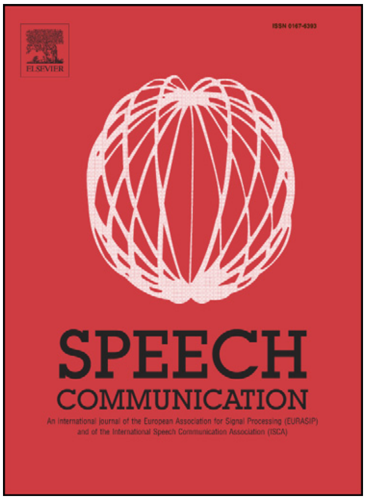
A scalable architecture for multilingual speech recognition on embedded devices

Martin Raab, Rainer Gruhn, Elmar Nöth


ISSN 0167-6393
ELSEVIER
SPEECH
COMMUNICATION
An international journal of the European Association for Signal Processing (EURASIP) and of the International Speech Communication Association (ISCA)

A scalable architecture for multilingual speech recognition on embedded devices

Martin Raab, Rainer Gruhn, Elmar Nöth

PII: S0167-6393(10)00134-2
DOI: 10.1016/j.specom.2010.07.007
Reference: SPECOM 1914

To appear in: *Speech Communication*

Received Date: 9 February 2009
Revised Date: 24 July 2010
Accepted Date: 30 July 2010

Please cite this article as: Raab, M., Gruhn, R., Nöth, E., A scalable architecture for multilingual speech recognition on embedded devices, *Speech Communication* (2010), doi: 10.1016/j.specom.2010.07.007

A scalable architecture for multilingual speech recognition on embedded devices

Martin Raab[*,a,b], Rainer Gruhn[a], Elmar Nöth[b]

[a] *Harman Becker Automotive Systems, Speech Dialog Systems, Ulm, Germany*
[b] *University of Erlangen-Nuremberg, Chair of Pattern Recognition, Erlangen, Germany*

Abstract

In-car infotainment and navigation devices are typical examples where speech based interfaces are successfully applied. While classical applications are monolingual, such as voice commands or monolingual destination input, the trend goes towards multilingual applications. Examples are music player control or multilingual destination input. As soon as more languages are considered the training and decoding complexity of the speech recognizer increases. For large multilingual systems, some kind of parameter tying is needed to keep the decoding task feasible on embedded systems with limited resources. A traditional technique for this is to use a semi-continuous Hidden Markov Model as the acoustic model. The monolingual codebook on which such a system relies is not appropriate for multilingual recognition. We introduce Multilingual Weighted Codebooks that give good results with low decoding complexity. These codebooks depend on the actual language combination and increase the training complexity. Therefore an algorithm is needed that can reduce the training complexity. Our first proposal are mathematically motivated projections between Hidden Markov Models defined in Gaussian spaces. Although theoretically optimal, these projections were difficult to employ directly in speech decoders. We found approximated projections to be most effective for practical application, giving good performance without requiring major modifications to the common speech recognizer architecture. With a combination of the Multilingual Weighted Codebooks and Gaussian Mixture Model projections we create an efficient and scalable architecture for non-native speech recognition. Our new architecture offers a solution to the combinatoric problems of training and decoding for multiple languages. It builds new multilingual systems in only 0.002% of the time of a traditional HMM training, and achieves comparable performance on foreign languages.

Key words: multilingual speech recognition, non-native speech, projections between Gaussian spaces, Gaussian mixture model distances

1. Introduction

Current state of the art systems already provide speech control, but with the limited processing power and memory of these systems it is difficult to provide speech recognition for many languages. There are situations where it is necessary to recognize multilingual speech. One example is when users drive to other countries and need to input navigation destinations. Another example is speech controlled music selection. The artists and titles in music collections can be from many different languages, and the system has to allow the selection of all of them via speech.

The issue becomes more complicated as the user utters many of these additional speech items with non-native accent. For the dialog in the car navigation and infotainment system, this means that there is a distinguished main language of the system and some additional languages. The main language of the system is the native language of the user. The additional languages are the languages that are dependent on the task.

In the first part of our literature review we analyze previous approaches for multilingual speech recognition. An approach that is used in many works to reduce the decoding complexity is knowledge based model sharing. In this approach, phonemes from different languages share one acoustic model when they have the same IPA (International Phonetic Alphabet, Ladefoged [18]) symbol. Examples are Weng et al. [40], Koehler [16], Uebler [37], Schultz and Waibel [33], Wang et al. [39], Niesler [22]. The works vary in the degree to

[*] Tel.: +49 (0)731 15239 441.
Email addresses:
`martin.raab@informatik.uni-erlangen.de` (Martin Raab),
`rainer.gruhn@alumni.uni-ulm.de` (Rainer Gruhn)

which they enforce the clustering between languages.

There are less works that experimented with data driven model sharing in the acoustic model. Koehler [16], Dalsgaard et al. [5] measure the log-likelihood difference on development data to determine the similarity of phonemes, as motivated by Juang and Rabiner [14]. Wang et al. [39] trains phones from different languages on the same codebook and measures the distances between phones by the Euclidean distance between the mixture weight vectors of the Hidden Markov Models (HMMs).

The knowledge based and the data driven approaches are well suited for the recognition of many languages if there are no additional knowledge sources. In our case, we know the native language of the speaker from the graphical user interface language of the system. This is the main language of interaction between the user and the system and a user usually utters commands, spellings and digit sequences in that language. Hence it is vital for a commercial system to recognize this main language with maximum performance. Therefore we introduced Multilingual Weighted Codebooks (MWCs) as a technique that does not deteriorate the performance in the main language. MWCs are basically a main language codebook that is enriched with some additional Gaussians to better cover all languages. We were able to show the benefits of MWCs for both native speakers and non-native speakers [29, 30].

There are also works that propose techniques for the efficient handling of multilingual language models. Examples are Harbeck et al. [8], Nöth et al. [24], Fuegen [6]. For our work, this is less relevant, as we focus on a command and control application or selection-from-list type applications with little room for the user to make non-native grammar mistakes.

The second part of our literature review focuses on non-native speech. [36] present several results with different adaptation techniques like MAP and MLLR and achieve up to 30% WER improvement. Bouselmi et al. [3] introduce confusion based acoustic model integration that allows additional HMM structures for frequently confused phoneme models. They report improvements of up to 70% WER and an absolute WA of up to 98.0% without speaker adaptation on the Hiwire test data [34] that we also use. However, using the Hiwire data for adaptation and testing is likely to give good results as the lexicon size is very limited and the same speakers are in the adaptation and test set. This was analyzed by Lang [19], where it was shown that standard Baum-Welch re-estimation gives comparable results to [3]. Lang also proved the overfitting problem as an adaptation on Hiwire did not lead to improvements on ISLE [21], another non-native corpora. Lang used the same recognizer and the same training data as the work in this paper.

These acoustic model adaptation methods have the drawback that they need adaptation data from the corresponding accents. The biggest database known to the authors covers almost 30 different accents [32] (overview of existing collections in Raab et al. [28]), but there are a lot more accents that are not covered. Other techniques try to circumvent the need for special training or adaptation data. [1] and [7] use manually derived pronunciation rules for the modification of lexicons. However, their approaches require expensive human work and achieve more moderate improvements in the range of 15% to 30% WER.

There are also methods that try to extract information about non-native accents from a comparison of the native language of the speaker and the spoken language. Witt [41] proposes three different algorithms for this, amongst other Model Merging. Improvements of up to 27% WER are reported for the methods without online adaptation. However, the work of Witt was performed on continuous HMMs and can not directly be applied to a semi-continuous HMM. Witt's algorithms do also benefit from adding Gaussians from other languages, so there is the question to what extent for example Model Merging can add on top of MWCs. The same question arises with work from Tan and Besacier [35].

