SIGNAL DRIVEN GENERATION OF WORD BASEFORMS FROM FEW EXAMPLES

Andreas Hauenstein
pc-plus GmbH
Munich
Germany
E-mail: hauensteina@acm.org

ABSTRACT

The work described in this paper attempts to automatically generate word baseforms as used in the pronunciation dictionaries of large vocabulary speech recognition systems. The input to the algorithm consists of several sample utterances per word. No additional information, like e.g. word spelling, is used. The task involves determining a suitable inventory of subword units (SWU) as well as determining the baseforms themselves.
Experiments show that improvements over a triphone based dictionary are possible with less than ten sample utterances per word if test and training vocabularies are different. A possible application would be a system based on a fixed inventory of HMM-models that needs to be adapted to different vocabularies.

1. INTRODUCTION

Large vocabulary speech recognition systems usually use pronunciation dictionaries to describe the composition of words from subword units (SWU). Popular examples of SWUs are phones and triphones. On the phone level, the dictionary entries (also called baseforms) are either designed by humans or are generated by a grapheme to phoneme converter. This paper describes an approach to find word baseforms by looking at sample utterances. Experimental results with a speaker-independent, continuous speech recognition system suggest that the approach is both useful and practicable. There are two main requirements that a recognition system that uses baseforms derived from sample utterances must fulfill to be practicable:

- The number of required sample utterances per word must be small (say, <10) to avoid tedious and expensive speech recording sessions.
- If there are no examples at all for a word, there must still be a way to add the word to the recognition vocabulary.

2. WORDMODEL QUANTIZATION

An overview of the complete training process is given in the following figure.

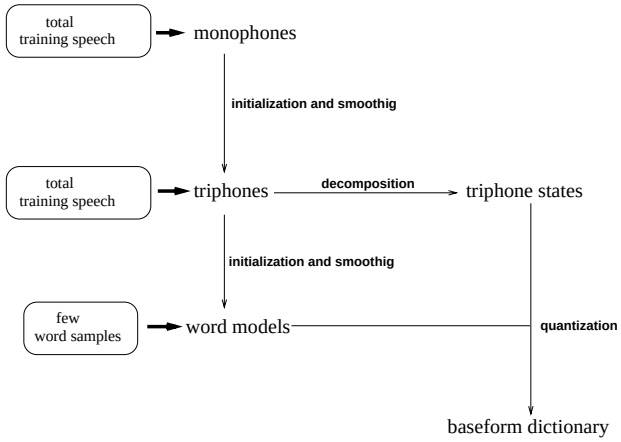

Figure 1: Overview of training process

The basic subword unit in the dictionary is the *triphone state* rather than the triphone itself. In the experiments, a set of 1353 triphones was used. Since each triphone was modeled by a three-state HMM, this results in 4059 triphone states. Using the triphone models, good initial word models can be constructed. Theses models are then retrained using a few word utterances and smoothed by interpolating with the original triphone parameters. Each state in the resulting word model is then mapped to the closest state in the set of 4059 triphone states. A similar approach was proposed in [3]. However, the experiments reported there were restricted to a vocabulary of 24 words and required over 100 sample utterances per word to build a new baseform. The experiments performed by the author showed that both the training process for the word models and the distance metric used when mapping the word model states to the triphone states are extremely crucial. Depending on the choices made, results varied between hopeless and practicable. The following three are the most important requirements:

3. DISTANCES

The emission probability of a feature vector in a state with a single semicontinuous codebook of size M is described by the formula

$$f(\vec{x}) = \sum_{k=1}^{M} p_k f(\vec{x}|c_k) \quad ,$$

where p_k is the kth trained weight in a state and $f(\vec{x}|c_k)$ is the kth gaussian density in the shared

codebook. For distance measurements between states we only look at the weights p_k. With this simplifying assumption, it is sufficient to discuss the distance computation between two sets of weights $P = (p_1, \cdots, p_M)$ and $Q = (q_1, \cdots, q_M)$.

