Gaussian Mixture Clustering and Language Adaptation for the Development of a New Language Speech Recognition System

Nikos Chatzichrisafis, Vassilios Diakoloukas, Vassilios Digalakis, Costas Harizakis

Abstract—The porting of a speech recognition system to a new language is usually a time-consuming and expensive process since it requires collecting, transcribing and processing a large amount of language-specific training sentences. This work presents techniques for improved cross-language transfer of speech recognition systems to new, target 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.

Index Terms — Clustering methods, Speech recognition, Languages

I. 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, including a variety of 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.

Furthermore, some of the European and Asian languages, for which well-trained speech recognizers already exist, represent only a small portion of the hundreds of languages worldwide. The need for the rapid development of speech recognition applications could emerge for many of these languages at any time, based on the continuously varying economic and political situation.

To alleviate the development burden for new acoustic models, several techniques have been proposed in the literature. All of these techniques are applied in three phases, which are presented below.

In the first phase, cross-language phone mappings that identify similar speech sounds across languages have to be obtained. In [1], [10] and [14], this is accomplished using knowledge-based methodology which relies only on acoustic-phonetic categorizations. These categorizations are based on the articulatory representations of the phonemes across languages, which have been defined for several languages worldwide by several organizations as in [11] and [9]. An alternative, automatic approach is proposed in [1] based on confusion matrices. The automatic approach allows sub-phonetic mappings as well, however it requires some data from the target-language.

In the second phase, a language-independent (LI) acoustic model is constructed using resources such as speech data and acoustic model parameters from several source languages, as well as the cross-language phone or sub-phone mappings obtained in the previous phase. There are different strategies that have been proposed for the construction of the language-independent model. In [1], each phone or sub-phone model in the inventory of the target language is constructed from the most similar phone or sub-phone model among the source languages. In [10], the language-independent phone models for each of the IPA symbols were trained using the corresponding training data from the source languages. A similar technique is described in [14], although the training process of the models differs. Two other approaches are investigated in [14]. In the first, each language-specific phoneme is trained with data from its own language, thus the trained system consists of a large number of phone models. The only multilingual component applied is a global linear discriminant analysis (LDA) matrix, which is applied to reduce the size of the feature vectors of all source languages [15]. Alternatively, the Gaussian components of the mixtures
are trained with data from all languages and shared across the language phone-models for each common IPA symbol, but the mixture weights remain specific for each language model. In all of the above approaches, the training data of the source languages should be available.

In the final phase, the final, target-language acoustic model is constructed using the language-independent model to bootstrap a training process when sufficient amounts of target-language training data are available [3], [14]. Alternatively, adaptation techniques might be applied. For instance, in [10] Maximum a Posteriori (MAP) language adaptation in an isolated-word task is applied while in [14] and [1] transformation-based (MLLR) or combination of MAP and MLLR adaptation techniques respectively are applied. Discriminative Model Combination (DMC) is also investigated in [1]. Similar adaptation techniques have been also successfully applied in dialectal acoustic model development tasks [12], [3].

In this paper, we present an alternative, novel approach to build the language-independent model, which will then be subjected to language adaptation based on well-known techniques. The main novelty of our approach relies on that it combines exclusively acoustic model parameters from the source languages, and it is not necessary to use any training data from any languages other than the minimum amounts of the target-language data needed for adaptation. The combination of the acoustic models is not a simple selection of the most similar phone model as in [1]. It is a more complex training strategy, which aims in the construction of language-independent models with reduced number of model parameters obtained from appropriate mixture and mixture-component merging. Our method combines HMM transition probabilities, Gaussians and mixture weights of several already developed, well-trained mono-lingual acoustic models [4]. In order to reduce the increased number of sub-phone models in the system, as well as the increased number of model parameters (mixtures and mixture components), we employ Gaussian-mixture clustering based on combinations of Gaussian-mixture distances and acoustic similarity to yield language-independent acoustic models.

There are several advantages of our approach compared to other relative works. First, the new language acoustic model development process is further speeded-up because the time consuming training process is decoupled from the often huge amounts of source-language training data. It also simplifies development infrastructure, since it does not require the management of huge speech databases. Furthermore, our approach is more flexible, since it can combine acoustic-models with different number of Gaussian components and different number of Gaussian mixtures and results in LI acoustic models of predetermined varying size using different techniques. Hence, based on the available amounts of training data and the phoneme set of the target-language, a different training strategy can be selected. We evaluate our technique on various evaluation sets, including a live field application with continuous speech data collected over a cell-phone network. Although we use the genonic continuous acoustic models with arbitrary tying that were presented in [4] for our experiments, our approach can be readily applied to other types of acoustic models. The algorithms that we present do not depend on the method used to obtain the mapping from states to Gaussian mixtures and can thus be applied to any continuous density Gaussian mixture Hidden Markov recognizer.