Finally, we have to deal with the limited resources of an in-car system. We use a semi-continuous speech recognizer [10] as a technique to keep the memory and processing demand of the system relatively low [15]. A similar system was proposed in Park and Ko [25]. Such a semi-continuous speech recognizer achieves parameter tying through one single set of Gaussians for all phoneme models.

We combine our semi-continuous system with MWCs as this data driven model sharing technique has the advantage that it does not degrade performance on the main language. The problem with MWCs is that they do depend on the actual language combination. This leads to an unacceptably high training effort for more than a couple of languages.

A solution to avoid these unacceptably high numbers of systems is to provide just the right system, instead of providing all possible systems. While making this decision is impossible in the offline part of the training of speech recognizers, it is possible on the actual embedded system in the car of the user.

Figure 1 depicts how such a process can look like for the two applications that we have in mind, multilingual destination input and music selection. In the destination

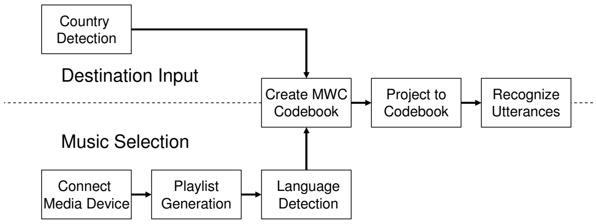


Figure 1: Generating a user adapted system on an embedded system.

selection, the system determines the language of nearby destinations. In the music selection, the languages of interest can be determined from the language distribution in the music database. There are three tasks that are common to both examples, language identification, MWC based codebook creation and the generation of HMMs on top of the generated codebook. All the tasks should be fast enough to run on an embedded system.

We do not go in detail about language identification of text in this paper as it is widely used and there are freely available tools like TextCat [23] for 69 different languages. One approach is for example that the languages are recognized based on n-gram frequencies of letter sequences that are specific for each language. Language recognition rates are in the range of 90% or higher for 30 letter sequences [38]. The MWC task was already discussed. The last task is to provide HMMs that use only Gaussians from this codebook. Due to the runtime constraints, we do not consider a common Baum-Welch training. Instead, we project the Gaussian Mixture Models (GMMs) from their different monolingual codebooks to the previously generated MWC. In this paper we present seven different methods for this projection, three of them were presented before [31].

The remainder of this paper is organized as follows. In the next section we present our multilingual baseline system. Section 3 describes our algorithms and introduces the concept of the scalable architecture. In Section 4 the experimental setup is described. Section 5 presents the experimental results. Finally, a conclusion is drawn in Section 6.

2. Benchmark and baseline systems

The starting point for our comparison systems is trained monolingual semi-continuous HMM speech recognizers. This means that we have trained triphone models for all languages.

The benchmark system for the recognition of multiple languages combines all triphone models in one large model set. This is nothing else as evaluating all monolingual recognizers in parallel. Thus this system can achieve monolingual performance in all languages. However, in this approach all Gaussians from all languages that are currently set active for recognition have to be evaluated. This violates the motivation for the use of a semi-continuous system as no longer only one fixed number of Gaussians has to be evaluated for all models. To summarize, this approach can be considered as an upper bound in performance, but requires a linear increase of resources on the embedded system with the number of considered languages.

Our baseline systems reduce the resource need through only using Gaussians of the current main language of the system. This gives monolingual performance for the native language of the user and does not increase the number of Gaussians that have to be evaluated. The drawbacks of this approach are significantly reduced performance on the additional languages and a training effort that is quadratic with respect to the number of languages considered. This also leads to the fact that a quadratic number of systems has to be deployed on the embedded system.

The following describes the necessary steps for the generation of our baseline system for one given main language. The HMM models of all additional languages are added to the model set of the main language recognizer. However, these additional models have to be trained again, as the Gaussians in the codebook have changed. Therefore, each phoneme model of each of the additional languages is rebuild with data from the corresponding language, but this time the HMMs can only model their output distribution with Gaussians from the main language codebook. Figure 2 sketches the procedure for an example bilingual German/English system.

3. Algorithm description

3.1. Multilingual Weighted Codebooks

To improve the performance on the additional languages of our baseline system, the monolingual codebook is replaced by a Multilingual Weighted Codebook (MWC). The MWC is basically the main language codebook plus some additional Gaussians. Figure 3 depicts an example for the extension of a codebook to cover an additional language. From left to right one iteration of the generation of MWCs is represented.

The picture to the left shows the initial situation. The Xs are mean vectors from the main language codebook, and the area that is roughly covered by them is indicated by the dotted line. Additionally, the numbered Os are mean vectors from the second language codebook.

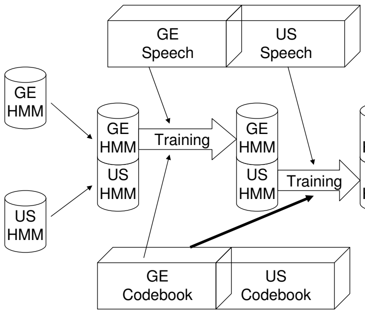


Figure 2: Baseline system for an example German/English bilingual system. Each HMM becomes trained with speech from its corresponding language. All HMMs use only Gaussians from the main language (German) codebook.

Supposing that both Xs and Os are optimal for the language they were created for, it is clear that the second language contains sound patterns that are not typical for the first language (Os 1,2 and 3).

The middle picture shows the distance calculation. For each of the second language codebook vectors, the nearest neighbor among the main language Gaussians is determined. These nearest neighbor connections are indicated by the dotted lines. Our previous experiments showed that using the Mahalanobis distance produces the best results [29].

The right picture presents the outcome of one iteration. From each of the nearest neighbor connections, the largest one (O number 2) was chosen as this is obviously the mean vector which causes the largest vector quantization error. Thus, the Gaussian O number 2 was added to the main language codebook.

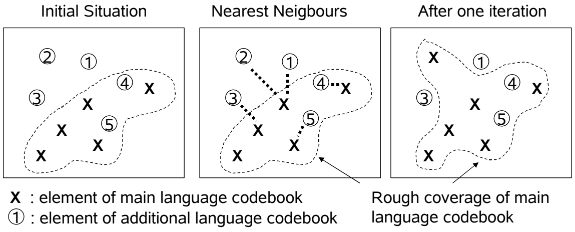


Figure 3: The idea of MWCs. The three pictures present one iteration of the MWC algorithm. On the left, the initial situation is depicted. The nearest neighbor calculation is shown in the middle. The rightmost picture presents the coverage final situation in which the coverage of the MWC has been extended through the addition of one extra Gaussian.

In Raab et al. [29] we have shown that Multilingual Weighted Codebooks (MWCs) increase performance on the additional languages for fluent non-natives without affecting performance on the main language. Raab et al. [30] proves that MWCs also help for the recognition of less fluent non-native speakers.

A negative aspect of MWCs is that they depend on the languages that are added. In fact, the number of different systems grows exponentially with the number of languages the system has to support.

3.2. Distance between GMMs

In the literature many distances between Gaussian Mixture Models have been proposed. Examples are an approximated Kullback Leibler divergence [9], the likelihood difference on a development set [14, 16] or the L2 distance [13, 12]. The likelihood difference on a development set has the disadvantage that development data is necessary. We use the L2 distance between Gaussians, as a closed solution exists for this distance, which is not the case for the Kullback-Leibler distance.

