Data-driven Pronunciation Modeling for ASR using Acoustic Subword Units

Thurid Spiess*†, Britta Wrede‡*, Gernot A. Fink*, Franz Kummert*

*Faculty of Technology, Bielefeld University, Germany,
†Institute of Computer Science, Augsburg University, Germany,
‡International Computer Science Institute, Berkeley, USA
tspiess@techfak.uni-bielefeld.de

Abstract

We describe a method to model pronunciation variation for ASR in a data-driven way, namely by use of automatically derived acoustic subword units. The inventory of units is designed so as to produce maximal separable pronunciation variants of words while at the same time only the most important variants for the particular application are trained. In doing so, the optimal number of variants per word is determined iteratively. All this is accomplished (almost) fully automatically by use of a state splitting algorithm and a variant distance measure. Compared to a baseline system using triphones as subword units and with minimal pronunciation variants, this method achieved a relative improvement of the word error rate by 10%.

1. Introduction

Pronunciation variation is still one of the main reasons for automatic speech recognition to produce poor results (see [1] for an overview and a listing of different phenomena). This is due to variations in the speech signal of one and the same speaker, depending on factors such as the particular situation, speaking style, emotional state or external influences, and between different speakers, depending e. g. on dialect, social background, sex or age. All those factors prohibit a one-to-one correspondence between acoustic signal and symbolic representation.

Still, most actual speech recognition systems do not take pronunciation variants into account and model only one way of pronunciation for every word in the lexicon. This is because adding variants for a word increases the chances of confusability between lexicon entries and enlarges the search space.

The methods to model pronunciation variation for ASR can be divided into knowledge-based and data-driven methods. The latter extract the information on pronunciation variation from the data whereas the former assume that the information is already available (e. g. in pronunciation dictionaries). Presently, none of these two classes of methods can be entirely preferred over the other [1]. The approach described here belongs to the data-driven class as information on pronunciation variants is obtained exclusively from the acoustic signals. At the same time, it aims at further automating the process of designing an ASR system by building up lexical entries from automatically derived subword units. Previous approaches with similar objectives are [2] and [3].

We model pronunciation variation on the lexicon level by allowing multiple lexical entries for one word. The acoustic model is optimised in respect to pronunciation variants by using specifically trained, namely acoustic subword units (ASUs). These units are derived automatically from the acoustic signal by grouping similar regions of the acoustic space. They stand thus in contrast to phonetic subword units such as phonemes, which are defined knowledge-based. No pronunciation modeling is performed on the language model level. This is because the speech corpus used for testing does not use a language model. It would be easy, however, to integrate the variants with standard techniques [1].

The proposed system starts with some initial variant number per word. Then, optimal model topologies for this number of variants are determined by use of a state splitting algorithm. Words are ranked according to a variant distance measure that indicates the similarity resp. the dissimilarity of the variants of the words. Based on this ranking, the variant numbers of words with dissimilar variants are increased by one while the variant numbers of words with similar variants are decreased by one. This procedure is iterated until no more variants are dissimilar or some other stop criterion is met.

Section 2 gives an overview of the system with special focus on the two main aspects of the design, the state splitting algorithm used to determine optimal model topologies, and the variant distance measure. Section 3 presents experiments with different configurations for the system parameters and the environment in which they were conducted. Finally, some concluding remarks are given in section 4.

2. System outline

In this section we describe the steps involved in building the system. The sequence of steps is shown in Fig. 1.

First of all, words applicable for data-driven modeling have to be chosen. The acoustic subword units are word-dependent, so not all words are applicable but only those with a minimal occurence count. Their initial number of variants is set to two.

Afterwards, the system design operates in two main steps which can be iterated until a stop criterion is met. The first step consists in training a HMM (hidden Markov model) for every variant of a word. Model parameters are initialized by distributing randomly all occurences of the word in the training corpus on its variants. Then, a state splitting algorithm (see 2.1) determines the optimal number of states for the HMMs. Parameter re-estimation is performed with the Baum-Welch algorithm. The states of the HMMs can be regarded as the acoustic subword units which are thus word-dependent.