The remainder of this paper is structured as follows: in section II, we describe our algorithm for building language-independent systems based on a number of diverse source-language acoustic models. In section III we present language adaptation methods employed in this work. Sections IV and V outline implementation issues, and present the experimental setup and recognition performance of the discussed systems, along with results of the corresponding language-independent systems. Section VI finally concludes.

II. ACOUTIC MODEL COMBINATION

This section covers the proposed acoustic model combination techniques. In section II.A, we first summarize the structure of the acoustic models under consideration. In section II.B, we present the acoustic model clustering techniques of the multilingual system that we investigated. Sections II.C to II.E provide details about the methods that combine and merge the transition probabilities and output distributions. The language adaptation method that we apply on the language-independent models is presented in section III.

A. Genonic Acoustic Models

A typical mixture observation distribution in a HMM-based speech recognition system has the form:

\[
p(x_t \mid s_t) = \sum_{q=Q(g)} p(q \mid s_t) N_g(x_t; \mu_g, \Sigma_g)
\]

(1)

where \(s_t\) is the HMM finite-state process which is modeled as a first-order Markov chain with transition probabilities \(a_{ij} = p(s_t = j \mid s_{t+1} = i)\). \(x_t\) is the observed feature vector at time \(t\), \(Q(g)\) is the pool of Gaussian components \(N_g(x_t; \mu_g, \Sigma_g)\) for the \(g\)-th Gaussian codebook (called genone) and \(p(q \mid s_t)\) is the mixture weight associated to the Gaussian component \(g\). Each Gaussian component is characterized by a mean vector \(\mu_g\) and a diagonal covariance matrix \(\Sigma_g\). For all the acoustic models, that we consider in this work there is a total of \(N_g\) genones and each genone consists of \(N_q\) Gaussians. The acoustic models have an arbitrary mapping between the HMM state and the genone indices defined as \(g = \gamma(s_t)\). The process of generating this mapping is described in [4].

B. Phone and State Level Mappings

Our primary goal is to build a language-independent system
based on the available source language-specific systems, which will serve as a seed for language adaptation. We first merge all language-specific context-dependent models and construct a multilingual system. The upper third of Fig. 1 depicts a small sample of the multilingual acoustic space. This sample consists of three context-dependent model units (triphones) derived from three different source languages, specifically Czech (cz), Italian (it) and Spanish (es). The arrows reflect the state-genome mapping for each sample triphone.

Porting this multilingual system to the target language can be done in several ways. The approach that we propose is to search for optimal combinations of the language-dependent models based on an appropriate phonemic inventory in order to create a generic language-independent system with a reduced number of phonetic models and associated parameters. In addition, parameter-clustering schemes are used to obtain a system with a parametric space that is suitable for language adaptation.

All clustering schemes that we investigated were initiated by simple mappings at a phone level. In particular, we first establish phone-level mappings for the acoustically related phones among the source languages. The mappings, which are based on the Computer Phonetic Alphabet (CPA) [11] symbol set, include all HMM states of all allophones and produce the states of the language-independent context-dependent models. Based on these mappings, we define below several methods for the construction of the language-independent state clusters.

1) Acoustically Motivated Clustering (AMC)

The first method, called acoustically motivated clustering (AMC), groups together the states of the different source-language context-dependent acoustic models that have the same allophone representation in the CPA symbol set. For example, the states of two triphones of different source-language acoustic models are grouped together if both the central phone and the left- and right-contexts have the same representation in the CPA alphabet. Fig. 1 shows the AMC implementation for the multilingual acoustic space sample. Since the Czech and Italian triphones in this example have the same allophone representation, they are clustered together and all genones that are associated to the states of the same cluster are now merged. In the final step, a language-independent model unit that represents the cluster is constructed and the merged genones are associated to its states.

This clustering and merging process results in a large amount of genones with an increased number of Gaussian components in the final language-independent system since new genones and Gaussian components are produced through merging. It is, therefore, necessary to perform Gaussian clustering to obtain genones with a common number of Gaussians. Moreover, we perform genome clustering using the process that we describe in section II.D to reduce the number of parameters in the system.

2) Phone-State Clustering

The second method is inspired from the Phonetically Tied Mixture (PTM) acoustic model tying scheme [4]. The source-language triphone models, which share the same language-independent central phone based on the CPA alphabet, are
clustered together and all genones associated to states with the same index of the models in the cluster are merged and associated to the corresponding-index state of the language-independent model.

We will refer to this method as Phone-State Clustering (PSC). The PSC system resulting from the initial multilingual sample system is illustrated in Fig. 2.

