Online Generation of Acoustic Models for Multilingual Speech Recognition

Martin Raab[1,2]*, Guillermo Aradilla*[1]*, Rainer Gruhn*[1]*, Elmar Nöth*[2]

[1]Harman Becker Automotive Systems, Speech Dialog Systems, Ulm, Germany
[2]University of Erlangen–Nuremberg, Chair of Pattern Recognition, Erlangen, Germany
martin.raab@informatik.uni-erlangen.de

Abstract

Our goal is to provide a multilingual speech based Human Machine Interface for in-car infotainment and navigation systems. The multilinguality is for example needed for music player control via speech as artist and song names in the globalized music market come from many languages. Another frequent use case is the input of foreign navigation destinations via speech. In this paper we propose approximated projections between mixtures of Gaussians that allow the generation of the multilingual system from monolingual systems. This makes the creation of the multilingual systems on an embedded system possible with the benefit that training and maintenance effort remain unchanged compared to the provision of monolingual systems. We also sketch how this algorithm can help together with our previous work to have an efficient architecture for multilingual speech recognition on embedded devices.

Index Terms: speech recognition, multilingual, codebook, semi-continuous, non-native

1. Introduction

The trend for in-car infotainment and navigation devices goes towards multilingual applications such as music player control. The problem of current speech recognizer technology is that each language has its own phoneme inventory and each of these phonemes is modeled by a statistical model. Therefore processing and memory demands increase with the number of languages and render multilingual speech recognition impossible for embedded systems. This forces users to switch back to haptic control for some tasks. A speech based interface that covers 20-30 of the worlds major languages would make this superfluous in most cases.

In our case, the acoustic model is a semi-continuous HMM system based on Gaussian distributions that model the features. Unlike the more common continuous HMM, a semi-continuous HMM system forces all HMM models to use the same set of Gaussians which is usually referred to as codebook. This has the advantage that a comparably small number of Gaussians has to be evaluated for every speech frame, thus requiring low processing and memory demand. These are important aspects for systems running on embedded hardware. However, a codebook is language dependent, and we showed repeatedly that the modeling with suboptimal codebooks has significant impacts on the overall recognition performance [1, 2].

One possible approach is to use the codebooks of all languages, however this is exactly causing the parameter increase that we want to prevent as for each speech frame a 20-30 times higher number of Gaussians has to be evaluated. A frequent approach in the literature to avoid this are multilingual phoneme models. Such models are either based on the International Phonetic Alphabet (IPA, [3]) like in [4, 5], on data driven phone comparisons like in [6, 7] or on a combination of both like in [8, 9]. In all cases, a global phoneme model automatically leads to only one codebook with a semi-continuous system. In [2] we showed that using one global codebook already leads to reduced performance for all languages. This makes sense as there is only a fixed amount of parameters available and it is shared equally across all languages. However, in a car infotainment system this is an undesirable side effect. A car infotainment should always recognize the native language of the user with maximum performance as this is the language for its command and control operation.

A way to achieve the goal of maximum performance for native languages of users is to recognize all languages with the codebook of the native language of the current user. This solution has two negative effects:

- The codebook is suboptimal for the additional languages
- The training and maintenance effort increase with the number of languages squared as each language has to be trained on all codebooks

A method to tackle the first problem was presented in [1]. The presented method improves the coverage of a codebook by adding additional Gaussians from foreign language codebooks to generate Multilingual Weighted Codebooks (MWCs). The results showed significant benefits for the additional languages and no negative effects for the native language of the user. However, this amplifies the second problem, as the MWC algorithm generates new codebooks that depend on the combination of languages. Thus there are much more different codebooks on which HMMs have to be trained.

This motivated our search for an algorithm that can alleviate the provision of multilingual acoustic models. In [10] we first presented projections between mixtures of Gaussians as a solution. The basic idea is that we have monolingual acoustic models for all languages. The problem we face is that they all use different Gaussians from different codebooks which forces the decoder to evaluate all Gaussians for every speech frame. Thus, a projection that projects an HMM state that is trained on one codebook to another codebook allows the recognition of every language with every codebook. In [10] this projection was based on minimizing an L2 distance between Gaussian mixture models. However, this approach has an impractical runtime.

