>20</td>
<td>970</td>
<td>1h 24min</td>
</tr>
<tr>
<td>40</td>
<td>1938</td>
<td>2h 46min</td>
</tr>
<tr>
<td>60</td>
<td>2859</td>
<td>4h 09min</td>
</tr>
<tr>
<td>120</td>
<td>5638</td>
<td>8h 16min</td>
</tr>
<tr>
<td>300</td>
<td>13760</td>
<td>19h 59min</td>
</tr>
<tr>
<td>&gt; 2000</td>
<td>101992</td>
<td>128h 32min</td>
</tr>
</tbody>
</table>

Table 3: Greek Target Language Training Sets

The evaluation sets were 2 sets of live application data, collected from cellular phones, described in Table 4. The Ferries evaluation set consists of expressions for querying departure/arrival information for ferries traveling the Aegean Sea, while the Stock data-set consists of ways asking the stock price of companies listed in the Athens Stock Exchange.

<table>
<thead>
<tr>
<th>Test Set</th>
<th>Ferries</th>
<th>Stock</th>
</tr>
</thead>
<tbody>
<tr>
<td># of Sentences</td>
<td>2194</td>
<td>3400</td>
</tr>
<tr>
<td># of Words</td>
<td>2928</td>
<td>4433</td>
</tr>
<tr>
<td>Dictionary size</td>
<td>794</td>
<td>1506</td>
</tr>
</tbody>
</table>

Table 4: Target Language Evaluation Sets

The evaluation was performed using context-free language models covering the complete evaluation set. The language models also extract semantic value slots from user utterances identifying company names, dates and travel destinations.

3.5. Experimental Results

Figures 1 and 2 compare the semantic interpretation (SI) error rate of the adapted LI systems against their ML-trained counterparts. The horizontal axis shows the number of speakers used for the development of each system. The first column (0 speakers) shows the performance of the initial, non-adapted, LI system.

For a fair comparison between language-adapted and ML-trained systems we built ML-trained systems i) using 1000
genomes with 32 Gaussians per genome, ii) PTM systems with Gaussian clusters of size 32 and iii) PTM systems with cluster size of 224 Gaussians. In the figures however, only the best performing ML-trained system is shown. The curve of those systems is labeled MLB. The configuration labeled baseline shows performance of the ML-trained system developed on over 2000 speakers.

Figure 1: Stock Evaluation Set

Figure 2: Ferries Evaluation Set

Among the AMC and DBC language-adapted systems, the system labeled DBC_SODG_40 showed the best performance and is thus the only system shown in the Figures. The performance of the language-adapted systems improves rapidly as the number of speakers used for adaptation increases. Utilizing more than 60 speakers, that is more than 4 hours of speech, leads to marginal performance benefits. It is worth noting that the performance of the cheating system is always 2.5-4% higher for all configurations resulting from language-adapted systems. The performance of the cheating system indicates the upper performance bound of the multilingual system, when using the native phonetic inventory and cluster map.

From the above figures it can be concluded that the ML-trained system outperforms the language-adapted one, if more than 300 speakers or 20 hours of speech are available.

4. Conclusions

We have developed a rapid language-development method that utilizes acoustic models of existing source languages to build an initial acoustic model for a new target language. Source-language acoustic models are combined using Gaussian mixture clustering to form an intermediate language-independent model using a phone-state clustering scheme, which results to a surprisingly compact system. The language-independent system is then subjected to adaptation using a relatively small target-language development-set.

After adapting the system using data from a read corpus, we evaluated performance on unseen evaluation-sets. We have shown that the language-adapted models, developed using our novel technique, outperform their traditionally-trained counterparts.

When very limited training data is available, our technique reduces the error rate dramatically over standard ML-trained systems: the error rate is halved when 20 speakers or 1.5 hours of training speech is available. This can significantly shorten the development cycle of an application in a new language, since systems with semantic interpretation error rates in the order of 10% that we observed with only 60 speakers or 4 hours of speech can serve as a first deployable prototype that can then be further improved via field adaptation.

5. References