In this example, all three language-dependent triphones correspond to the same central phone in the CPA alphabet. Thus, they are all clustered in the same cluster and the genones associated to their states are merged. This results in a relatively small number of genones with a huge amount of Gaussian components, which are subsequently reduced through Gaussian clustering. In the final step, we form common language-independent model units based on the common allophone representation of both the central phone and their left and right context but their states are now associated to the merged genones derived from the cluster with the same central phone.

3) Distance Based Clustering (DBC)

An alternative to the AMC and PSC methods described above is to perform distance-based clustering (DBC) on the multilingual space to reduce the number of genones. Instead of having three genones for all source-language triphone models with the same central phone, we determine the number of genones dynamically through an agglomerative hierarchical clustering procedure.

We first merge the two nearest genones among all the source languages that are used by the same language-independent central phone based on a distance metric. Each genone produced from this merging is reduced in size through Gaussian clustering and is associated to the states of each seed genone. This merging process is repeated until a predefined maximum distance is reached. In the example of Fig. 3, all source-language triphones share the same central phone, thus any seed genone is candidate for being merged with others to construct a language-independent genone. For instance, as we demonstrate in Fig. 3 the final state of both language-independent model units ([f][U][n] and [T][U][p]) are associated to the same common genone.

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 that are sufficiently “close”.

C. Combining HMM Transition Probabilities

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. The transition probabilities of each language-independent context-dependent HMM are obtained by averaging the transition probabilities of the language-specific HMMs of the same allophone representation in the CPA symbol set. Although averaging is a simple method, our experiments proved that it is sufficient. The acoustic likelihoods are dominated by the Gaussians scores rather than the transition probabilities.

D. Combining Gaussian Mixtures

In the various clustering schemes described in section II.B, the Gaussian mixtures associated to each state in a language-independent state cluster are concatenated together to form a language-independent Gaussian mixture (genone). This process increases dramatically both the Gaussian components and the mixture weights in the concatenated genone. Assuming there are $N_{ls}$ states in the state cluster, then the language-independent genone consists of a total of $N_{ls} \times N_q$ Gaussian components. In this section, we show how to reduce the total number of Gaussian components, as well as the overall number of genones, using clustering schemes.

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 [7]. Given the Gaussians $q_1 = N(x; \mu_1, \Sigma_1)$ and $q_2 = N(x; \mu_2, \Sigma_2)$, we can define the entropy-based distance between them as follows:

$$D_H = \frac{(n_1 + n_2)}{2} \cdot H(\Sigma_{1,2}) - \frac{n_1}{2} \cdot H(\Sigma_1) - \frac{n_2}{2} \cdot H(\Sigma_2)$$

where $H(\Sigma_q) = \log_2(\det(\Sigma_q))$ is the self-information content of the $q$-th Gaussian’s variance $\Sigma_q$. As in [4], we weight the entropy of each Gaussian by the quantity

$$n_q = \sum_{i \epsilon \mathcal{F}^{-1}(g)} \sum_{t} \zeta_i(s_t, q),$$

which is the total

![Fig. 3 DBC language-independent system](image-url)
number of samples used to estimate the $q$-th Gaussian in the $g$-th genone and $\zeta_i(s_i, q) = p(s_i, q \mid x_1, \ldots, x_T)$ denotes the accumulation counts of the Gaussian $q$ at time $t$ estimated using the forward-backward algorithm.

If we define the first and second order statistics of the Gaussian $q$ as:

$$\langle x \rangle_q = \sum_{s \in \mathcal{S}} \sum_{t \in \mathcal{T}} \zeta_i(s_i, q) x_i$$

$$\langle xx^T \rangle_q = \sum_{s \in \mathcal{S}} \sum_{t \in \mathcal{T}} \zeta_i(s_i, q) x_i x_i^T$$

we can obtain the merged Gaussian's mean vector $\mu_{k,2}$ and covariance matrix $\Sigma_{k,2}$ as:

$$\mu_{k,2} = \frac{<x>_1 + <x>_2}{n_1 + n_2} \mu_1 + \frac{n_2}{n_1 + n_2} \mu_2$$

$$\Sigma_{k,2} = \frac{<xx^T>_1 + <xx^T>_2}{n_1 + n_2} - \mu_{k,2} \cdot \mu_{k,2}^T.$$

When diagonal covariance matrices are used, the formula for the estimation of the covariance is applied in an element-wise fashion.

The new mixture weight associated to the merged Gaussian is

$$p(q_{k,2} \mid s_i) = \frac{n_1}{n_1 + n_2} p(q_1 \mid s_i) + \frac{n_2}{n_1 + n_2} p(q_2 \mid s_i).$$

The Gaussian merging process is repeated until the desired number of Gaussians in the pool is reached.