Ideally the algorithm should run very fast to allow the generation of the multilingual system dependent on the music collection of the user as shown in Figure 1. With this process, we can guarantee that the system covers the languages in the music collection of the user and can scale the performance for additional languages by adding more or less Gaussians to the native codebook in the MWC step. An important aspect is that all the tasks depicted in Figure 1 need to run on the embedded

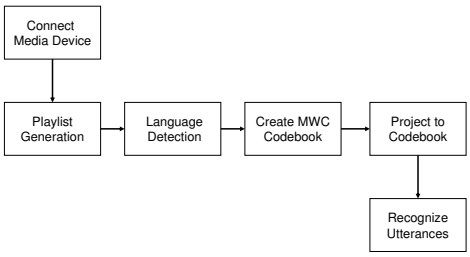

Figure 1: Generating a user adapted multilingual system on an embedded system

system to make the tailoring for every user possible. If this requirement is met, the necessary models can be generated online. This paper evaluates alternative projections to make the last step of Figure 1 possible on an embedded system.

2. Baseline system

The goal of the projections that are presented in this paper is to circumvent additional training and maintenance effort for multilingual speech recognition. They were not designed to increase recognition accuracy. Therefore, the baseline in our case is actually an upper bound for what the projections can achieve. This upper bound is a conventionally trained semi-continuous system that we extend to cover multiple language by adding additional HMMs for the phonemes of the additional languages.

- Add all additional language HMMs to the acoustic model
- Train these additional HMMs with training data from the corresponding language, not changing the codebook

An important aspect is that the codebook of the main language is not extended, as an almost infinite number of extensions is possible and a suitable extension can first be determined when the target language distribution is known. However, as shown in Figure 1 this is first known on the embedded system at runtime. To keep the numbers comparable, the modified system will also not modify the codebook in this paper, but our final system will make these modifications in order to provide good performance for all important languages (=languages that occur frequently in a users music collection). This information can be retrieved from the language detection component in Figure 1.

3. Proposed system

In this section we describe algorithms that can achieve a projection of a GMM distribution to another codebook. In our previous work [10] we have evaluated mathematically motivated projections. However, these projections were too slow for our desired online generation of multilingual HMMs. In this paper, we propose four different approximated projections as an alternative.

In all cases, the goal is to map all HMMs of all L languages to one fixed set of N Gaussians (= Recognition Codebook, RC). When we have chosen any codebook the only way to achieve such a mapping is by mapping all M^l Gaussians of each Monolingual Codebook (MC^l) to the RC. Each Gaussian $\mathcal{N}$ is represented by its mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. We map based on the smallest Mahalanobis distance (Gaussian Distance D_G). Only the covariances of the Gaussians to replace are considered, as this ensures that the distances are not affected by flat (=with large variance) Gaussians in the RC.

$$map_G(\mathcal{N}^i_{MC^l}) = \mathcal{N}^j_{RC},\ 0 \le i < M^l, 0 \le j < N, 0 \le l < L$$
$$j = \arg\min_k D_G(\boldsymbol{\mu}^i_{MC^l}, \boldsymbol{\mu}^k_{RC}, \boldsymbol{\Sigma}^i_{MC^l}) \quad (1)$$

In the introduction we have motivated that it is advisable to use the monolingual codebook from the main language as RC. This case offers 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 are mapped to main language HMMs. 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 of their probability distributions. In our system the probability distribution $p_{\mathbf{s}}$ of every state $\mathbf{s}$ is a Gaussian mixture distribution with M^l Gaussians.

$$p_{\mathbf{s}_l}(\mathbf{x}) = \sum_{i=0}^{M^l} w^i \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \quad (2)$$

The expected value of $\mathbf{x}$ for each state $\mathbf{s}$ is then

$$E(p_{\mathbf{s}_l}(\mathbf{x})) = E(\sum_{i=1}^{M^l} w^i_{\mathbf{s}_l} \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i))$$
$$= \sum_{i=1}^{M^l} w^i_{\mathbf{s}_l} \boldsymbol{\mu}_i \quad (3)$$