A popular choice for a distance within the speech recognition community is the *information loss*:

$$d_{loss}(P,Q) = H(P \oplus Q) - \frac{n_p}{n_p + n_q} H(P) - \frac{n_q}{n_p + n_q} H(Q)$$

where H(P) denotes entropy, n_p is the training count for P and

$$(P \oplus Q)_i = \frac{n_p p_i + n_q q_i}{n_p + n_q} \quad .$$

A little thought reveals that this distance has the strange property that distributions with small training counts are close to *any other* distribution. If used for quantization purposes, most retrained word model states will be mapped to the least trained elements in the set of triphone states. Results are catastrophic. Of course, the exact same property makes d_{loss} a good choice for clustering purposes.

Three more suitable candidates are *directed divergence*, *euklidean distance*, and *absolute distance:*

$$d_{div} = \sum_{i=1}^{M} p_i \log p_i - \sum_{i=1}^{M} p_i \log q_i \quad ,$$

$$d_{euk} = \sum_{i=1}^{M} (p_k - q_k)^2 \quad ,$$

$$d_{abs} = \sum_{i=1}^{M} |p_i - q_i| \quad .$$

Before applying these distances to the quantization task, they need to be generalized to suit the multi-stream architecture of the recognition system. In a two stream system, emission probability is

$$f(\vec{x}) = \sum_{k=1}^{M} p_k f(\vec{x}|c_k) * \sum_{k=1}^{M} \tilde{p}_k f(\vec{x}|\tilde{c}_k) \quad ,$$

where the $\tilde{p}_k$ are the weights for the second stream, e.g. Δ-coefficients. Experiments show that it is absolutely important not to ignore this information during state quantization. For the generalizations of d_{div} and d_{euk} to multiple streams, efficiently computable formulations can be derived exactly, e.g.

$$d_{div} = \sum_{i=1}^{M} p_i \log p_i + \sum_{i=1}^{M} \tilde{p}_i \log \tilde{p}_i - \sum_{i=1}^{M} p_i \log q_i - \sum_{i=1}^{M} \tilde{p}_i \log \tilde{q}_i$$

For d_{euk} something similar is possible, but the notation is somewhat messy, so it is omitted here. For d_{abs} things are not so easy, so an approximate substitute will have to do:

$$d_{abs} \approx \sum_{i=1}^{M} |p_i - q_i| + \sum_{i=1}^{M} |\tilde{p}_i - \tilde{q}_i| \quad .$$

In experiments, the differences between these three choices in terms of word accuracy achieved with the resulting baseforms was not remarkable. However, the experiments also showed that the generalization to include all streams was essential. When only one stream was considered during quantization, accuracy degraded in an unacceptable way. It is quite common practice to make similar simplifying assumptions (like unimodality) during distance computation, e.g. in works on distribution sharing. It might be well worthwhile to invest some effort into developing efficient and sufficiently exact distances for continuous mixture distributions.

4. THE RECOGNITION SYSTEM

This section describes the basic structure of the recognition system used for the experiments.

As acoustic features 12 MFCC coefficients were used as computed by the HTK ([4]) toolkit. Frame length was 30ms, with 100 frames per second. These features were augmented by 12 first order delta coefficients, energy and Δ-energy, resulting in a feature vector of dimension 26.

These features are modeled as three streams which are assumed to be independent of each other: The MFCC - coefficients themselves, the delta coefficients, and the energy with its first derivative. The features from each stream are quantized using one semicontinuous codebook per stream. The covariance matrices used in the codebooks were diagonal matrices. This assumption is acceptable if multiple streams are used.

Triphones and monophones were modeled by three state Bakis type HMMs. Each baseform is terminated by a one state model supposed to account for interword effects.

The above design decisions result in a system with a structure very similar to the one described by Huang ([2]). Note that apart from the MFCC coefficients, the HTK toolkit was not used. The system is described in more detail in [1].

5. EXPERIMENTS

The test setup used 10960 utterances by 97 speakers to train the monophones and triphones (training set A). The vocabulary size of the training set was 1486. The recognition vocabulary contained 365 words, some of which occured in the training vocabulary. Word models were trained on utterances of the 365 test words, as cut from continuous speech (training set B). The baseforms resulting from quantizing these models onto triphone states were used during recognition on the test set C, which uses the same vocabulary as set B. Note that the training set consists of two parts: One domain independent, large general purpose training set to train triphones, and a small set of sample utterances from a

specific application vocabulary. The test set C contained 200 sentences uttered by a speaker who had no utterances in set A or set B. No language model was used, so perplexity was 365.