2) Genone Merging

The combination of Gaussian mixtures for the development of the language-independent system increases the number of genones in the system and makes the system less robust and unsuitable for adaptation. To reduce the number of parameters, we repeatedly cluster and concatenate the two “closest” genones until a predefined global distance threshold is reached.

The distance between two genones can be defined similarly to the distance between two multivariate Gaussian distributions using the entropy of a mixture of Gaussian distributions. However, since there is no analytic expression for the entropy of a Gaussian mixture [2], one has to rely on approximations or lower bounds of the entropy to define the distance. One set of reasonable approximations for the distance between two genones can be defined using the entropy distance of their Gaussian components. A first approximation is computed by finding the minimum sum of Gaussian distances between all possible pairings between the Gaussians of the two genones. This approximation, however, is intractable because of the large computing cost associated to examining all possible Gaussian pairings.

Since exhaustive search for the optimum pairing can be computationally very intensive, we experimented with two additional sub-optimal genone-distance approximations. 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 genone that had the largest amounts of training data.

Based on the distances defined above, the two nearest genones are merged and a new merged genone consisting of all the Gaussian components of the seed genones is formed. To reduce the Gaussians in the merged genone we perform Gaussian merging through the same merging process described in section II.D.1). Finally, the set of HMM states that were clustered together are associated to the same merged genone.

E. Mixture Weight Smoothing

When two genones from different source-language acoustic models are merged and their Gaussians are pooled together to form the new merged genone, some of the mixture-weights of the state distributions in the language-independent acoustic model will be zero-valued. We therefore, normalize mixture-weight components by removing an empirically estimated fraction from each non-zero mixture-weight component and redistributing it uniformly to the zero-valued ones.

III. ADAPTATION

All resulting language-independent systems are adapted using the adaptation technique presented in [5]. It combines maximum likelihood (ML) transformation adaptation and maximum a-posteriori (MAP) estimated models to build the final language-adapted system. In this way, we combine the advantages of both families of techniques and we obtain both large gains in recognition performance for small amounts of adaptation data as well as performance improvement asymptotically as adaptation data increase.

The combined adaptation method is shown in Fig. 4.

---

**Fig. 4** Language Adaptation using combined transformation-based and Bayesian adaptation.
Language adaptation data is used to estimate a set of linear transformations \((A_c, b_c)\), where \(A_c\) denotes the rotation matrix and \(b_c\) the bias vector for the transformation class \(c\). The rotation matrix can be full, block-diagonal or diagonal and the estimation formulae are naturally described in [6] and [8].

The transformations are then applied to the sufficient statistics of the language-independent (LI) models as follows:

\[
\langle x \rangle_{gq}^{LTA} = A_c \langle x \rangle_{gq}^{LI} + b_c
\]

\[
\langle x^T \rangle_{gq}^{LTA} = A_c \langle x^T \rangle_{gq}^{LI} A_c^T + A_c \langle x \rangle_{gq}^{LI} b_c + b_c \langle x^T \rangle_{gq}^{LI} A_c^T + n_{gq}^{LI} b_c b_c^T,
\]

where \(\langle x \rangle_{gq}^{LI}\) and \(\langle x^T \rangle_{gq}^{LI}\) are the first- and second-order sufficient statistics for genome \(g\) and mixture component \(q\) of the language-independent system, computed by the forward-backward algorithm using all the available data of the various source languages. \(\langle x \rangle_{gq}^{LTA}\) and \(\langle x^T \rangle_{gq}^{LTA}\) are the first- and second-order statistics of the linear transformation-based language adapted (LTA) system. Using the same language adaptation data, we can also train using Maximum-Likelihood a language-dependent (LD) system with the following sufficient statistics:

\[
n_{gq}^{LD} = \sum_{s,\xi(s,q)} \sum_{t} \xi(s,q) x_t
\]

\[
\langle x \rangle_{gq}^{LD} = \sum_{s,\xi(s,q)} \sum_{t} \xi(s,q) x_t
\]

\[
\langle x^T \rangle_{gq}^{LD} = \sum_{s,\xi(s,q)} \sum_{t} \xi(s,q) x_t x_t^T
\]

We finally linearly combine the counts according to:

\[
\langle x \rangle_{gq}^{LA} = \lambda \langle x \rangle_{gq}^{LTA} + (1 - \lambda) \langle x \rangle_{gq}^{LD}
\]

\[
\langle x^T \rangle_{gq}^{LA} = \lambda \langle x^T \rangle_{gq}^{LTA} + (1 - \lambda) \langle x^T \rangle_{gq}^{LD}
\]

\[
n_{gq}^{LA} = \lambda n_{gq}^{LI} + (1 - \lambda) n_{gq}^{LD},
\]

where \(\lambda\) is an empirical parameter which determines the emphasis given on the LI models. Its value should decrease as the adaptation data increase and vice versa. The final language-adapted model means and diagonal covariance matrices are then estimated as:

\[
\mu_{gq}^{LA} = \frac{\langle x \rangle_{gq}^{LA}}{n_{gq}^{LA}}
\]

\[
\Sigma_{gq}^{LA} = \frac{\langle x^T \rangle_{gq}^{LA}}{n_{gq}^{LA}} - \mu_{gq}^{LA} (\mu_{gq}^{LA})^T
\]