The covariance which is needed for the Mahalanobis distance is a global diagonal covariance $\boldsymbol{\Sigma}_{All}$ estimated on all training samples. With D_S we define our state based mapping as

$$\mathbf{map_S}(\mathbf{s}^i_l) = \mathbf{s}^j_{RS}, 0 \le i < S_l, 0 \le j < RS, 0 \le l < L$$
$$j = \arg\min_k D_S(E(\mathbf{s}^i_l), E(\mathbf{s}^k_{RS}), \boldsymbol{\Sigma}_{All}) \quad (4)$$

Based on D_S we can also 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

$$D_H(\mathbf{q}_1, \mathbf{q}_2) = \sum_{i=1}^{3} D_S(\mathbf{s}^i_{\mathbf{q}_1}, \mathbf{s}^i_{\mathbf{q}_2}) \quad (5)$$

Similar as for D_S, map_H can be defined with D_H. D_G and D_S provide consistently good performance for different tests, while they use rather different information for their calculation. Therefore we also wanted to test a combined map_{G+S}. This map is defined as

$$\mathbf{map_{G+S}}(\mathbf{s}^i_l) =$$
$$\gamma_{G+S}\, \mathbf{map_S}(\mathbf{s}^i_l) + (1-\gamma_{G+S}) \begin{pmatrix} w^1_{\mathbf{s}^i_l} map_G(\mathcal{N}^1_{MC^l}) \\ w^2_{\mathbf{s}^i_l} map_G(\mathcal{N}^2_{MC^l}) \\ \vdots \\ w^{M^l}_{\mathbf{s}^i_l} map_G(\mathcal{N}^{M^l}_{MC^l}) \end{pmatrix}$$
$$0 \le l < L, 0 \le i < S_l \quad (6)$$

with the combination weight γ_{G+S}. γ_{G+S} has to be determined in experiments. In all cases, no retraining is performed after the mapping.

4. Experimental setup

Our semi-continuous speech recognizer uses 11 MFCCs with their first and second derivatives per frame. Monolingual recognizers for English, French, German, Spanish and Italian are trained on 200 hours of Speecon data [11] with 1024 Gaussians 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. Table 1 describes the native test sets and Table 2 the

Table 1: Descriptions of the native test set for each language

Testset	Language	Speech Items	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 2: Description of the non-native test sets

Testset	Accent	Speech Items	Vocab.
Hiwire_FR	French	5192	140
Hiwire_SP	Spanish	1759	140
Hiwire_IT	Italian	3482	140
Hiwire_GR	Greek	3526	140

non-native test sets. The native tests are city names from an in-house database. The Word Accuracy (WA) differences that the results show between the languages are due to different noise conditions in the different tests.

The non-native test sets contain command and control utterances in accented English from the Hiwire database [12]. The Hiwire database was identified as the most appropriate existing and available database after a review of existing non-native databases [13]. To indicate that the language information in the test name only specifies the accent, the language information is given after the test name (in contrast to before for the different native tests). Our baseline results are in average 6% absolute WA lower than in [12], but these results were achieved with 17k Gaussians, and we have only 1024 Gaussians. The Speecon database is also noisier than the Hiwire database, but we preferred to use Speecon, as it includes similar training material for more than 20 languages.

5. Experiments

5.1. Runtime

A key aspect of the projections is their runtime as they finally have to run on an embedded system. Table 3 gives the times for the projection of an English HMM set with 1800 phoneme models to a German codebook on a Intel PC with 3.6 GHz. In order to reduce the load on the embedded device the calculation of the mappings is separated from the actual projection (=modification of the HMMs). The runtime for the precomputation of the mapping is shown in the second column. The third column shows the actual runtime of the estimation of the output probabilities of the HMM models. Only this processing time is actually needed on the embedded system. Of course, embedded systems are slower, which will increase the time. However, already current car infotainment systems have the capabilities to