In the second step the similarity between the variant models of every word is calculated using a special distance measure (see 2.2). The number of models is then decreased by one for words with similar variants, while it is increased by one for words with dissimilar variants. Then again, step 1 is executed, and HMMs

This work was partially funded by a grant from the DFG in the graduate program 256 'Task-oriented Communication'.

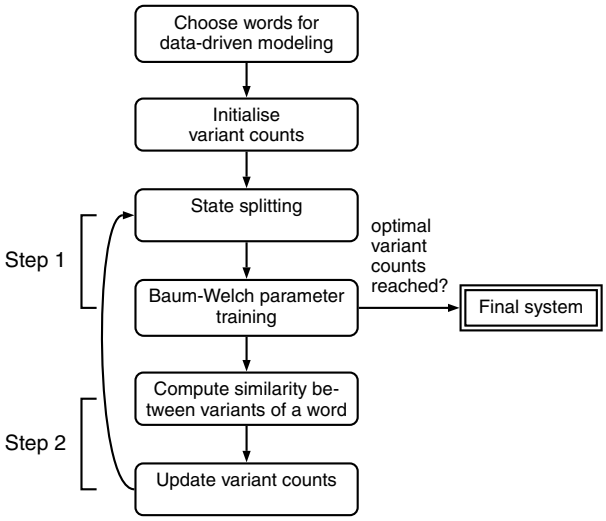

Figure 1: *System overview.*

are trained for the variants of every word. The process is repeated until some stop criterion is met. This can be either that no more variants are dissimilar or that a predefined maximum number of variants is reached.

This procedure should find the most differing pronunciation variants for every word and thus not increase the acoustic confusability significantly. By controlling the number of variants at the same time, the search space is not excessively enlarged. Recognition time and WER (word error rate) improvement support the method while also a higher degree of automation is reached.

2.1. State splitting algorithm

A good topology of the models is crucial to the performance of the system. Since the models develop their characteristics during the iterative parameter training, the model topology as well should be determined iteratively. Therefore, a state splitting algorithm is employed to find the optimal topology. Starting with an initial number of states in a model, such an algorithm splits states according to a split criterion after a predefined number of training iterations. This is repeated until some stop criterion is met.

Thus, the initial number of states in a model, the splitting criterion and the stop criterion have to be defined.

We use $0.1 * f_{min}$ as the initial number of states where f_{min} is the number of frames in the shortest occurence of the word in the training corpus. f_{min} is a rough clue on the optimal number of states, so $0.1 * f_{min}$ is definitely an underestimation of the latter.

After two iterations of Baum-Welch parameter re-estimation all states chosen by the splitting criterion are split. This criterion should select states which model very inhomogenous regions of the acoustic space. Entropy as well as transition probability of a state are hints for such regions. States with high entropy apparently capture a high degree of acoustic information. However, this could also be caused by one and the same phone with high variations. A high self-transition probability indicates that the state models a long time span which contains probably lots of information. Though, this can be sensible when the feature vectors in this time span are very similar. So, entropy or self-transition probability alone are no secure indicators of a good state to split, while a combination of both ensures that states modeling long *and* inhomogenous acoustic regions are split.

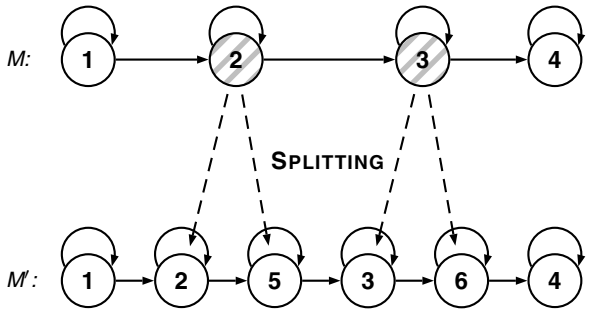

Figure 2: *State splitting algorithm: states with high entropy and high self-transition probability are split. Initial state parameters are copied.*

The thresholds used for high entropy and high self-transition probability are 3 and 0.7 respectively. These values have been determined empirically.

States chosen for splitting are duplicated. After the following training iterations, their parameters differ as they come to model different acoustic regions.