### IV. Experimental Setup

#### A. Source and Target Languages

To evaluate our techniques, we selected Greek as the target language since large amounts of training data and accurate pronunciation corpora were available. Acoustic models of the source languages listed in Table I were used for development. Details about the target language are given in Table II.

All source acoustic models were state-of-the-art continuous Gaussian-mixture models targeted at real-life telephony applications and trained on a variety of speech databases including field data from each of the source languages.

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.

#### B. Modeling with Limited Data

As shown in Table I, not all phonemes are covered by our source language set. Dealing with non-existent, or insufficient number of phones is therefore necessary. In our experiments, we approximated missing Greek phones from source languages using knowledge-based mappings between source and target language, effectively reconstructing the full Greek phoneme set.

For PSC, which relies on genones from phone-states of multiple source languages, we concatenated source-language genones to yield language-independent genones with a higher Gaussian component count.

#### C. System Configuration

The techniques presented in section II were applied to the set of source-language acoustic models with several configurations. In all our experiments, we used acoustic models configured with a 27-feature front end that outputs 8 cepstral coefficients, cepstral energy, and their first- and second-order differences. The cepstral features are computed from an FFT filterbank. The various language-independent systems that were constructed for our experiments are:

#### TABLE I

<table>
<thead>
<tr>
<th>Source Language</th>
<th>Number of Phonemes</th>
<th>Coverage of target-language phoneme set</th>
</tr>
</thead>
<tbody>
<tr>
<td>American English (en_US)</td>
<td>46</td>
<td>78.6%</td>
</tr>
<tr>
<td>Mexican Spanish (es_MX)</td>
<td>26</td>
<td>64.3%</td>
</tr>
<tr>
<td>Czech (cz)</td>
<td>40</td>
<td>75.0%</td>
</tr>
<tr>
<td>French (fr)</td>
<td>38</td>
<td>60.7%</td>
</tr>
<tr>
<td>German (de)</td>
<td>41</td>
<td>67.9%</td>
</tr>
<tr>
<td>Italian (it)</td>
<td>33</td>
<td>75.0%</td>
</tr>
<tr>
<td>Norwegian (no)</td>
<td>40</td>
<td>67.9%</td>
</tr>
</tbody>
</table>

#### TABLE II

<table>
<thead>
<tr>
<th>Target Language</th>
<th>Number of Phonemes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Greek (el)</td>
<td>31</td>
</tr>
</tbody>
</table>


1. Baseline: This is the well-trained native system. It is a typical ML system, trained on a large amount of training data from over 2000 speakers.

2. MLB: This a system trained with the standard ML training method without using any knowledge from other source languages, but the amount of training data that are used is limited to the same amount of adaptation data used in the remaining language adapted systems. The MLB system is constructed for comparison to the language adapted systems. The degree of tying and the number of parameters of the MLB system is optimized separately for the various amounts of adaptation data, as described in more details in section V.C.

3. AMC: A system built using the AMC method described in section II.B.1).

4. AMC_SOD1_100: To reduce the amount of parameters in the AMC system we apply the suboptimal genome-distance approximation SODN for N=1, which checks for the “optimal” pair-wise distance between the Gaussians with the largest amount of training data in each genome. The distance that we apply as a threshold is 100K.

5. AMC_SOD1_500: This system is identical to the AMC_SOD1_100 system except the distance threshold is higher (500K instead of 100K)

6. PSC: This system is build using the PSC method. Since this reference system produces small number of genones we increased the genome size from 32 to 224 Gaussians in order to increase the parameters of the system.

7. DBC_SOD1_40: This is a system built using the DBC technique described in section II.B.3) and the same sub-optimal genome-distance approximation algorithm (SODN for N=1) as with the AMC_SOD1_100 system. The genome distance threshold is set to 40K.

8. DBC_SOD4_70: Is a system built using the DBC technique and the sub-optimal genome-distance approximation algorithm SODN for N=4. This algorithm considers the top four Gaussians in each genome in terms of assigned training data to compute the genome pair-wise distance. The distance threshold is set to 70K.