Table 3: Runtime aspects of the different projections

Projection	Precomputations	Runtime
map_G	2s	0.2s
map_S	12s	0.1s
map_H	4s	0.1s
map_{G+S}	14s	0.3s
L2	330s	30s
Retrained	-	14,400s

execute these calculations in a couple of seconds. To make a comparison to our previous work possible, we also mention the runtime aspects of an projection algorithm that is minimizing an L2 distance [10]. The upper bound of a conventional retraining of HMMs is presented in the last row of the table. Apart from the fact that the in-car system has no access to the necessary speech databases, the runtime difference already forbids to execute conventional HMM trainings on embedded systems. Only with the projections an online generation of the multilingual models becomes feasible.

5.2. Combination weight

This experiment is focused on the evaluation of the influence of the combination weight γ_{G+S} for the last mapping. Figure 2

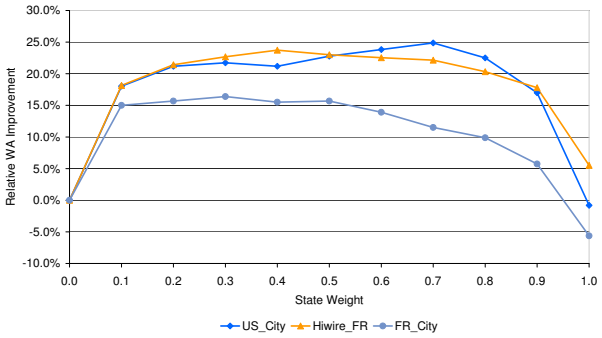

Figure 2: Influence of the combination weight in projection 7 on the performance on different test sets

demonstrates that the exact value of the combination weight is not important, all values between 0.3 and 0.7 lead to consistent improvements for the three depicted test sets. Only graphs for three test sets are visualized to keep the graph readable. The following experiments will always use a γ_{G+S} value of 0.5.

5.3. Projections on native speech

This section evaluates native language tests for five languages with German as the main language (=the German codebook is applied). Figure 3 presents the Word Accuracies (WA) on the native German, English, French, Spanish and Italian test. For the main language German nothing is changed by the mappings. For the other languages the HMMs have to be mapped from their native codebooks to the German codebook. As we explained before, the conventional retraining is the upper bound that projections can achieve. The graphs illustrates the performance of the different projections for the different test sets. One aspect is that the relative performance between the projections

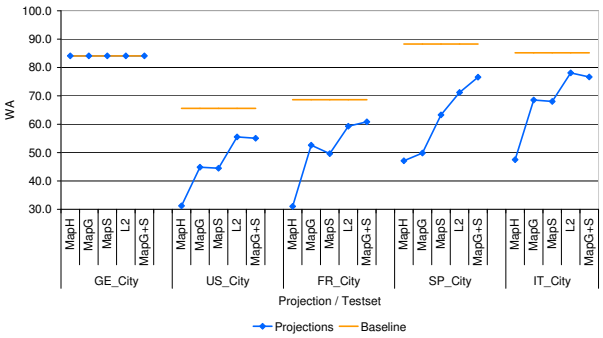

Figure 3: Performance of online generated HMMs on native speech of different languages

is independent of the language. In all languages, the HMM based mapping performs worse and $\mathbf{map_{G+S}}$ and the L2 based projection perform best. There is no clear trend whether the L2 based projection is better or worse than $\mathbf{map_{G+S}}$. Furthermore, the results show that the best projection can always achieve a performance in vicinity of the performance of the baseline. The gap in word accuracy is tolerable as our approach provides a multilingual acoustic model with optimal language distribution without the need for trainings of n^2 models. Opposed to this, the projections hardly require any additional effort for the provision of additional languages. Together with our previous work about MWCs the projection algorithm can therefore build the desired target architecture depicted in Figure 1 that allows to tailor the speech recognition system to the current needs of the user.