For the convergence of the number of states 5 iterations of state splitting turned out to be sufficient. The definition of a state deleting criterion has been considered but found not to be necessary because it was never applied before the 5th iteration.

2.2. Variant distance measure

The number of variants per word should comply with maximal separability. Thus, a measure has to be found to compute the similarity resp. the dissimilarity between the models of a word's variants.

The models of the variants differ in the number of states and in the states themselves. So, first an alignment of the states of the two models has to be found. To this end, we use the edit distance (e. g. [4]): A sequence of states is transformed into another by substituting one or more states in the first sequence by one or more states in the second sequence. The cost of the alignment is the sum of the distance between aligned states. The edit distance is then the alignment with minimal cost. For our purposes, we need to modify the edit distance slightly, because computing the edit-distance as the sum of the distances between aligned states leads to generally high values for long models and low values for short models. Since similarity of models should be independent from their length, we do not take the whole cost as distance value, but the average of the two largest distances between aligned states taken from the alignment with minimal cost.

The next step is to define a state distance measure. HMM states differ in their transition probabilities and in their output probabilities. Since transition probabilities are not significant here concerning the difference of states, we rely on the output probabilities. We use semi-continous HMMs, so all states share the same Gaussians differing only in their mixture weights. The mixture weights c_i of a state describe a discrete probability distribution with $\sum_i c_i = 1$. So, in principle, every distance measure for discrete probability distributions can be employed here as state distance measure.

A good measure for the distance between distribution A and distribution B is the cross-entropy or Kullback-Leibler dis-

tance:

$$\delta_{KL}(A,B) = \sum_{i=1}^{I} a_i \log(\frac{a_i}{b_i}) \quad (1)$$

where $A = (a_1, a_2, \ldots, a_n)$ and $B = (b_1, b_2, \ldots, b_n)$. It has one drawback, namely, it is not symmetric. Still, it can be made symmetric by averaging $\delta_{KL}(A,B)$ and $\delta_{KL}(B,A)$. Following [5] we decided to use the harmonic mean and call it from now on the resistor average distance. Ignoring constants and monoton functions (absolute values are not of importance here) we get

$$\delta_R = \frac{\delta_{KL}(A,B)\delta_{KL}(B,A)}{\delta_{KL}(A,B) + \delta_{KL}(B,A)} \quad (2)$$

Another possibility is the following version of the symmetric cross-entropy proposed by Kullback and Leibler [6], relating distributions A and B to the joint distribution $A + B$:

$$\delta_S = \delta_{KL}(A, A+B) + \delta_{KL}(B, A+B) \quad (3)$$

Finally, the Bhattacharyya distance is another well-known distance measure for probabilities:

$$\delta_B(A,B) = -\log \sum_{i=1}^{I} \sqrt{a_i b_i} \quad (4)$$

These measures are now used in order to modifiy the definition of the edit distance by averaging over the two most different pairs of states of two compared models.

$$D_\delta(M_1, M_2) = \max\{\frac{\delta(S_i,T_j) + \delta(S_k,T_l)}{2} | S_i \neq S_k \vee T_j \neq T_l\} \quad (5)$$

where (S_i, T_j) and (S_k, T_l) are aligned pairs of states in an alignment with minimal cost for the models $M_1 = S_1 \ldots S_{N_1}$ and $M_2 = T_1 \ldots T_{N_2}$ and δ is one of the distance measures described above.

All variants of a word are compared with (5) and the word receives the distance between the closest pair of its variants as a score. For the distance measures (2) – (4), these scores depend strongly from the particular system configuration and the training iteration. Consequently, it is difficult to define absolute thresholds separating words with similar variants from those with dissimilar ones. This is why, currently, words are ranked by the score and then a predefined percentage of them is considered as similar, the rest as dissimilar.