9. DBC_SODG_40: In this system the greedy sub-optimal distance metric (SODG) is for a distance threshold of 40K

10. DBC_SODG_230: The final language-independent system is produced using the same technique with the DBC_SODG_40 system but we consider higher distance threshold (230K).

All language-independent systems described above are summarized in Table III. The last column shows the amount of Gaussians each system uses, defined as the product of the total number of genones with the number of Gaussians per genome. As can be seen, the Baseline system consists of 986 genones and 31552 Gaussians. In the AMC system the number of genones and Gaussians dramatically increase. This is due to the design of the AMC algorithm, since new genones are introduced through merging. For instance, in the example of Fig. 1 the genones labeled (cz-1+it-1), (cz-2+it-2) and (cz-3+it-3) are new genones that have been constructed through merging and were not originally present in any of the source-language acoustic models. The same stands for the DBC algorithm. However, using suboptimal distance metric techniques (SODG and SODN) we can merge genones and reduce the total number of parameters in the system in order to obtain more appropriate models for language adaptation.

D. Cheating Reference System

To obtain an estimate of the upper performance bound of a language-independent system, we also built a “Cheating” language-independent system as follows: we assumed that the mapping information of allophone states to genones, the phonemic inventory and the HMM transition probabilities were known from a Baseline target-language ML-trained system on a large amount of training data. The parameters that were estimated from the source-language systems were the Gaussian components of the mixtures. Source-language models that were not present in the Baseline target system have not been incorporated into the cheating system.

E. Training and Evaluation Data Sets

For the production of the target-language acoustic models, we used the development sets shown in Table IV. This development corpus stems from a data-collection over a landline telephone network, where over 2000 speakers were recruited. Speakers were prompted to answer spontaneous questions, as well as to read from a data sheet containing numbers, yes/no expressions, common names, date expressions and general newspaper text. Textual data included in the read corpus was designed to be phonetically balanced and optimized for telephony applications. The corpus from over 2000 speakers, in the last line of Table IV, was used to train the Baseline system. We used this corpus to extract the gender-balanced subsets with 10 to 300 speakers. Due to the design of the data-collection strategy where no two recording sessions were alike, even small development sets have a fair coverage of the phonemic inventory of the target language.

Table V shows the evaluation sets used in this paper. The Date evaluation set consists of common words to utter dates
TABLE IV
GREEK TARGET LANGUAGE TRAINING SETS

<table>
<thead>
<tr>
<th>Number of 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 V
GREEK TARGET LANGUAGE EVALUATION SETS

<table>
<thead>
<tr>
<th>Date</th>
<th>Digits</th>
<th>Ferries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Sentences</td>
<td>1357</td>
<td>221</td>
</tr>
<tr>
<td>Number of Words</td>
<td>8167</td>
<td>1547</td>
</tr>
<tr>
<td>Dictionary Size</td>
<td>204</td>
<td>24</td>
</tr>
</tbody>
</table>

Fig. 5 System performance for various mixture-weight smoothing factors

weight smoothing factor for three evaluation tasks, the Date, Digits and Ferries. The smoothing factor varies from zero to one. A smoothing factor of zero is equivalent to no smoothing. A smoothing factor of one results in uniformly distributed mixture weights. As can be observed in Fig. 5 a smoothing factor near 0.5 can significantly improve performance.

B. Performance of the source language acoustic models to Greek test-sets

The second series of experiments aimed at the evaluation of how closely the selected source languages match the Greek language. For instance, it is well known that Greek and Spanish have a close match in their phonetic inventory but this is not the case for the other languages we used in the development of the LI system such as French, German, Norwegian and American English.

We first evaluated each of the source languages separately on the Greek Date test-set by substituting Greek pronunciation dictionary with their source-language equivalents. Performance of the various systems is shown in Table VI and is compared to the baseline, well-trained Greek model. It can be seen that Spanish is the best source language among the languages that we consider, whereas the American English, which is covering a larger fraction of the Greek phoneme set (see Table I), is the worst match.

We then repeated the experiment by using the acoustic models of two source languages and listing in the application dictionary all possible combinations of the two source languages phonemes that corresponded to the Greek phoneme of the original Greek pronunciation. One of the two languages was always Spanish. The results, which are shown in Table VI, indicate that combining two source languages results in a better performing acoustic model for the target language, with the exception of the Spanish-plus- American English model.

As an example, for the single source language case the performance (WER) of the recognizer is 21.16% for the Spanish and 27.27% for Czech source models. If we consider both Spanish and Czech as source languages, the WER drops to 16.74% without adaptation. This is attributed to the increased phoneme coverage of the target language obtained when using multiple source languages.