5.1. Triphone Training

As a performance baseline, a fairly generic triphone approach was chosen. All within-word triphones occurring in training set A more than 40 times were assumed to be sufficiently trained. Their parameters were interpolated with those from the corresponding monophones to provide robustness. Triphones occurring less than 40 times were assumed to be undertrained and were replaced by their center monophones. This results in a word accuracy of 66.3 % on set C. This rather unsatisfactory performance is caused by the change in vocabulary. On the same test set, accuracy increases to 81% if triphone training is based on set B. This discrepancy illustrates the well known fact that triphones are particularly suited to setups where test and training vocabulary are identical.
The question now is by what margin the 66.3% can be improved without changing any HMM parameters, just by tuning the pronunciation dictionary.

5.2. Wordmodel Training

The wordmodels trained on set B are the basis for the automatically generated baseforms. Their training has to be done carefully to guarantee good results with few examples.
Before training starts, wordmodels are constructed as sequences of the appropriate triphones. The word samples cut from set B are then used for retraining. In a final step, the resulting model parameters are interpolated with the parameters as they were before training started. Accuracy was measured without smoothing (NS), with total smoothing (TS, = triphones), with 50% smoothing (HS) and with parameter interpolation weighted by the training count for the parameters (WS). The results are displayed in the following table:

#samples	NS	TS	HS	WS
all	60.0	66.3	86.0	82.1
5			61.2	

This shows that unsmoothed training actually degrades performance, even though *all* available word samples were used. The number of samples per word ranged from at least 15 up to about 100 for the most frequent words. Weighted smoothing offered no advantage over the simplistic 50% approach, which yielded 86% accuracy. When monophones are used for interpolation and initialization, accuracy only slightly degrades to 84.9 % (not in the table).
When the number of training utterances is restricted to five samples per word chosen with a random number generator, word model accuracy is only 61.2 %.

The robustness of semicontinuous models is important when using few examples. With discrete models and 5 samples per word, accuracy collapses to 49.8 % from 61.2 % in the semicontinuous case, even though the same smoothing techniques are employed. This huge difference almost vanishes as the number of samples increases.

5.2. Distances

In section 4, several ways to compute the distance between distributions during quantization were suggested. The following table compares the distances in terms of word accuracy when they are used for baseform generation.

abs	div	euk	euk 1
74.1	73.9	74.7	56.6

While there is little difference between the distances that use all available information, it becomes clear that the restriction to one stream (column euk 1) is not acceptable.

5.2. Samples per Word

Figure 2 shows both the recognition accuracy of the wordmodels themselves and of the resulting baseforms, depending on the number of sample utterances used for retraining.

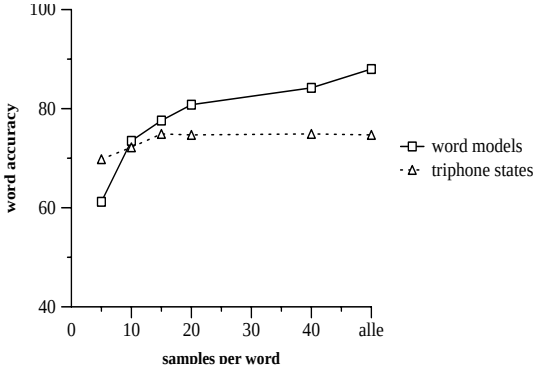

Figure 2: Recognition accuracy depending on number of training samples.

The surprising result is that while more sample utterances per word have a strong effect on the quality of the resulting word model, this is no longer true for the baseform obtained by the state quantization process. When only five sample utterances per word are used, word accuracy even increases after quantization. The recognition accuracy using the normal triphone baseforms was 66.3%. Even with only 5 examples, 69.8% accuracy results, as can be seen in the figure.
Using more than 15 sample utterances does not help a lot any more. A limit is reached at 75% accuracy. This might indicate that the recognition rate is limited by the quality of the triphones rather than the baseforms in the dictionary.