The L2 distance [20] between two Gaussian mixture models A and B is defined by

$$D_{L2}(A,B) = \int (\boldsymbol{\alpha}^T \mathbf{a}(\mathbf{x}) - \boldsymbol{\beta}^T \mathbf{b}(\mathbf{x}))^2 d\mathbf{x} \tag{1}$$

$\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are the weight vectors of the Gaussian vectors **a** and **b**.

$$\boldsymbol{\alpha} = \begin{pmatrix} w_1^a \\ w_2^a \\ \vdots \\ w_n^a \end{pmatrix}, \mathbf{a}(\mathbf{x}) = \begin{pmatrix} \mathcal{N}(\mathbf{x};\boldsymbol{\mu}_1^a,\boldsymbol{\Sigma}_1^a) \\ \mathcal{N}(\mathbf{x};\boldsymbol{\mu}_2^a,\boldsymbol{\Sigma}_2^a) \\ \vdots \\ \mathcal{N}(\mathbf{x};\boldsymbol{\mu}_n^a\boldsymbol{\Sigma}_n^a) \end{pmatrix} \tag{2}$$

$$\boldsymbol{\beta} = \begin{pmatrix} w_1^b \\ w_2^b \\ \vdots \\ w_m^b \end{pmatrix}, \mathbf{b}(\mathbf{x}) = \begin{pmatrix} \mathcal{N}(\mathbf{x};\boldsymbol{\mu}_1^b,\boldsymbol{\Sigma}_1^b) \\ \mathcal{N}(\mathbf{x};\boldsymbol{\mu}_2^b,\boldsymbol{\Sigma}_2^b) \\ \vdots \\ \mathcal{N}(\mathbf{x};\boldsymbol{\mu}_m^b,\boldsymbol{\Sigma}_m^b) \end{pmatrix} \tag{3}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean and the covariance of the Gaussians. The distance D_{L2} can be calculated as follows

$$\begin{aligned} D_{L2}(A,B) &= \int (\boldsymbol{\alpha}^T \mathbf{a}(\mathbf{x}) - \boldsymbol{\beta}^T \mathbf{b}(\mathbf{x}))^2 \, d\mathbf{x} \\ &= \sum_i \sum_j \alpha_i \alpha_j \int a_i(\mathbf{x}) a_j(\mathbf{x}) \, d\mathbf{x} \\ &\quad - 2 \sum_i \sum_j \alpha_i \beta_j \int a_i(\mathbf{x}) b_j(\mathbf{x}) \, d\mathbf{x} \\ &\quad + \sum_i \sum_j \beta_i \beta_j \int b_i(\mathbf{x}) b_j(\mathbf{x}) \, d\mathbf{x} \end{aligned} \tag{4}$$

with $a_i(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i^a, \boldsymbol{\Sigma}_i^a)$ and $b_i(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i^b, \boldsymbol{\Sigma}_i^b)$. In order to solve this problem, the correlation $\int \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)\, \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)\, d\mathbf{x}$ between the Gaussians needs to be calculated. Petersen and Pedersen [26] state that

$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) = c_c \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \quad (5)$$

with

$$c_c = \mathcal{N}(\boldsymbol{\mu}_1; \boldsymbol{\mu}_2, (\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}2)) \quad (6)$$

$$\boldsymbol{\mu}_c = (\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1})^{-1}(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2) \quad (7)$$

$$\boldsymbol{\Sigma}_c = (\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1})^{-1} \quad (8)$$

Thus all correlations between all Gaussians can be calculated and written in three matrices $\mathbf{M}^{AA}$, $\mathbf{M}^{AB}$ and $\mathbf{M}^{BB}$.

$$M_{ij}^{AA} = \int a_i(\mathbf{x}) a_j(\mathbf{x}) d\mathbf{x} \quad (9)$$

$$M_{ij}^{AB} = \int a_i(\mathbf{x}) b_j(\mathbf{x}) d\mathbf{x} \quad (10)$$

$$M_{ij}^{BB} = \int b_i(\mathbf{x}) b_j(\mathbf{x}) d\mathbf{x} \quad (11)$$

Hence Equation (4) can be written as

$$D_{L2}(A, B) = \boldsymbol{\alpha}^T \mathbf{M}^{AA} \boldsymbol{\alpha} - 2\boldsymbol{\alpha}^T \mathbf{M}^{AB} \boldsymbol{\beta} + \boldsymbol{\beta}^T \mathbf{M}^{BB} \boldsymbol{\beta} \quad (12)$$

3.3. Optimal projections between Gaussian spaces

The purpose of Equation (12) is to measure distances between different *given* Gaussian mixtures. In this work it is more interesting to find an $\boldsymbol{\alpha}_{min}$ that minimizes $D_{L2}(A, B)$. The solutions from this section were first presented in Raab et al. [31].

To obtain the minimum we differentiate D_{L2} with respect to $\boldsymbol{\alpha}$:

$$\frac{\partial D_{L2}}{\partial \boldsymbol{\alpha}} = 2\,\mathbf{M}^{AA}\boldsymbol{\alpha} - 2\,\mathbf{M}^{AB}\boldsymbol{\beta} \quad (13)$$

In order to find the minimum, we set the gradient to $\vec{\mathbf{0}} = (0, 0, \ldots, 0)^T$. This leads to the optimal weights $\boldsymbol{\alpha}_{min}$.

$$\boldsymbol{\alpha}_{min} = (\mathbf{M}^{AA})^{-1}\mathbf{M}^{AB}\boldsymbol{\beta} \quad (14)$$

This $\boldsymbol{\alpha}_{min}$ is a true minimum when the second derivative of D_{L2} is positive definite. The second derivative is $2M^{AA}$. M^{AA} is a correlation matrix and therefore positive semidefinite. As long as none of the Gaussians is linear dependent on the other Gaussians, this matrix is positive definite and $\boldsymbol{\alpha}_{min}$ a true minimum.

Projection 1. An optimal projection from GMM A to B that minimizes the D_{L2} error $D_{L2}(A, B)$. The projection creates negative weights for Gaussians, and there is no normalization of the sum of the Gaussian weights.

Despite the fact that the proposed projection is optimal with regard to the L2 distance, it is likely to be suboptimal for the use in a common speech recognizer. The reasons are that

1. The elements of $\boldsymbol{\alpha}_{min}$ do not sum to one, thus some states can always have higher scores than others.
2. Some weights for Gaussians are negative. In our decoder the corresponding log probabilities are replaced by a threshold.

The first problem can be solved with the Lagrange constraint that all weights have to sum to one. The Lagrange function to minimize can be stated as:

$$L(\alpha, \lambda) = \boldsymbol{\alpha}^T \mathbf{M}^{AA} \boldsymbol{\alpha} - 2\boldsymbol{\alpha}^T \mathbf{M}^{AB} \boldsymbol{\beta} + \boldsymbol{\beta}^T \mathbf{M}^{BB} \boldsymbol{\beta} + \lambda(\sum_i (\alpha_i) - 1) \quad (15)$$

with the additional Lagrange multiplier λ. Differentiating this function gives

$$\frac{\partial L}{\partial(\alpha, \lambda)} = \begin{pmatrix} 2\,\mathbf{M}^{AA} & \vec{\mathbf{1}} \\ \vec{\mathbf{1}}^T & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \lambda \end{pmatrix} - \begin{pmatrix} 2\,\mathbf{M}^{AB} & \vec{\mathbf{0}} \\ \vec{\mathbf{0}}^T & 1/\lambda \end{pmatrix} \begin{pmatrix} \boldsymbol{\beta} \\ \lambda \end{pmatrix} \quad (16)$$

where $\vec{\mathbf{1}} = (1, 1, \ldots, 1)^T$.

Setting the derivation to $\vec{\mathbf{0}}$ and removing λ from the second matrix leads to

$$\begin{pmatrix} \boldsymbol{\alpha}_{min} \\ \lambda \end{pmatrix} = \begin{pmatrix} 2\,\mathbf{M}^{AA} & \vec{\mathbf{1}} \\ \vec{\mathbf{1}}^T & 0 \end{pmatrix}^{-1} \cdot \begin{pmatrix} 2\,\mathbf{M}^{AB} & \vec{\mathbf{0}} \\ \vec{\mathbf{0}}^T & 1 \end{pmatrix} \begin{pmatrix} \boldsymbol{\beta} \\ 1 \end{pmatrix} \quad (17)$$

Resulting in an $\boldsymbol{\alpha}$ vector that sums up to one. When $\mathbf{M}^{AA^{-1}}$ is known, the inverse of the complete matrix can be computed efficiently with the Schur complement [42].