To create a boot model for the target language, it is possible including expression like “tomorrow” or “next week”, while the Digits test set consists of utterances of seven isolated digits. These evaluation sets are held out sets that resulted from the landline data-collection described above. Special care has been taken that speaker populations in the development and evaluation sets are distinct. We additionally evaluated our methods on live application data, collected over a cellular GSM network. This test set is named Ferries and consists of expressions for querying departure and arrival information for ferries traveling the Aegean Sea.

The evaluation was performed using context-free language models covering the complete evaluation set. The language models also generate semantic value slots uniquely identifying, dates, digits and travel destinations. The semantic values returned by the language models are used to evaluate system performance in terms of semantic interpretation error rate (SIER).

F. A Note on Evaluation Metrics

It is known that optimizing word-error-rates (WERs) does not always imply better system performance [16], [13]. In addition, WERs do not directly express perceived usability by the end-user, and thus semantic interpretation error rates are better suited for comparing recognition systems for the applications studied in this paper. For this reason, when we compare performances, we always refer to semantic interpretation error rates (SIER).

We report WER rates for the sake of completeness, but would like to point out that for some evaluation sets WER might give an inaccurate image of system performance, which might be caused due to difficulties such as discriminating inflectional cases (το vs. τον, τις vs. της, etc.).

V. EXPERIMENTAL RESULTS

A. Evaluation of Mixture Weight Smoothing Impact

We first performed a series of experiments to find the impact of the mixture weight smoothing factor in the system performance and optimize the system. Fig. 5 plots the semantic interpretation error rate (SIER) versus the mixture-
to search for the optimal combination of source-language models, since as we saw in the Spanish-American English example above performance may not always improve by using more source models. However, since our focus was on developing algorithms for acoustic model combination, we used in our experiments all source language models that we had available.

C. Comparison of Various System Configurations.

In Fig. 6, Fig. 8 and Fig. 10 we compare the semantic interpretation error rate (SIER) of the adapted LI systems against their ML-trained counterparts on various test-sets. The horizontal axis shows the number of adaptation speakers used for the development of each system. The first value (0 speakers) shows the performance of the initial, non-adapted, LI system.

To compare our algorithms to standard ML training techniques, we use the following strategy: since we have limited amounts of training data, we build multiple ML-trained systems using various degrees of tying and various numbers of parameters as well for each of the various amounts of training data. The baseline MLB system that we compare our techniques to is always the best-performing ML-trained system. Specifically, we built ML-trained systems with a wide range of parameters i) using 1000 genones with 32 Gaussians per genone, 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 for each data point. The curve of those systems is labeled MLB.

We also compare our algorithms to a well-trained ML system and a “Cheating” system that uses knowledge about the phonetic context dependencies of the target language. The configuration labeled “Baseline” shows performance of the well-trained system developed on over 2000 speakers. Finally the label “Cheating” refers to the reference system where the mapping information of allophone states to genones, the phonemic inventory and the HMM transition probabilities are known from a target-language ML-trained system on a large amount of training data.

Among the AMC and DBC language-adapted systems, the system labeled DBC_SODG_40 showed the best performance as can be seen in the comparison of various system performances on the Ferries evaluation set in Table VII.

This is more obvious when no adaptation is performed, or for small number of adaptation speakers. Similar findings were observed on other evaluation sets. Therefore, we show the performance of DBC_SODG_40 in the comparative diagrams in Fig. 6 to Fig. 11. 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 gains. For the more important field-application evaluation set Ferries the MLB system is significantly worse compared to the language adapted one for less than 120 speakers and 8 hours of speech. 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 marks the upper performance bound of the multilingual system, when using the native phonetic inventory and cluster information. The latter indicates the importance of language-dependent modeling of phonetic context dependencies. However, it can be seen that most of the gain is already obtained from our algorithms since some of the systems are very close to the cheating system.

Finally, from the Ferries evaluation set which contains data recorded under different conditions than the training data, it can be concluded that the ML-trained system outperforms the language-adapted one when more than 300 speakers or 20 hours of speech are available. Otherwise, our proposed method is preferable.

VI. 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. When 10 speakers or 40 minutes of training speech is available, semantic error rate on the Ferries evaluation set is almost halved. In addition, our methods perform better than standard ML training when there is a mismatch between training and testing conditions.

The proposed methods decouple the development process of new languages from time and resource consuming iterations.
over typically huge amounts of source-language training data. In addition, our algorithm can significantly shorten the development cycle of an application in a new language, since the systems with the semantic interpretation error rates that we obtain 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.

ACKNOWLEDGMENT

REFERENCES


Nikos Chatzichrisafis was born in Esslingen, Germany, on October 31, 1973. He received the Diploma in Electrical and Computer Engineering from the Technical University of Crete, Greece, in 1998, where he also received his Ms. Eng. degree in 2001.