3. Evaluation

3.1. System configuration

For an evaluation of the presented approach the ESMERALDA (Environment for Statistical Model Estimation and Recognition on Arbitrary Linear Data Arrays) [7] speech recognition system was used. In the baseline system words are modelled by a sequence of semi-continuous HMMs modeling context-dependent triphones. In general each triphone model consists of two or three states, thus imposing a minimal duration of 2 or 3 frames respectively on each phone. For each word one canoncial pronunciation model was given. The codebook holds 512 shared Gaussians with diagonal covariances. The features consist of 12 standard MFCC (mel frequency cepstral coefficients) plus energy and first and second order derivatives giving a 39-dimensional feature vector. They are computed over a window of 16 ms length which is shifted over the signal with a frame rate of 10 ms. No language model was used.

Training and evaluation were performed on the German SLACC (Spoken LAnguage Car Control) Corpus [8] which consists of short utterances for the control of non-safety relevant functions in a car. The read speech was recorded in different cars with a far-field microphone mounted to the front jamb. In total 22 different speakers were recorded from which 4 were kept separately for the test set. Thus, the training set consists of 18 speakers reading 9207 utterances which give a total duration of about 9.5 hours. In the test set the data of the remaining 4 speakers is used which gives 1787 utterances and a total of around 2 hours.

3.2. Recognition Results

Various settings for the system parameters were tried on the 1787 utterances of the test corpus. In order to ensure the trainability of variants only words with a minimum occurence count of 50 were modeled with acoustic units. This resulted in 95 words. As the corpus is relatively small, the maximum number of variants per word was set to 3 in order to have enough training examples for every variant.

Different configurations for the variant distance measure and the percentage of words defined as having similar/dissimilar variants were tested. For the variant distance measure, (2) – (4) were employed as state distance measures. Beginning with 2 variants for each of the 95 words, in the next step 3 variants were trained for $P_1\%$ of these words with the most dissimilar variants while the remaining $(100 - P_1)\%$ obtained only one variant. In the third and last step, $P_2\%$ of the words with 3 variants kept their number of variants while for $(100 - P_2)\%$ of them which had the most similar variants, only two variants were trained. Thus, in step III less variants were trained than in the previous step II.

The following percentages were tested:

- $P_1 = P_2 = 33$,
  assumption: the majority of words has no variants;
- $P_1 = P_2 = 50$,
  assumption: half of the words has variants;
- $P_1 = 66, P_2 = 50$,
  assumption: the majority of words has variants.

Results for these configurations are shown in Table 1.

δ	P_1	Step II (WER)	P_2	Step III (WER)
R		**20.7**		21.3
S	33	20.9	33	21.3
B		22.6		21.5
R		21.6		22.4
S	50	22.6	50	22.9
B		22.5		21.7
R		22.2		24.2
S	66	24.1	50	23.7
B		23.3		22.9

Table 1: *Recognition results (WER) for different state distance measures (R=resistor average distance, S=symmetric Kullback-Leibler distance, B=Bhattacharyya distance) and percentages P_1, P_2 for similar/dissimilar variants.*

Subsequently, we compared the best system of Table 1 (i. e. the system with the resistor average distance as state distance

measure and $P_1 = 33$ in step II) to the knowledge-based baseline system in terms of recognition accuracy and decoding time (see Table 2). The baseline system uses triphones as subword units and has only a single pronunciation for every lexical entry, except for the word "zwei" (*two*), which is defined with two variants, namely `/tsvaI/` and `/tsvo:/`. Apart from that, configurations are as given in section 3.1. This system is used in practice and can be considered as the standard system. To show the effect of pronunciation modeling alone, we also trained a system using acoustic subword units for the 95 most frequent words but with no variants. Finally, to demonstrate the impact of both pronunciation modeling and acoustic subword units, WER and decoding time of a system with triphones and no variants at all is also given.

System	WER	Time
triphones, no variants	23.2	0:58h
triphones, variants for "zwei"	22.4	1:06h
ASUs, no variants	21.6	0:56h
best system with variants and ASUs	20.7	1:15h

Table 2: *Comparison with baseline systems. WER and decoding time for the test corpus.*

3.3. Discussion

As Table 1 shows, a system with the resistor average distance yields the best performance overall. However, we cannot simply draw the conclusion that the resistor average distance is the best suited distance measure. One reason for this is that this performance was unexpectedly achieved by a system that should rather not have optimal variant counts. Furthermore, it was observed that, when using the resistor average distance, the order of the ranked word scores depends strongly on the random distribution of the variants on the occurences of the corresponding words when initializing state parameters.