Projection 2. An optimal projection from GMM A to B that minimizes the D_{L2} error $D_{L2}(A, B)$. Additionally, the constraint that all Gaussian weights have to sum to one is enforced. There are negative weights for Gaussians after the projection.

Solving the issue of negative weights is a more difficult convex optimization problem [4]. A common method to solve it are the Karush Khun Tucker constraints [17]. These are basically a generalization of the Lagrange constraints and can work with inequalities by introducing slack variables **s** that transform every inequality in an equality, which can be solved as any Lagrange constraint. In the case here, an inequality constraint has to be introduced for every element of $\boldsymbol{\alpha}$. This gives the new function KKT for the distance between the mixture distribution A and B.

$$\begin{aligned} KKT(\boldsymbol{\alpha}, \lambda, \boldsymbol{\gamma}) = \boldsymbol{\alpha}^T \mathbf{M}^{AA} \boldsymbol{\alpha} - 2\boldsymbol{\alpha}^T \mathbf{M}^{AB} \boldsymbol{\beta} \\ + \boldsymbol{\beta}^T \mathbf{M}^{BB} \boldsymbol{\beta} + \lambda(\sum_i (\alpha_i) - 1) \\ + \sum_{i=1}^{n} \gamma_i(-\alpha_i + s_i^2) \end{aligned} \tag{18}$$

with $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \dots, \gamma_n)$ and $\mathbf{s} = (s_1, s_2, \dots, s_n)$.

When α_i is zero, constraint i is said to be active, otherwise the constraint is inactive. If constraint i is active, γ_i is greater 0. To find the optimal solution, all possible combinations of active constraints and inactive constraints need to be evaluated.

In practice it is not possible to check all the possible combinations for the optimal value. Similar problems have to be solved for Neural Networks [27, 2]. Basically, the idea is to perform a gradient descent on the optimization criterion and a gradient ascent on the equality constraint. Biehl et al. [2] show that a quadratic optimization problem that ignores negative values converges with gradient descent. In our case, the actual implementation needed well tuned update weights to prevent oscillations caused by the opposed equality and inequality constraints. Nevertheless, the sequential iterative optimization algorithm achieved on average almost the same L2 distance as Projection 1 with only three iterations.

Projection 3. An "almost optimal" projection from GMM A to B that minimizes the D_{L2} error $D_{L2}(A, B)$. The weights of the projected distribution sum to one and there are no negative weights.

When applying these projections to our recognizer not all Gaussians are comparable, as the different languages have different LDAs (Linear Discriminant Analyses). Therefore each Gaussian was saved before it was modified by an LDA. Thus we can make our comparisons with comparable Gaussians. These Gaussians are also used for the approximated projections in the next section.

3.4. Approximated projections between Gaussian spaces

In the previous section, three different projections with different constraints were introduced. Each of them has some disadvantages for employment in an embedded speech recognition system. Therefore, we propose some experimentally motivated projections.

The goal of each projection is to map all HMMs of all L languages to one fixed set of N Gaussians (= Recognition Codebook, RC) which can be either mono- or multilingual. Such a mapping can be achieved by mapping all M^l Gaussians of each additional language codebook (= Monolingual Codebook, MC^l) to the RC. Each Gaussian $\mathcal{N}$ is represented by its mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The Mahalanobis distance measures the distance between Gaussians (D_G).

Projection 4. An approximated projection that only compares individual Gaussians in the different codebooks to derive a mapping. Each additional language Gaussian is replaced by another Gaussian according to map_G.

$$\begin{aligned} map_G(\mathcal{N}^i_{MC^l}) = \mathcal{N}^j_{RC} \\ (0 \leq i < M^l, 0 \leq j < N, 0 \leq l < L) \\ j = \arg\min_k D_G(\boldsymbol{\mu}^i_{MC^l}, \boldsymbol{\mu}^k_{RC}, \boldsymbol{\Sigma}^i_{MC^l}) \end{aligned} \tag{19}$$

When all Gaussians from the main language are in the RC, there are further possibilities how HMMs from other languages can be linked to the RC. All states from the main language map only to Gaussians from the RC. Thus when all S states are mapped to RS main language states only Gaussians from the RC are used. The same is true when all HMMs H are mapped to main language HMMs RH. Both of these additional mappings have the advantage that they consider the combination of Gaussians in their distance.

We map states based on the minimum Mahalanobis distance (D_S) between the expected values E of their Gaussian mixture models. The covariance which is needed for the Mahalanobis distance is a global diagonal covariance $\boldsymbol{\Sigma}_{All}$ estimated on all training samples. This covariance can also be calculated from the Gaussians in the codebook, thus there is no need for the actual training data. With D_S we define our state based mapping as Projection 5.

Projection 5. An approximated projection that compares states from additional languages to main language HMM states to derive a mapping. Each individual

HMM state is replaced by another HMM state according to $\mathbf{map_S}$.

$$\begin{aligned}\mathbf{map_S}(\mathbf{s}_l^i) &= \mathbf{s}_{RS}^j \\ &\quad (0 \le i < S_l, 0 \le j < RS, 0 \le l < L) \\ j &= \arg\min_k D_S(E[\mathbf{s}_l^i], E[\mathbf{s}_{RS}^k], \mathbf{\Sigma}_{All}) \end{aligned} \tag{20}$$

Based on D_S we can define a distance between HMMs (D_H). In our system each context dependent phoneme is represented through a three state HMM model. In this case the distance between two phonemes $\mathbf{q}_1$ and $\mathbf{q}_2$ is

$$D_H(\mathbf{q}_1, \mathbf{q}_2) = \sum_{i=1}^{3} D_S(\mathbf{s}_{\mathbf{q}_1}^i, \mathbf{s}_{\mathbf{q}_2}^i) \tag{21}$$

With D_H we can define Projection 6.

Projection 6. An approximated projection that compares HMMs from additional languages to main language HMMs to derive a mapping. Each additional language HMM is replaced by a main language HMM according to $\mathbf{map_H}$.

$$\begin{aligned}\mathbf{map_H}(\mathbf{q}_l^i) &= \mathbf{q}_{RH}^j \\ &\quad (0 \le i < H_l, 0 \le j < RH, 0 \le l < L) \\ j &= \arg\min_k D_H(\mathbf{q}_l^i, \mathbf{q}_{RH}^k) \end{aligned} \tag{22}$$

D_G and D_S provide consistently good performance for different tests, while they use rather different information for their calculation. Therefore we also test a combined map_{G+S}.

Projection 7. An approximated projection that compares both Gaussians and HMM states to derive a mapping. Each additional language state gets a new output distribution probability according to map_{G+S}.

$$\begin{aligned}\mathbf{map_{G+S}}(\mathbf{s}_l^i) &= \gamma_{G+S}\ \mathbf{map_S}(\mathbf{s}_l^i) \\ &+ (1 - \gamma_{G+S}) \begin{pmatrix} w_{\mathbf{s}_l^i}^1 map_G(\mathcal{N}_{MC^l}^1) \\ w_{\mathbf{s}_l^i}^2 map_G(\mathcal{N}_{MC^l}^2) \\ \vdots \\ w_{\mathbf{s}_l^i}^{M^l} map_G(\mathcal{N}_{MC^l}^{M^l}) \end{pmatrix} \\ &\quad (0 \le l < L, 0 \le i < S_l) \end{aligned} \tag{23}$$

with the combination weight γ_{G+S}.

γ_{G+S} has to be determined in experiments. An additional retraining after each of the projections would probably increase the performance. In our experiments no retraining was performed, as this keeps the creation of new multilingual systems as simple as possible and on-demand acoustic model creation feasible.

3.5. Overview of projections

In the previous two sections, several methods for the projection of HMMs from one language to another were proposed. Table 1 summarizes the main information about them. The method column describes which information is used for the projection. The probability column indicates whether the result of the projection is a correct probability distribution.

