Continuous Local Codebook Features for multi- and cross-lingual Acoustic Phonetic Modelling

Frank Diehl, Asunción Moreno, Enric Monte

TALP Research Center
Universitat Politècnica de Catalunya (UPC)
Jordi Girona 1-3, 08034 Barcelona, Spain
{frank,asuncion,enric}@gps.tsc.upc.edu

Abstract

In this paper we present a method for defining the question set for the induction of acoustic phonetic decision trees. The method is data driven resulting in a continuous feature space in contrast to the usual categorical one. We apply the features to a multilingual speech recognition task, outperforming consistently the standard method using IPA-based characteristics. An extension to cross-lingual applications together with first preliminary results are given too.

1. Introduction

A central question in the design of an automatic speech recognition (ASR) system is the definition of proper acoustic phonetic entities. This question gets even more important when trying to share the acoustic features space among different languages, or to share it with a third, yet unseen language.

The common state of the art approach for defining the acoustic models is the use of a phonetic decision tree. Such a tree constitutes a functional mapping from a feature to a model domain. Therefore, defining for all phonetic circumstances of the input domain a proper hidden Markov model (HMM) in the output domain. One crucial topic of this mapping function is the definition of the input domain. A standard approach is the use of phonetic features assigned to the phonemes. This usually works quite well, but exhibits the problem that the design depends on phonemical knowledge. In case of a multi- or even cross-lingual system design this might be a severe problem. Not only knowledge for one but for all languages is needed. Additionally, this knowledge should be comparative. This is in strong contrast to the information found in phonetic dictionaries and textbooks assigned to one specific language. Information and transcriptions given there usually follow the principle of phonological contrast [1] using a broad transcription. This leads to the problem that equal phonetic features may be assigned to clearly distinguishable phonemes of different languages.

To cope with these problems [2] suggests a feature construction by bottom-up clustering of the acoustic information given by the HMMs seen in a database. Constructing phonetic broad classes based on a phoneme confusion matrix is proposed in [3].

In [4] and [5] local similarities between the probability density functions of HMMs are identified, and used for constituting phonetic features. For identifying the similarities, advantage is taken of the prototype character of the mixture components forming the probability density functions (PDF) of semicontinuous HMMs (SCHMM).

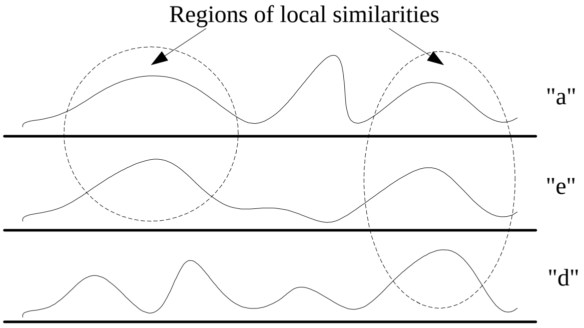

Figure 1: *Local codebook similarities.*

In this work we extend this concept of so called ”local codebook features” (LCB) to a method for constructing continuous phonetic features in a low dimensional feature space. We also work out the conceptual base for their use in cross-lingual tasks and give simulation results.

The paper is organized as follows. In section 2 continuous LCB-features are presented, followed by section 3 extending the concept to cross-lingual use. Section 4 discusses the question generation. In section 5 a system overview is given followed by section 6 with the test set up. Test results are presented in section 7 and the conclusions are given in section 8.

2. Continuous local codebook features

The basic idea behind LCB-features [4], [5] is that similar phonetic properties should be reflected by similar shaped probability density functions (PDF) on a local scale. In Figure 1, this idea is depicted. It shows the assumed probability density functions of the phonemes ’a’, ’e’, and ’d’, with two locally similar regions. The similarities might be caused by common articulation properties of the phonemes as e.g. ’voiced’ or ’open’. Constructing features based on this idea means to identify such similarities in the PDFs of HMMs. In case SCHMMs this can be efficiently done by comparing the prototype weights of the PDF’s mixture components.

For the sake of simplicity, we assume during derivation of the method, that the beforehand trained incontextual HMMs have only one state and refer to only one codebook. In the following, this allows to leave out the state and codebook indices. Hence, the PDF

This work was granted by the CICYT under contract TIC2002-04447-C02.