He joined Dialogos Communications S.A., Chania, Greece on October 1998, where he developed language components for commercial recognition systems. Mr. Chatzichrisafis was with NASA Ames Research Center where he co-developed the Clarissa Procedure Browser, the first speech recognition system deployed in space. He is working as a freelance consultant, currently with the Translation Department of Geneva University, where he develops the MedSLT speech-to-speech translation system.

Mr. Chatzichrisafis’ research interests are in multilingual speech recognition systems, machine translation and user interfaces. He can be contacted at Nikos.Chatzichrisafis@voZup.com

Vassilios Diakoloukas was born in Rhodes, Greece, on November 27, 1971. He received his Diploma in Physics from the University of Crete, Greece, in 1994, the M.S. degree in Electrical Engineering from the University of Durham, England, in 1996, and the Ph.D. degree in Electronics and Computer engineering from the Technical University of Crete, Greece in 2000.

In November 2001 he joins Dialogos S.A., Chania, Greece, as the head of the research department where he worked for the development and improvement of speech recognition acoustic models for Nuance’s speech recognition system. Since December 2003 he is in the department of Electronics and Computer engineering of the Technical University of Crete in Chania. He is the main researcher in several European and National research projects and he teaches undergraduate courses on Signals and Systems and Information Theory.

Mr. Diakoloukas’ research interests are in pattern and speech recognition, speech and image processing and computer networks. He has authored many journal articles and refereed conference proceedings, and has received the Young Author Best Paper award on Speech Processing from the IEEE Signal Processing Society in 2000. He can be contacted at vdiak@telecom.tuc.gr

Vassilios Digalakis was born in Chania, Greece, on February 2, 1963. He received the Diploma in electrical engineering from the National Technical University of Athens, Greece, in 1986, the M.S. degree in electrical engineering from Northeastern University, Boston, MA, in 1988, and the Ph.D. degree in electrical and systems engineering from Boston University, Boston, MA, in 1992.

From January 1992 to February 1995 he was with the Speech Technology and Research Laboratory of SRI International in Menlo Park, CA. At SRI, he was a principal investigator for ARPA research contracts and he developed new speech recognition and speaker adaptation algorithms for the DECRYPT speech recognition system. Since 1995, he is a Professor in the department of Electronic and Computer engineering of the Technical University of Crete in Chania. He is also the Director of the Telecommunications Systems Institute of the Technical University of Crete. He teaches undergraduate and graduate courses on Speech Processing and on Digital and Analog Communications.

His research interests are in pattern and speech recognition, speech processing and digital communications.

Prof. Digalakis has authored more than 50 articles in journals and refereed conference proceedings, and has received two best paper awards from the IEEE Signal Processing Society. He can be contacted at vas@telecom.tuc.gr

Costas Harizakis was born in Athens on November 6, 1975. He received his diploma in Electronics and Computer Engineering from the Technical University of Crete in 1999, where he also received his M.S. degree in 2003.

He joined Dialogos S.A., Chania, Greece, in 1999 as chief software architect, where he is leading the development of the dialogos voice XML platform. Since October 2003 he is a doctoral student in the department of electronics and computer engineering of the Technical University of Crete in Chania.

Mr Harizakis research interests are in software system design, telephony platforms, storage systems and distributed proxy caches. Mr. Harizakis can be contacted at harizakis@speech.gr.
Table VII

Adaptation performance (Word and Semantic Interpretation error rates) of several language-independent systems on the Ferrries test-set.

<table>
<thead>
<tr>
<th>Adaptation</th>
<th>Speakers</th>
<th>AMC</th>
<th>DBC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>WER</td>
<td>WER</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>36.00</td>
<td>31.22</td>
<td>38.39</td>
</tr>
<tr>
<td>10</td>
<td>24.01</td>
<td>18.00</td>
<td>24.56</td>
</tr>
<tr>
<td>20</td>
<td>22.98</td>
<td>16.55</td>
<td>22.23</td>
</tr>
<tr>
<td>60</td>
<td>19.13</td>
<td>13.58</td>
<td>18.85</td>
</tr>
<tr>
<td>120</td>
<td>18.03</td>
<td>11.85</td>
<td>18.27</td>
</tr>
<tr>
<td>300</td>
<td>15.51</td>
<td>10.94</td>
<td>16.53</td>
</tr>
</tbody>
</table>

Fig. 6 Date Evaluation Set – Semantic Interpretation Error Rate
Fig. 7 Date Evaluation Set – Word Error Rate

Fig. 8 Digits Evaluation Set – Semantic Interpretation Error Rate
Fig. 9 Digits Evaluation Set – Word Error Rate

Fig. 10 Ferries Evaluation Set - Semantic Interpretation Error Rate
Fig. 11 Ferries Evaluation Set - Word Error Rate