Table 1: Comparison of projection methods

Projection	Method	Probability
Pro1	L2 minimization	no
Pro2	L2 minimization	no
Pro3	L2 minimization	yes
Pro4	Gaussian mapping	yes
Pro5	State mapping	yes
Pro6	HMM mapping	yes
Pro7	Pro4 + Pro5	yes

3.6. Scalable architecture

In Section 3.3 and Section 3.4 several projections between Gaussian spaces where defined. Each of these projections allows to use only one codebook for all languages, which keeps the decoding feasible on an embedded system. Only at the moment of application it is known which languages have to be recognized. Therefore, if the projections can be calculated on the embedded system, there is no combinatoric problem for the training algorithms.

Thus, the defined projections generate a speech recognizer for every language combination without increasing the training effort. To actually have a scalable architecture, an algorithm is needed that can improve the performance. This can be achieved with the MWC algorithm defined in Section 3.1. This increases the number of Gaussians in our system and hence the memory demand, but the decoding complexity can be kept much lower as with monolingual recognizers that run in parallel. A graphical representation of the overall process was given in Figure 1 of the introduction.

4. Experimental setup

Our semi-continuous HMM speech recognizer uses 11 MFCCs with their first and second derivatives per frame and LDA for feature space transformation. Monolingual recognizers for English, French, German, Spanish and Italian are trained on 200 hours of Speecon data [11] with 1024 Gaussians with full covariance in the codebook ($L = 5, M^l = 1024, 0 \leq l < L$). The HMMs are context dependent and the codebook for each language is different. We have between 2000-3000 triphone models for each language, each represented by a 3-state HMM. The language model is specified as a context free grammar.

Table 2 describes the native test sets for these five languages. The test sets are all from proprietary in-car data, but some of them are cleaner than others and match the training data better. Due to this some languages have higher recognition rates than other languages. Each test set contains city names. The number of different city names in our context free grammars is specified in the fourth column of Table 2. As some city names can be repeated, the number of words can be higher than the number of entries in the vocabulary.

Table 2: Descriptions of the native test set for each language

Testset	Language	Words	Vocab.
GE_City	German	2005	2498
US_City	English	852	500
IT_City	Italian	2000	2000
FR_City	French	3308	2000
SP_City	Spanish	5143	3672

Table 3: Description of the non-native test sets

Testset	Accent	Words	Vocab.
Hiwire_FR	French	5192	140
Hiwire_SP	Spanish	1759	140
Hiwire_IT	Italian	3482	140
IFS_MP3	German	831	63

Table 3 shows the non-native test sets, mostly from the Hiwire database [34]. The spoken language in the Hiwire tests is English. The native language background of the speaker varies, as indicated in Column 2. The Hiwire test sets are as specified in the distribution of the Hiwire database and contain command and control utterances in an aeronautic scenario. The MP3 test is performed on data that was especially collected for this work and contains Italian, French and Spanish artists and song names. Depending on which information is more interesting, either the spoken languages of the test is indicated before the name, or the native language of the speakers is indicated after the name.

5. Experiments

We motivate our new approaches by evaluating the state of the art approach for multilingual speech recognition in Section 5.1. Section 5.2 shows that MWCs perform well for both native and non-native speech. We always test our systems on native speech as well as on non-native speech as we expect that many people that use for example navigation systems for foreign destination input are quite fluent in the spoken language. Therefore our system also has to recognize fluent speech of the spoken language well. Section 5.3 compares the different projection methods that we have proposed in order to reduce the exponentially increased training effort which is coming from the application of MWCs. Section 5.4 evaluates the combination of the MWC algorithm and the best projection which allows efficient recognition of any language combination on embedded systems.

5.1. State of the art

The literature review about multilingual speech recognition indicated that a global phoneme model is the preferred solution for dealing with many languages. If there is only one phoneme model, there is also only one codebook for all languages in a semi-continuous system. Thus the question arises how well a global codebook can model phonemes from different languages. Therefore we built a codebook with training data from five languages, 200 hours of Speecon data for each language. The phonemes from each language are trained with speech from the corresponding language and this global codebook. The global codebook contains 1424 Gaussians. Table 4 shows that the performance in all languages is decreased. The loss is language dependent, for example German and English suffer more than Spanish and Italian. Nevertheless, these results are sufficient for the statement that a multilingual codebook performs worse than a monolingual codebook for each language, even if it is allowed to be a little

Table 4: Comparison between monolingual codebooks and a multilingual codebook

Codebook	1024 Benchmark	1424 Multi
German	**84.1**	80.8
English	**75.5**	70.5
Italian	**92.3**	90.6
French	**76.1**	72.2
Spanish	**91.9**	91.4

larger. As the global phoneme model induces a global codebook, the same conclusions can be drawn for this approach. This conclusion is concordant with [16, 39].

5.2. Multilingual Weighted Codebooks

The performance is evaluated on German, English, Italian, French and Spanish test sets. German is chosen as main language for the MWC construction. The MWC algorithm can only take two codebooks as input. Therefore we put all Gaussians from the additional languages in a large codebook with 4096 Gaussians. Together with the German codebook this is the input to the MWC algorithm. Figure 4 shows the results of the baseline and several MWC systems. The baseline experiment uses the 1024 German Gaussians as codebook. The other systems add 200, 400 and 800 Gaussians from the additional languages. Thus, the total codebook sizes are 1224, 1424 and 1824. With these codebooks, the same retraining as for the baseline systems was performed. This means, each language got a different HMM set, and this HMM set was trained with speech from the corresponding language.

For German the benchmark and baseline systems are identical, therefore there is only one line visible in the graph. The MWC performance on the German test set varies also insignificantly. This indicates that the extensions to the codebook do not hurt the performance on the main language and is a benefit compared to the state of the art approach discussed in Section 5.1. The performance on the other tests shows that MWCs improve significantly over our baseline system. For Spanish the MWC with 1424 Gaussians almost achieves the benchmark performance. The differences between the different test sets are not relevant, as they are mainly due to the match between training and testing data, which is higher for example for Spanish and Italian, and lower for English and French.

To some extent the improvements of the MWCs can also be due to the fact that the MWCs contains more Gaussians than the baseline system. Therefore we also tested a system with an only German codebook that contains 1824 Gaussians and compared it to the MWC with 1824 Gaussians. Table 5 demonstrates that Gaussians from other languages help more than more German Gaussians for the recognition of the additional languages.

Table 6 presents the performance of MWCs on non-native accents. The benchmark system for Hiwire is the monolingual English system. For the four lingual MP3 test no benchmark performance is given, as there are utterances that contain more than one language and no monolingual system can recognize such utterances. The baseline systems and MWCs are different for each column. The reason is that it makes for example more sense to recognize Spanish accented English with a MWC that contains the full Spanish codebook.

That this is the right approach for non-native accented speech is proven by the fact that the baseline systems outperform the benchmark system significantly in all cases. In Word Error Rate (WER), the native language codebook gives actually improvements in the range of 25% relative WER, thus very similar to what the literature about non-native speech recognition could achieve without non-native adaptation data. The fact that a baseline system is better than the benchmark system can occur in these tests, as the tested speech differs strongly from the native training speech. In general the MWCs keep the performance, there are no significant improvements when additional Gaussians from other languages are added.

The absolute performance of the systems in Figure 4 and Table 6 is actually quite similar. Of course, the non-native accented speech is harder to recognize by the speech recognizer, but the vocabulary size is smaller for the non-native tests, and together these two factors lead to rather similar performance for our native and non-native tests.

To summarize, Table 6 shows that training the spoken language on native language codebooks of the speakers helps significantly for the recognition of strongly accented speakers. However, such systems do not perform well for the recognition of more fluent speakers of the language, as shown in Table 4. For such speakers, it is necessary to add some additional Gaussians to the codebook to allow a better modeling of the spoken language. These additional Gaussians do not diminish the benefit of using the native language codebook of the speakers (Table 6).