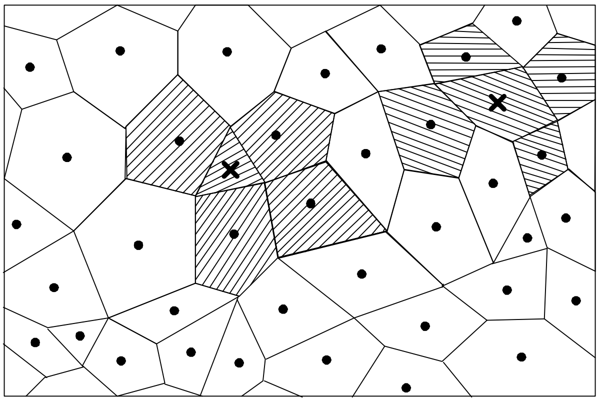

Figure 2: *Locality and neighborhood. The picture shows* 2 *localities '**x**' with neighborhood* $\tilde{L} = 5$, .

of one SCHMM is given as

$$G_{mix}(i) = \sum_{l=1}^{L} c_{il} \cdot G(\mu_l, \cdot) \qquad i \in \{1, ..., I\}, \tag{1}$$

with L the codebook size and i the HMM index. $G(\mu_l, \cdot)$ names the l^{th} mixture component with mean vector μ_l. For the local search, we define a locality just by a mixture component, and therefore by every index $\tilde{l}$ out of the L mixture components. We also define the neighborhood of $\tilde{l}$ as the $\tilde{L}$ mixture components closest to the locality $\tilde{l}$.

Figure 2 depicts the concept for a two dimensional case. It shows a part of a codebook with class regions and the mean values (black dots). Two localities are accentuated by marking the corresponding means by crosses instead of dots. The neighborhood is set to $\tilde{L} = 5$, and the equivalent regions with the five closest mean values to the localities are drawn hatched.

For a formal derivation of the method, we define the distance $d(\tilde{l}, l)$ between the mixture components by equation (2)

$$d(\tilde{l}, l) = \|\mu_{\tilde{l}} - \mu_l\| \qquad l, \tilde{l} \in \{1, ..., L\}. \tag{2}$$

Identifying for each locality $\tilde{l}$ the $\tilde{L}$ closest neighbors is done by evaluating equation 2 for all l and $\tilde{l}$. As result we get for each locality $\tilde{l}$ an index set $S_{\tilde{l}}$ naming the indices of the $\tilde{L}$ closest mixture components of locality $\tilde{l}$. Using $min^{(n)}$ to signify the "n^{th} smallest value of" we can express $S_{\tilde{l}}$ as

$$S_{\tilde{l}} = \left\{ l \mid \arg \min_{1 \le l \le L}{}^{(n)} d(\tilde{l}, l), \quad n \in \left\{1, ..., \tilde{L}\right\} \right\}, \tag{3}$$

where $\tilde{l} \in \{1, ..., L\}$. Applying these index sets to the PDF of each HMM i by summing up the corresponding weights of the codebook mixture components, we get local, cumulative probability masses $m^*_{\tilde{l}i}$. Grouping all $m^*_{\tilde{l}i}$ together, we derive matrix

$$M^*_{\tilde{L}} = [m^*_{\tilde{l}i}]_{LxI}. \tag{4}$$

In $M^*_{\tilde{L}}$ the subindex $\tilde{L}$ names the neighborhood size used to construct the features.

Note that each of the matrix's column name one incontextual HMM, and each row stands for one locality, centered around one mixture component $G(\mu_l, \cdot)$, in the associated probability space. Therefore, local similarities between different phonemes are reflected by similar cumulative probability masses $m^*_{\tilde{l}i}$ in one row of $M^*_{\tilde{L}}$. We also mention, that for $\tilde{L} = 1$ the matrix $M^*_{\tilde{L}}$ reduces just to the B-matrix, i.e. the original mixture weights of the underlying HMMs.

Matrix $M^*_{\tilde{L}}$ can already be used for training an acoustic-phonetic decision tree. Each column of $M^*_{\tilde{L}}$ constitutes a vector in the definition space of the tree. Different properties between phonemes are reflected by different distance relations between the column vectors of $M^*_{\tilde{L}}$.

A problem associated with the use of $M^*_{\tilde{L}}$ is its dimensionality. In case of SCHMMs a codebook usually consists of several hundred kernels. In our case we use $L = 256$, fixing also the column length of $M^*_{\tilde{L}}$ and therefore the dimension of the input space. Besides being cumbersome to handle in practice, the high dimension is also in contrast to what should be necessary to describe basic acoustic-phonetic properties respective differences between phonemes. When working with standard IPA-features, feature vectors consist typically of less than 10 components. We therefore assume that the underlying dimensionality of the problem should be around 10.