5.4. Projections on non-native speech

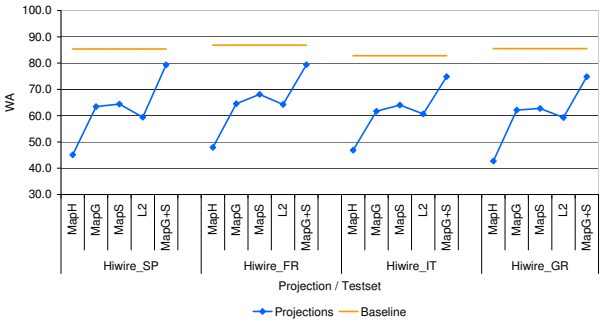

Figure 4: Performance of online generated HMMs on non-native accents of English

To complete the evaluations of the projections, the performance on accented English is depicted in Figure 4. The curves show the same tendencies as in the native case, again the HMM map is worst and the combined state and Gaussian mapping is best. The only difference is that the combined Gaussian and State mapping outperforms the L2 based projection. This strongly argues for the application of $\mathbf{map_{G+S}}$ in our final system as it is both faster and better than the L2 based projection. Furthermore, the results of $\mathbf{map_{G+S}}$ are again within the reach of the results of a conventional retraining.

6. Conclusion

At the beginning of the paper we have presented our approach how to make multilingual speech recognition on embedded devices feasible for many languages. Compared to existing approaches our new approach has the advantage that it can explicitly tailor a system for the current needs of a user. A major problem with this architecture is the additional effort at training and decoding. In this paper we have dealt with the aspect of increased training and maintenance effort for the provision of multilingual systems by projecting HMMs trained on one set of Gaussians to a new set of Gaussians. We proposed four new techniques for this task and evaluated their performance. The results showed that a projection that is based on information on the Gaussian and the state level achieves performance in the vicinity of traditional HMM trainings. However, our new approach can generate multilingual HMMs within fractions of a second from monolingual HMMs. Only due to this the whole architecture that we propose becomes realizable. That is also the reason why we refer to it as an online generation of multilingual acoustic models. Future work will include an evaluation of our full system with a combination of the MWC and the projection algorithm presented in this paper. We expect that the final system has both good performance and that it requires almost no additional training and maintenance efforts.

7. References

[1] M. Raab, R. Gruhn, and E. Nöth, "Multilingual weighted codebooks," in *Proc. ICASSP*, Las Vegas, USA, 2008, pp. 4257–4260.

[2] ——, "Multilingual weighted codebooks for non-native speech recognition," in *Proc. TSD*, Brno, Czech Republic, 2008, pp. 485–492.

[3] P. Ladefoged, "The revised international phonetic alphabet," *Language*, vol. 66, no. 3, pp. 550–552, 1990.

[4] F. Weng, H. Bratt, L. Neumeyer, and A. Stolcke, "A study of multilingual speech recognition," in *Proc. Eurospeech*, 1997, pp. 359–362.

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

[6] O. Andersen, P. Dalsgaard, and W. Barry, "Data driven identification of poly- and mono-phonemes for four European languages," in *Proc. Eurospeech*, 1993, pp. 759–762.

[7] P. Dalsgaard, O. Andersen, and W. Barry, "Cross-language merged speech units and their descriptive phonetic correlates," in *Proc. ICSLP*, 1998, paper number 482.

[8] J. Koehler, "Multilingual phone models for vocabulary-independent speech recognition tasks," *Speech Communication Journal*, vol. 35, no. 1-2, pp. 21–30, 2001.

[9] H. Lin, L. Deng, J. Droppo, D. Yu, and A. Acero, "Learning methods in multilingual speech recognition," in *Proc. NIPS*, Vancouver, Canada, 2008.

[10] M. Raab, O. Schreiner, T. Herbig, R. Gruhn, and E. Nöth, "Optimal projections between Gaussian mixture feature spaces for multilingual speech recognition," in *Proc. DAGA*, Rotterdam, Netherlands, 2009.

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

[12] J. Segura *et al.*, "The HIWIRE database, a noisy and non-native English speech corpus for cockpit communication," 2007, http://www.hiwire.org/.

[13] M. Raab, R. Gruhn, and E. Nöth, "Non-native speech databases," in *Proc. ASRU*, Kyoto, Japan, 2007, pp. 413–418.