5.3. Comparison of optimal and approximated projections

There are two attributes our projection must have. First, it must be executable on the embedded system, and second, it should be as efficient as possible. Table 7 presents Word Accuracies (WA) on the native US cities task, the degree of optimality according to the distance proposed in Section 3.2 and an indication of the time needed for the projection on an Intel PC with 3.6 GHz.

Where possible, we tried to precompute elements that have to be computed only once for every language and do not depend on the actual combination. Examples are the distance between states in Projection 5 and Projection 7, as well as for the distance between all HMMs in

Table 5: Comparison of an MWC to a monolingual codebook of the same size.

	GE_City	US_City	IT_City	FR_City	SP_City
GE 1824	83.8	67.6	81.4	70.2	89.8
MWC 1824	84.3	72.0	89.7	72.9	91.0

Table 6: Word accuracies with MWCs on the non-native accented tests. All MWCs contain the full codebook of the native language of the speaker.

Codebook	Hiwire_SP	Hiwire_FR	Hiwire_IT	IFS_MP3wK
Benchmark 1024	82.5	83.9	81.6	-
Baseline 1024	86.6	86.0	86.2	60.5
MWC 1224	86.6	86.4	86.7	59.7
MWC 1424	85.7	86.1	86.0	61.3
MWC 1824	86.0	85.8	85.1	59.9

Table 7: Comparing optimal and approximated projections. The first column shows the word accuracy on the native US City test. The second column gives distances to the monolingual US English HMM models. The third column shows the runtime in seconds for precomputations. The fourth column shows the actual runtime of the estimation of the output probabilities of the HMM models. The runtime is given for the projection of one language with 1800 phoneme models to another codebook

Projection	WA	Distance L2	Precomp.	Runtime
Pro1	5.2	**4.08e-9**	330s	30s
Pro2	49.7	4.08e-9	330s	30s
Pro3	**55.5**	4.10e-9	330s	90s
Pro4	44.8	6.80e-8	2s	0.2s
Pro5	44.5	6.64e-8	12s	0.1s
Pro6	31.2	5.29e-8	4s	0.1s
Pro7	55.1	5.07e-8	14s	0.3s
Baseline	65.6	4.13e-8	-	14,400s
Benchmark	75.5	0	-	14,400s

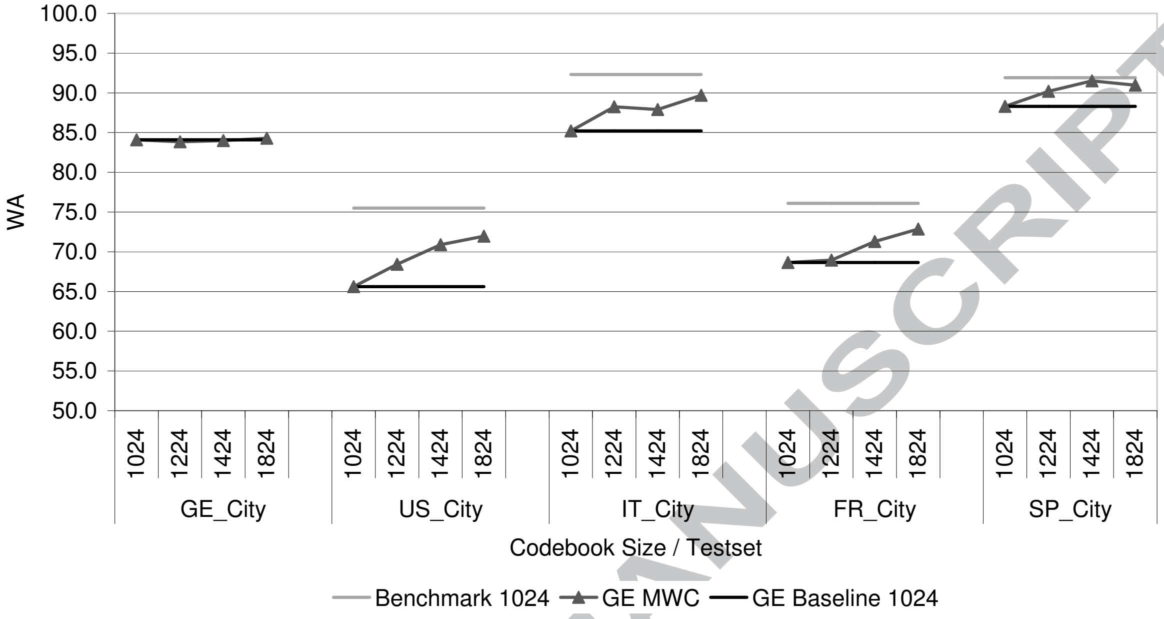


Figure 4: MWCs on native speech of five languages. All MWCs contain the full German codebook.

Projection 6. The runtime for these additional precomputations is given in column 4. For Projection 1-3 the correlations between Gaussians are precomputed.

As expected, the optimal Projections 1-3 give by far the lowest error in L2 distance. However, Projection 1 results in a weight vector for the HMM states that is so different from regular probability distributions that a standard recognizer achieves only very low recognition rates. Projection 2 adds the normalization that weights have to sum to one, and this leads already to a reasonable recognition performance. Compared to other projections it is clear that the negative weights for some Gaussians still pose a problem for the decoding. Both projection 1 and 2 are also quite slow, as the projection of each of the 5400 HMM states requires the multiplication with a large matrix.

Projection 3 gives the best overall performance, but is significantly slower than all other projections. This is due to the fact that an sequential, iterative gradient descent is performed. Furthermore, after each update of a weight all other weights are adjusted to keep the constraint that the sum equals one at every step. This is repeated three times for each weight. The total number of changes to each weight leads to the high runtime and in succession to the fact that Projection 3 is not applicable for the proposed scalable architecture.

From the approximated projections, both Projection 4 (Gaussian mapping) and 5 (State mapping) achieve good performance in spite of their simplicity. Finally, Projection 7 (combined Gaussian + state mapping) has the best overall performance with both good recognition rate and fast runtime. The results also show that the projections alone reduce the performance significantly, both compared to the benchmark and the baseline. However, for practical application the projections are an interesting alternative as they allow multilingual recognition with no additional training and decoding effort.

In the above discussion Projection 7 was used with a weight of 0.5. This combination weight was determined in a grid search where we investigated values between 0 and 1 in 0.1 steps. Figure 5 shows that a wide range of values for the combination weight are acceptable, all values between 0.3 and 0.8 led to good results.

5.4. Scalable architecture with approximated projections and MWCs

In Section 5.2 we have shown that MWCs can improve the speech recognition performance across languages. Section 5.3 demonstrated that the training complexity can be reduced with approximated projections. This section evaluates the performance of the combination of approximated projections and MWCs.