To cope with the request of a reduced dimensionality some kind of dimension reductions is needed. The first choice would be a singular value decomposition (SVD) of matrix $M^*_{\tilde{L}}$ and its subsequent reduction by projecting its entries to the space spanned by its strongest principal components. Unfortunately, dimension reduction by SVD distorts the transformed space. That is, it does not preserve distance relations between transformed vectors, actually constituting the information content of $M^*_{\tilde{L}}$.

Therefore, for dimension reduction simultaneously trying to preserve the relative distances between elements we propose a nonlinear multidimensional scaling technique, the Sammon mapping [6]. The original Sammon mapping is defined as a projection of a set of points from a high dimensional into a low dimensional space (e.g. of dimension 2), under the criterion of maintaining the mutual relative distance relationships between the points.

As objective function to minimize, the cumulative error E

$$E = \frac{1}{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \delta^*_{ij}} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{(\delta^*_{ij} - \delta_{ij})^2}{\delta^*_{ij}} \tag{5}$$

between the element distances δ^*_{ij} in the original space and the corresponding element distances δ_{ij} in the lower dimensional, transformed space is defined.

Distance δ^*_{ij} is taken as the euclidean distances between the vectors m^*_i and m^*_j. The summation runs over all vectors, i.e. $n = I$. A corresponding definition applies for δ_{ij} in the transformed space. Minimizing E with respect to the transformed vectors m_j is done by gradient descend

$$E^{(\tau+1)} = E^{(\tau)} - \mu \nabla_{m_k} E^{(\tau)}, \qquad k = \{1, ..., I\}. \tag{6}$$

In general, dimension reduction can not be done without introducing a certain amount of residual mapping error. The most commonly used criterion to measure this error is *Kruskal's stress*:

$$S = \sqrt{\frac{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} (\delta^*_{ij} - \delta_{ij})^2}{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \delta^2_{ij}}}. \tag{7}$$

In section 4 we give typical values of S.

To match better the needs of the subsequent decision tree construction, we extended the classical Sammon mapping slightly. First, for convenience, we center the transformed vectors around the origin by subtracting their mean after each gradient descend iteration. Second, the final vector space $M_{\tilde{L}}$ is aligned to its principal components. This operation constitutes just a change of the coordinate system. Neither the first nor the second operation change the distance properties between the elements $M_{\tilde{L}}$, and therefore do not affect the mapping.

At this point we have to emphasize that the constructed features are ordered, continuous and not categorical as the commonly used IPA-features or phonetic broad classes. Therefore we name them ”continuous LCB-features”.

Continuous LCB-features constitute a major change in constructing an acoustic-phonetic decision tree. Instead of partitioning the input space of the decision tree according to categorical questions as e.g. ”Is the left context of the model a vowel?”, we now partition the input space by continuously shifting hyperplanes parallel to the axes through it. I.e. a typical question is of the form: ”Is the third component of the model's feature less than -1.795?”.

3. Cross-lingual continuous LCB-features

In case of a cross-lingual application we try to regress models for a new language by the regression tree constructed for some other languages. In contrast to model regression using IPA-features, we need a small development database of the new language to derive LCB-features from. Basically this is the same process as described in section 2. Nevertheless, it is crucial that the resulting features are consistent with the LCB-features used to construct the original decision tree.

We start by training incontextual SCHMM for the new language using the multi-lingual codebook of the other languages. By this the resulting new matrix $N^*_{\tilde{L}}$ is consistent with the original matrix $M^*_{\tilde{L}}$. Directly applying Sammon's mapping on $N^*_{\tilde{L}}$ is not possible. This is due to the fact, that the resulting transformation depends on its initialization and on the data points to transform. Just running the Sammon mapping would correctly preserve the distance relations between the entries in the new matrix $N^*_{\tilde{L}}$, but would be completely inconsistent with the original features $M_{\tilde{L}}$.

To overcome the problem we propose a constraint Sammon mapping fitting the new data points given by $N^*_{\tilde{L}}$ into the original, transformed map $M_{\tilde{L}}$. That is, we run a Sammon mapping for the compound features $\left[M^*_{\tilde{L}} N^*_{\tilde{L}}\right]$ initializing the process randomly for the $N^*_{\tilde{L}}$ and by the originally transformed features $M_{\tilde{L}}$ for the $M^*_{\tilde{L}}$. Consequently, the low dimensional result space contains right from the beginning the correctly placed points $M_{\tilde{L}}$. Gradient descend is performed only respective the new feature vectors $N_{\tilde{L}}$, but with equation 5 referring to all feature vectors. I.e., we fit the new feature vectors n_i into the old, fixed vectors m_i, trying to preserve the mutual distances within the new features and between the new and the old features.