An advantage of the symmetric Kullback-Leibler distance is that it yields only slightly worse performance than the resistor average distance without sharing the disadvantages mentioned above. Still, for both distances, a deterioration from step II to step III can be observed, whereas using the Bhattacharyya distance, the WER improves in the last step. The latter one, however, does not achieve as good overall results as the other two, so no distance measure can be considered to be absolutely superior over the others. One reason for this could be that the optimal ratio for the variant counts was not found, which leads to the assumption that even more improvements of the WER could be possible with this method. In order to realise this more experiments are necessary.

Comparisons with the baseline systems also prove the potential of the method. Acoustic subword units alone lead to a relative improvement of 3.6% of the WER compared to the standard system. Adding pronunciation modeling leads to a further 6.5% improvement. This means, that the approach presented here reduces the WER of the standard system by almost 10%, which is a significant improvement. The need for at least slight pronunciaton modeling is evident looking at the relatively poor performance of the system with triphones and no variants at all.

At the same time, employing acoustic subword units does not degrade recognition time, as figures in Table 2 show. Pronunciation modeling, however, leads to an increase in time, though this is still a justifiable increase considering the WER improvements.

4. Conclusion

We have shown that the proposed approach of addressing the problems of pronunciation variation and subword unit selection in a single approach is able to reduce the WER by up to 10%. This indicates that a significant degree of pronunciation variation occurs at a sub-symbolic level which escapes those approaches where symbolic manipulations are used to model variation. For example, variations which occur due to deletions of whole phonemes might not be captured well by the phonemic SWUs which represent canonical pronunciations. This effect is expected to be even stronger in spontaneous speech.

In detail, the results indicate that the increase in performance is not simply due to a higher number of parameters since the best results were achieved with a system that only trained variants for the smallest subset of words (with $P_1 = 33$ - cf. table 2). However, reducing the number of variants reduced the performance in almost all cases which indicates that a reduction of parameters does have a negative effect in certain cases. This also indicates that the proposed system is very sensitive to some of the arbitrarily chosen factors.

Thus, in future work it would be desirable to have a framework that provides a data based fixation of the parameters. In particular it would be desirable to provide a data-driven threshold to distinguish between similar and dissimilar variants of a word as well as to determine if a state should be split or not.

In order to allow new words to be modelled by ASUs it would be necessary to find a mapping between ASUs and a symbolic representation. This could be achieved for example by specifying the conditional probabilities of ASUs given a certain grapheme.

5. References

[1] H. Strik and C. Cucchiarini, "Modeling pronunciation variation for ASR: A survey of the literature," *Speech Communication*, vol. 29, pp. 225–246, 1999.

[2] T. Holter and T. Svendsen, "Combined optimisation of baseforms and model parameters in speech recognition based on acoustic subword units," in *Proc. IEEE Workshop on Automatic Speech Recognition*, 1997, pp. 199–206.

[3] R. Singh, B. Raj, and R. Stern, "Automatic generation of subword units for speech recognition systems," *IEEE Trans. on Speech and Audio Processing*, vol. 10, pp. 89–99, 2002.

[4] D. Gusfield, *Algorithms on Strings, Trees, and Sequences*, Cambridge University Press, New York, 1997.

[5] D. Johnson and S. Sinanovic, "Symmetrizing the Kullback-Leibler distance," submitted to IEEE Trans. on Information Theory, http://cmc.rice.edu/docs/docs/Joh2001Mar1Symmetrizi.pdf, 2002.

[6] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, pp. 76–86, 1951.

[7] G. A. Fink, "Developing HMM-based recognizers with ESMERALDA," in *Lecture Notes in Artificial Intelligence*, Václav Matoušek, Pavel Mautner, Jana Ocelíková, and Petr Sojka, Eds., Berlin Heidelberg, 1999, vol. 1692, pp. 229–234, Springer.

[8] C. Schillo, "Das SLACC Korpus," Tech. Rep., Faculty of Technology, Bielefeld University, 2001.