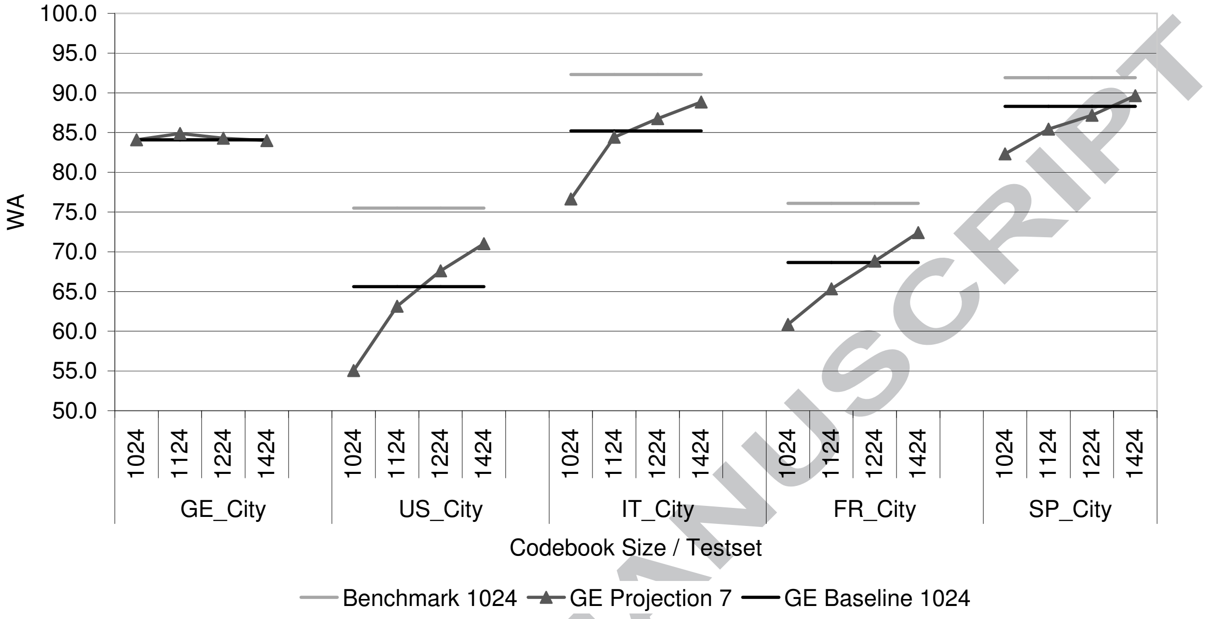


Figure 6: Scalable architecture on five different native language test sets.

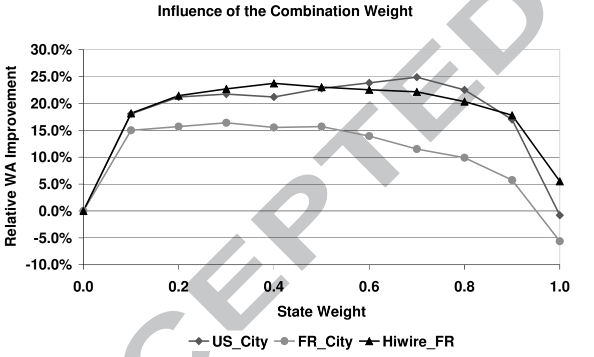


Figure 5: Effect of the combination weight in Projection 7 for three different test sets. All weight values between 0.3 and 0.8 perform significantly better than only one of the projections alone.

Figure 6 depicts that the projections are as good as a retrained system when 200 more Gaussians are used for four of the five languages. Of course, to some extent this is an unfair comparison, as we compare a system with more Gaussians to our baseline system. However, we are convinced that this is the fairest possible comparison regarding the actual behavior of our embedded target system. The reason is that the larger codebook depends on the combination of languages, in some case we may want to have 50 Italian and 30 French additional Gaussians, in other cases we would prefer to have 60 Japanese and 20 English Gaussians. It all depends on the test set, and in our scenario we first know the test on the embedded system itself. With the traditional training approach, we can not react to the different test sets by training more Gaussians for some languages. With the proposed scalable architecture, we can react and provide the right system.

Figure 7 depicts the performance of the scalable architecture on our non-native speech tests. As in Table 6, the MWCs used are actually different, and each test is tested with an MWC that contains the all Gaussians from the native language codebook of the speakers. The baseline and benchmark systems are also the same as in Table 6, which means that the upper line indicates the performance of our baseline systems. The performance of Projection 7 is significantly improved for the Hiwire tests when more Gaussians are added, but the MP3 test changes only slightly. We believe that this is due to the fact that the speakers have so few knowledge of the Italian, Spanish and French song names that they are really using German sounds to pronounce them. The figure shows that the systems generated by the scalable architecture perform slightly worse than both the benchmark and baseline systems, but given the fact that the benchmark systems require more resources for each additional language, and the training effort of the base-

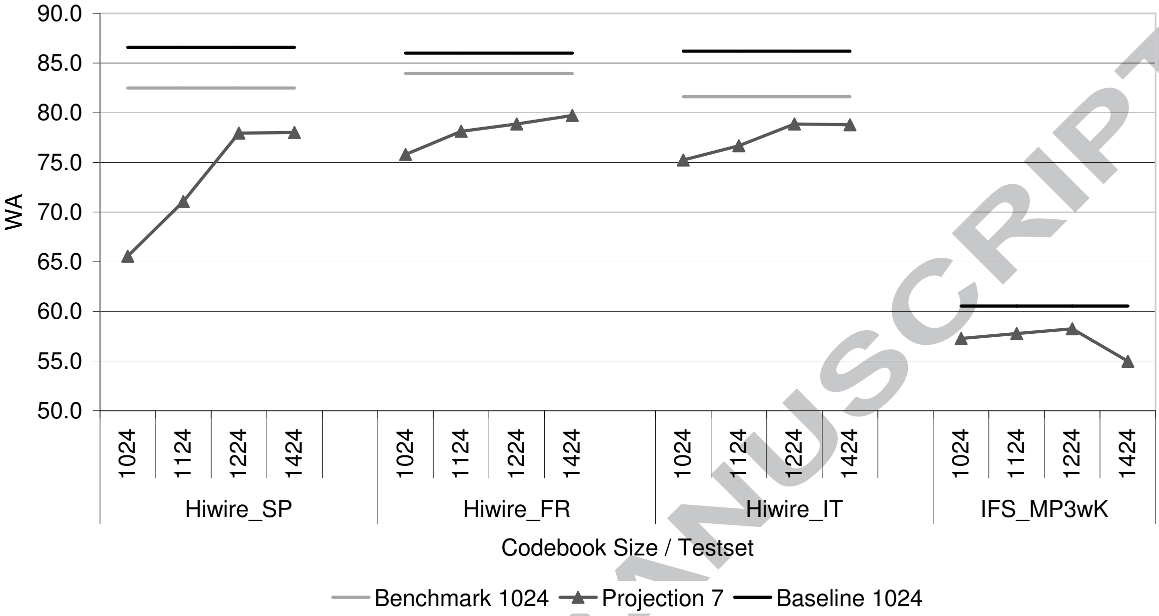


Figure 7: Scalable architecture on the non-native accented tests.

line systems increases with the number of languages, the scalable architecture is the method of choice if many language combinations are possible, and most of them will be needed rarely.

6. Conclusion

In this paper we have explained the combinatoric problems that come with the provision of multilingual speech recognition for many languages. For the efficient introduction of multilingual knowledge, we use Multilingual Weighted Codebooks that have low decoding complexity and good recognition performance for both almost fluent and less fluent non-native speakers. To keep the training effort reasonable, we have defined several projections between Gaussian spaces. From these projections, Projection 7 proved itself to be the most suitable one for speech recognition, as it is either better or faster than the other proposed projections. Though we think that in other non-speech applications the more exact L2 based projection might be more appropriate.

A combination of the proposed algorithms leads to an architecture with both low training and decoding effort. The scalable architecture outperforms our baseline systems by up to 5.4% absolute word accuracy, and performs almost similar as monolingual benchmark systems on non-native accented speech. Additionally, there are several advantages of our new scalable architecture for commercial application. First, it is customer friendlier, as it can recognize speech from all language combinations. Second, it is easier to provide and maintain due to the reduced redundancy. Third, the performance is better than that of our baseline system for fluent speakers of foreign languages. Fourth, it is cheaper, as it is not necessary to train speech recognizers for many different language combinations.

In a final comparison to the state of the art as identified in the literature we can say that our approach is more suitable if the native language of the user is known, and maximum performance in this language is paramount. In other cases, where the native language of the speaker is not known, or many speakers have to be recognized simultaneously, the global phoneme model remains the architecture of choice.

7. Acknowledgements

The authors thank Olaf Schreiner, Tobias Herbig and Dr. Volker Schubert for help with mathematical problems. Dr. Franz Gerl was a great help for the iterative solution of the optimization problem. We also thank Raymond Brueckner and Dr. Guillermo Aradilla for advice and discussion. Finally, we thank the reviewers for their valuable comments to initial versions of the paper.