4. Question generation

For model definition we use a binary decision tree [7] splitting nodes according to a binary question and an entropy based impurity measure. In the classical approach, a binary question is composed out of phonetic attribute values assigned to the SAMPA representations of the phonemes we use in our system. The attribute values are taken from corresponding IPA descriptions [1] of the phonemes.

In case of LCB-features, we started by training incontextual models (3 states) for each language. This is followed by extracting the features as described in section 2. The features are based on common multi-lingual mel-cepstrum coefficients (MFCC) codebooks. The neighborhood is varied between $\tilde{L} = 1$ and $\tilde{L} = 36$. With a codebook size of 256 and 122 phonemes, 47 for German, 44 for English and 31 for Spanish, the intermediate matrix $M^*_{\tilde{L}}$ is of dimension 256x122. Sammon mapping is performed to dimension 10, i.e. $M_{\tilde{L}}$ is of dimension 10x122. *Kruskal's stress* results to lie in the range of $3 - 6\%$. A clipping of matrix $M_{\tilde{L}}$ is presented in table 1. These features are directly overtaken for constructing question by the decision tree.

Table 1: *Example of LCB-features for* 4 *phonemes.*

Phoneme					
2:	2.177	-0.786	-3.199	2.566	...
6	1.680	0.447	2.749	-2.175	...
9	-2.066	-1.169	2.562	1.833	...
a:	1.609	-2.587	-1.795	1.721	...

State tying is done without a-priori distinction between different base phones. That is, we build one tree for each state positions but each tree performs the model definition over the whole phoneme set.

5. System overview

We use a SCHMM system computing every 10ms twelve mean free mel-cepstrum coefficients (MFCC) (and the energy) over a 25ms frame. First and second order differential MFCCs plus the differential energy are employed. For each sub-feature, a codebook is constructed consisting of 256 and 32 (delta energy) Gaussian mixtures, respectively. Common multi-lingual codebooks are used.

Acoustic phonetic modelling is done by demiphones, [8]. They can be thought of as triphones which are cut in the middle giving a left and a right demiphone.

According to the demiphone concept, LCB-features are constructed on the last state of the left and the first state of the right incontextual demiphones. This leads to two independent features spaces, one for the left and onefor the right demiphones.

6. Test set up

Training and testing the systems are performed using SpeechDat-I/II fixed telephone databases. Four languages are used. For Spanish (S), English (E), German (G) a 1000 speaker training and a 400 speaker test part is extracted. Slovenian (V) serves as target language for the cross-lingual experiments. To get a Slovenian baseline recognition rate also a pure Slovenian system is build using a 900 speaker training and a 100 speaker test set.

For training, we use $7500 - 8100$ phonetically-rich sentences per language. Testing is done using phonetically-rich words mixed with application words. For Spanish, German and English approx. 2600 and for Slovenian 614 phrases apply. The resulting grammars, just word lists, consist of approx. 1300 words for Spanish, German and English and 372 words for Slovenian.

For the cross-lingual regression of multi-lingual models for a Slovenian system a subset of 200 speaker, 1740 phrases, out of the 900 speaker training set is used. First, context independent Slovenian models are trained. In a second step LCB-features are constructed. With these features the multi-lingual decision tree is entered and models are regressed.

7. Tests and test results

First we compare multi-lingual German, English, Spanish models defined by common IPA-features with corresponding multi-lingual models defined by continuous LCB-features. Table 2 presents the corresponding word error rates (WER). In case of LCB-features the used neighborhood sizes $\tilde{L}$ is given too.

Table 2: *Multi-lingual test results, WER* [%].

	$\tilde{L}$	S	G	E
IPA	-	7.41	9.48	27.96
LCB	1	6.13	8.81	27.53
LCB	6	6.77	8.96	27.13
LCB	12	6.35	9.19	26.98
LCB	18	6.81	8.78	26.82
LCB	24	6.58	8.89	26.86
LCB	30	6.62	9.19	26.98
LCB	36	6.77	8.78	27.13

Table 2 shows that IPA-features are always outperformed by LCB-features. No clear conclusion respective the optimal neighborhood size can be drawn. Variations in the WER due to a different choice of $\tilde{L}$ seem to be of statistical nature and are within the error bars of approx. $1.3\% - 2.4\%$. We note that some of the best results are already achieved with a neighborhood of $\tilde{L} = 1$. That is, just using the dimension reduced original B-matrix as definition space for the decision tree performs very well.