5.3. Example Baseforms

An example of a triphone baseform and a baseform resulting from the automated process might serve to make some of the above explanations more understandable. The notation we use for triphones is
`l_m_r`
which stands for the phone /m/ preceded by /l/ and followed by /r/. The special symbols `^` and `$` denote the beginning and the end of a word, respectively. The 'phone' /ssil/ is the interword model with one state.
When referencing triphone states, the state number (starting at 0) is appended to the triphone notation. So
`l_m_r_0`
is the first state of the above triphone.

With this notation, the triphone baseform for the german word 'Zug' (meaning train) is:

```
^_t_s t_s_u s_u_k u_k_ssil k_ssil_$
```

Note that all triphones occured frequently enough in the training material, otherwise monophones would have been substituted.
The somewhat lengthy baseform for the same word after quantization is:

```
n_t_ssil_0 n_t_s_1 l_t_s_2 t_s_U_0
t_s_u_1 t_s_u_2 s_u_k_0 s_u_k_1
s_u_k_2 u_k_ssil_0 u_k_ssil_1
u_k_ssil_2 u_ssil_$_0
```

This example nicely illustrates some of the typical properties of the automatically generated baseforms. First of all, the center phone almost never differs from the center phone in the original triphone dictionary. Just the left and right context varies, as in the first phone of the example.
More remarkably, the state number (0-2) almost never changes, i.e. the beginnings of phones are mapped to the first state of some triphone, the center to state 1, and the end almost inevitably to a state `x_x_x_2`, although there is no restriction whatsoever in the algorithm.

6. CONCLUSION

The wordmodel-quantization approach presented above was successful in a setup where the domains of test and training, i.e. the vocabulary or the frequency distribution of words in the vocabulary, was different. This is a situation where a triphone based dictionary does not guarantee the usual performance.
The important design decisions are summarized by the following points:

- The word model training must use semicontinuous HMMs. Discrete models are not robust enough, and continuous models do not have simple and exact distance measures.
- The parameters obtained after training a word model must be interpolated with the original triphone parameters, even when several hundred training utterances are used.
- The distance metric used when mapping word model states to triphone states must consider all distribution parameters. If e.g. the emission probabilities for the Δ-Parameters are omitted from distance computation to gain speed, the quality of the resulting baseforms degrades considerably.

Experiments showed that a substantial gain in recognition accuracy (66.3% → 72.2%) was already obtained with ten randomly chosen sample utterances per word. Even with five samples per word, there was still noticeable improvement. This is quite remarkable, since the whole word models used to create the baseforms in this case had less performance than the models built from the resulting state sequences. It seems that the quantization onto the well-trained inventory is able to compensate some effects of undertraining.

We have thus met the two requirements mentioned in the introduction: It is possible to improve the pronunciation dictionary with a reasonably small number of sample utterances. Because the underlying inventory of SWUs does have a phonetic interpretation, it is still possible to enter new words into the vocabulary, even if no sample utterances at all are available, simply by using a triphone representation. This baseform might later be improved online as samples of the new word come in.

However, in other experiments not reported here, it was not possible to beat the performance of triphones for identical test and training domains. This is neither astonishing nor really a drawback of the approach, since automatically generated baseforms are most useful in applications were new words have to be integrated into the vocabulary.

7. REFERENCES

[1] A. Hauenstein, “Aussprachewörterbücher zur automatischen Spracherkennung”, Infix-Verlag, Sankt Augustin, 1996, ISBN 3-89601-133-2

[2] X.D. Huang, Y. Ariki and M.A. Jack “Hidden Markov models for speech recognition”, Edinburgh University Press, Edinburgh, 1990, ISBN 0-7486-0162-7

[3] Mei-Yuh Hwang and Xuedong Huang, *“Subphonetic modeling with markov states - senone”*, Proc. ICASSP’92, pp. 33-36, San Francisco, 1992.

[4] S.J.Young *“The HTK hidden markov toolkit: design and philosophy"* TR 152, Cambride University, Cambridge, 1994