References

[1] Bartkova, K., Jouvet, D., 2006. Using multilingual units for improved modeling of pronunciation variants. In: Proc. ICASSP. Toulouse, France, pp. 1037–1040.

[2] Biehl, M., Anlauf, J. K., Kinzel, W., 1990. Perceptron learning by constrained optimization: the AdaTron algorithm. In: Proc. ASI Summer Workshop Neurodynamics. Clausthal, Germany.

[3] Bouselmi, G., Fohr, D., Illina, I., 2007. Combined acoustic and pronunciation modeling for non-native speech recognition. In: Proc. Interspeech. Antwerp, Belgium, pp. 1449–1552.

[4] Boyd, S., Vandenberghe, L., 2004. Convex Optimization. Cambridge University Press.

[5] Dalsgaard, P., Andersen, O., Barry, W., 1998. Cross-language merged speech units and their descriptive phonetic correlates. In: Proc. ICSLP. Sydney, Australia, p. no pagination.

[6] Fuegen, C., 2003. Efficient handling of multilingual language models. In: Proc. ASRU. St. Thomas, USA, pp. 441–446.

[7] Goronzy, S., Sahakyan, M., Wokurek, W., 2001. Is non-native pronunciation modeling necessary? In: Proc. Interspeech. Aalborg, Denmark, pp. 309–312.

[8] Harbeck, S., Nöth, E., Niemann, H., 1998. Multilingual speech recognition in the context of multilingual information retrieval dialogues. In: Proc. TSD. pp. 375–380.

[9] Hershey, J. R., Olsen, P. A., 2007. Approximating the Kullback Leibler divergence between Gaussian mixture models. In: Proc. ICASSP. Honolulu, Hawaii, pp. 317–320.

[10] Huang, X., Lee, K. F., Hon, H. W., 1990. On semi-continuous hidden Markov modeling. In: Proc. ICASSP. Albuquerque, USA, pp. 689–692.

[11] Iskra, D., Grosskopf, B., Marasek, K., van den Huevel, H., Diehl, F., Kiessling, A., 2002. Speecon - speech databases for consumer devices: database specification and validation. In: Proc. LREC. Las Palmas de Gran Canaria, Spain, pp. 329–333.

[12] Jensen, J. H., Ellis, D. P. W., Christensen, M., Jensen, S. H., 2007. Evaluation of distance measures between Gaussian mixture models of MFCCs. In: Proc. ISMIR. Vienna, Austria, pp. 107–108.

[13] Jian, B., Vemuri, B. C., 2005. A robust algorithm for point set registration using mixture of Gaussians. In: Proc. IEEE Int Conf Comput Vision. Beijing, China, pp. 1246–1251.

[14] Juang, B. H., Rabiner, L. R., 1985. A probabilistic distance measure for Hidden Markov Models. AT&T Technical Journal 64 (2), 391–408.

[15] Koch, W., 2004. Optimierungsverfahren für einen universellen Spracherkenner mit robusten, effizienten Algorithmen. Ph.D. thesis, University Kiel, Kiel, Germany.

[16] Koehler, J., 2001. Multilingual phone models for vocabulary-independent speech recognition tasks. Speech Communication Journal 35 (1-2), 21–30.

[17] Kuhn, H. W., Tucker, A. W., 1951. Nonlinear programming. In: Proc. of 2nd Berkeley Symposium. Berkeley, USA, pp. 481–492.

[18] Ladefoged, P., 1990. The revised international phonetic alphabet. Language 66 (3), 550–552.

[19] Lang, H., 2009. Methods for the adaptation of acoustic models to non-native speakers. Diplomarbeit, Institute of Information Technology, University Ulm, Ulm, Germany.

[20] Lieb, E. H., Loss, M., 2001. Analysis. American Mathematical Society.

[21] Menzel, W., Atwell, E., Bonaventura, P., Herron, D., Howarth, P., Morton, R., Souter, C., 2000. The ISLE corpus of non-native spoken English. In: Proc. LREC. Athens, Greece, pp. 957–963.

[22] Niesler, T., 2006. Language-dependent state clustering for multilingual speech recognition in Afrikaans, South African English, Xhosa and Zulu. In: Proc. ITRW. Stellenbosch, South Africa.

[23] Noord, G., 2009. Textcat. `http://odur.let.rug.nl/ vannoord/TextCat/`.

[24] Nöth, E., Harbeck, S., Niemann, H., 1999. Multilingual speech recognition. In: Ponting, K. (Ed.), Computational models of speech pattern processing. NATO ASI Series F. Berlin, Germany, pp. 363–375.

[25] Park, J., Ko, H., 2004. Compact acoustic model for embedded implementation. In: Proc. Interspeech. Jeju Island, Korea, pp. 693–696.

[26] Petersen, K., Pedersen, M., 2008. The matrix cookbook. `http://matrixcookbook.com`.

[27] Platt, J. C., Bar, A. H., 1988. Constrained differential optimization for neural networks. Tech. rep., Caltech, USA.

[28] Raab, M., Gruhn, R., Nöth, E., 2007. Non-native speech databases. In: Proc. ASRU. Kyoto, Japan, pp. 413–418.

[29] Raab, M., Gruhn, R., Nöth, E., 2008. Multilingual weighted codebooks. In: Proc. ICASSP. Las Vegas, USA, pp. 4257–4260.

[30] Raab, M., Gruhn, R., Nöth, E., 2008. Multilingual weighted codebooks for non-native speech recognition. In: Proc. TSD. Brno, Czech Republic, pp. 485–492.

[31] Raab, M., Schreiner, O., Herbig, T., Gruhn, R., Nöth, E., 2009. Optimal projections between Gaussian mixture feature spaces for multilingual speech recognition. In: Proc. DAGA. Rotterdam, Netherlands, pp. 411–414.

[32] Schaden, S., 2006. Regelbasierte Modellierung fremdsprachlich akzentbehafteter Aussprachevarianten. Ph.D. thesis, University Duisburg-Essen, Duisburg, Germnay.

[33] Schultz, T., Waibel, A., 2001. Language-independent and language-adaptive acoustic modeling for speech recognition. Speech Communication 35, 31–51.

[34] Segura, J., Ehrette, T., Potamianos, A., Fohr, D., Illina, I., Breton, P.-A., Clot, V., Gemello, R., Matassoni, M., Maragos, P., 2007. The HIWIRE database, a noisy and non-native English speech corpus for cockpit communication. `http://www.hiwire.org/`.

[35] Tan, T. P., Besacier, L., 2007. Acoustic model interpolation for non-native speech recognition. In: Proc. ICASSP. Honolulu, Hawaii, pp. 1009–1013.

[36] Tomokiyo, L. M., Waibel, A., 2001. Adaptation methods for non-native speech. In: Proc. MSLP. Aalborg, Denmark, pp. 39–44.

[37] Uebler, U., 2001. Multilingual speech recognition in seven languages. Speech Communication 35, 53–69.

[38] Ueda, Y., Nakagawa, S., 1990. Prediction for phoneme/syllable/word-category and identification of language using HMM. In: Proc. ICSLP. Kobe, Japan, pp. 1209–1212.

[39] Wang, Z., Topkara, U., Schultz, T., Waibel, A., 2002. Towards universal speech recognition. In: Proc. ICMI. Pittsburgh, USA, pp. 247–252.

[40] Weng, F., Bratt, H., Neumeyer, L., Stolcke, A., 1997. A study of multilingual speech recognition. In: Proc. Eurospeech. Rhodes, Greece, pp. 359–362.

[41] Witt, S., 1999. Use of speech recognition in computer-assisted language learning. Ph.D. thesis, Cambridge University Engineering Department, Cambridge, UK.

[42] Zhang, F., 2005. The Schur Complement and Its Applications. Springer.