The next test series focus on the cross-lingual use of already available models. Multi-lingual models trained on Spanish, English and German are used as source models for a recognition task in Slovenian. We start by building two Slovenian baseline systems. The first system, IPA$_{mono}$, is a pure Slovenian one. Model definition is done using standard IPA-features for decision tree construction. The system named LCB$_{mono}$ is the mono-lingual LCB-feature counterpart. Table 3 shows that for both systems the recognition rates are almost equal.

Table 3: *Slovenian mono-lingual test results, WER* [%].

	IPA$_{mono}$	LCB$_{mono}$
WER	9.61	9.60

At this point we need to say that the LCB$_{mono}$ system is not completely mono-lingual. For building the system we actually used the multi-lingual codebook used for constructing the Spanish, English, German models. In other words, the influence of the codebook mismatch is negligible. Furthermore, to get a more comparable setup to the subsequent cross-lingual tests, also the LCB-features are build with the multi-lingual codebook. That is, the Slovenian monophone models needed to extract the features space are trained on this codebook. Finally, also with respect to the cross-lingual case, we used 200 speakers out of the 900 speakers of the Slovenian training set to build the LCB-features. This was done in order to resembles a cross-lingual task, by assuming a small database for the target language.

In summary we conclude that also for Slovenian the LCB-feature approach works very well. The harm of constructing the LCB-features with a reduced database using the multi- instead of a mono-lingual Slovenian codebook barely effects the performance of the ASR system.

Finally we present preliminary results for the cross-lingual step. As baseline an IPA-feature based model set, IPA$_{cross}$ is constructed by regressing Slovenian IPA-features through an usual IPA-base multi-lingual decision tree. For the LCB-case LCB-features are build according to the procedure described in section 3. For construction we use the same 200 speaker subset of the Slovenian database as used for the LCB$_{mono}$ system. Table 4 shows the disappointing results of this test. On the one hand side we have the IPA$_{cross}$ WER benchmark being with 46.09% in a region also reported by other groups [9]. On the other hand we get very high error rates using the LCB-approach.

Table 4: *Slovenian cross-lingual test results, WER* [%].

	IPA$_{cross}$	LCB$_6$	LCB$_{12}$	LCB$_{18}$
WER	46.09	84.20	85.99	76.71

The results from table 4 are in strong contrast to the good findings for LCB-features in the mono- and multi-lingual cases. At the moment it is not clear what causes this bad behavior. A possible explanation might be some kind of database missmatch causing an offset between the original Spanish, English, German and the cross-lingual Slovenian LCB-features. Further investigation is needed to clarify this question.

8. Summary and Conclusions

In this work we provide a framework for the data-driven construction of so called continuous LCB-features. We showed their usefulness for model definition in several tasks. For the multi-lingual Spanish, English, German and the mono-lingual Slovenian case, model definition based on LCB-features outperformed consistently the standard, IPA-based approach.

We also provided the technical base for cross-lingual feature construction and some initial results. Further research is necessary to understand and overcome the problems in the final cross-lingual model regression.

9. References

[1] The International Phonetic Association, *Handbook of the International Phonetic Association,* Cambridge University Press, The EdinBurgh Building, Cambridge CB2 2RU, UK, 2003 edition, 1999.

[2] K. Beulen et al, "Automatic question generation for decision tree based state tying," in *Proc. ICASSP'98, Seattle, USA.*

[3] A. Žgank et al, "Data driven generation of broad classes for desicion tree construction in acoustic modelling," in *Proc. EUROSPEECH'03, Geneva, Switzerland.*

[4] F. Diehl and A. Moreno, "Local codebook features for mono- and multilingual acoustic phonetic modelling," in *Proc. AST'04, Maribor, Slovenia.*

[5] F. Diehl and A. Moreno, "Quasi-continuous local codebook features for multilingual acoustic phonetic modelling," in *Proc. ICASSP'05, Philadelphia, USA.*

[6] J. W. Sammon, "A nonlinear mapping for data structure analysis," in *IEEE Transactions on Computers, Vol. 18, 1969.*

[7] F. Diehl and A. Moreno, "Acoustic phonetic modelling using local codebook features," in *Proc. ICSLP'04, Jeju, Korea.*

[8] J. B. Mariño et al, "The demiphone: an efficient subword unit for continuous speech recognition," *Proc. Eurospeech'97,* pp. 1215–1218, Rhodes, Greece.

[9] A. Zgank; Z. Kacic; K. Vicsi; G. Szaszak; F. Diehl; J. Juhar; S. Lihan, "Crosslingual transfer of source acoustic models to two different target languages," in *COST278 Workshop on Robustness Issues in Conversational Interaction, 2004.*