A Low-Cost Phonetic Transcription Method

Pablo Fetter *Udo Haiber* *Peter Regel-Brietzmann*

Daimler-Benz AG, Research and Technology, Wilhelm-Runge-Str. 11,
D-89081 Ulm, Germany
e-mail: fetter@dbag.ulm.daimlerbenz.com

ABSTRACT

In this paper our goal is to find the phonetic transcription of spoken utterances. We present a method which uses information extracted directly from the word-based search to compute the most likely phoneme sequence. Utterances are transcribed during recognition, so that the phonetic representation of the input is available after the search. Using this method, the computational cost of the word-based search remains almost unaltered, and the phonetic transcription is obtained almost for free.

1. INTRODUCTION

Phonetic transcriptions of spoken speech are necessary in a number of speech recognition issues. They can be used to incorporate new words into the system vocabulary, or to compute confidence measures, to name just two possibilities.

Most of the previous research on transcribing new words [1, 2, 6] and on using transcriptions to compute confidence [8] have the following characteristics in common:

- Two separate search processes are used (one for the word-based search and one for the phoneme-based search), and consequently,
- the computational cost of the phonetic transcription is high.

In the current study, we present a phonetic-transcription method which uses information extracted directly from the word-based search to compute the most likely phoneme sequence. Thus, a second search is avoided, and little additional computation is required for the phoneme transcription.

We will next describe our transcription approach, and its implementation in the Daimler-Benz large-vocabulary continuous-speech recognition system. We will then present some experimental results and general conclusions.

2. THE "CHEAP TRANSCRIPTION ALGORITHM"

The cheap transcription algorithm aims at obtaining an accurate phonetic transcription of the spoken utterance. In the framework of a word recognizer, this method is "cheap" because it is embedded in the word-based search.

In the Daimler-Benz speech recognition system [4], the lexicon is implemented as a tree for efficiency reasons. This tree is compiled off-line and can be directly accessed and traversed during decoding. In the tree, every node is a (pointer to an) HMM state. During recognition, all active paths are expanded in every frame so that the static lexicon structure becomes a highly dynamic, growing tree (see Figure 1 for an example). Here, the search for the best path is performed by means of the Viterbi algorithm.

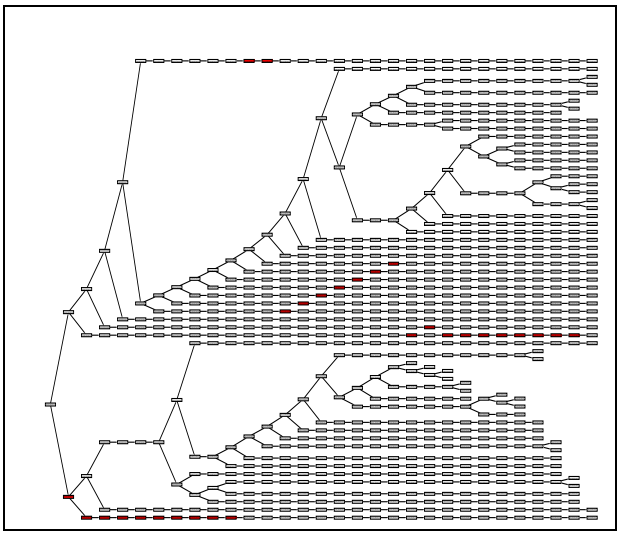

Figure 1: The search space of the first 30 frames (0.3 sec) in a Verbmobil sentence. Every column in the graph represents a frame and every node an HMM state. The nodes marked in black correspond to the best path in the current frame.

Keeping this picture in mind, it is intuitively clear that the phonetic information encoded in the HMMs is accessible during the lexical search. As mentioned before, every node in the lexicon tree represents one state of an HMM. And every HMM is again one context-dependent subword unit. So for phonetic transcription we store the best lexicon node in every

Phoneme chain:

```
U  o: U  o: o: o: o: o: k k k k k k k k k k e: e: e: e: e: e: e: e: e: e: e: e: e: e: e: e: e: e: e: f
```

Smoothed phoneme chain:

```
o: o: o: o: o: o: o: o: k k k k k k k k k k e: e: e: e: e: e: e: e: e: e: e: e: e: e: e: e: e: e: e: e:
```

Smoothed and collapsed phoneme chain:

```
o:--------------------k-------------------e:-----------------------------------------------------
```

Figure 2: Smoothing and collapsing frame-based chains. In this example, the word *okay* (transcribed `/?o:ke:/` in German) was uttered. The first row shows the phonemes extracted in each frame during the search. The second row shows the smoothed phoneme chain. In the third row, the smoothed chain was collapsed into the string `/o:ke:/`. In this case, the glottal stop (`?`) was not recognized.

frame (more on the meaning of "best lexicon node" below). Afterwards, we map the context-dependent subword unit into the corresponding context-*in*dependent model, which is one in a set of 36 German SAMPA phonemes[1] in the system used for this study. At the end of the decoding process, we obtain a chain consisting of as many phonemes, as (10 msec) frames in the utterance. In order to obtain the actual phonetic transcription, this chain is first smoothed and then collapsed into a set of phonemes. Figure 2 shows an example of this method.

We must emphasize that the algorithm ignores some of the constraints and knowledge sources that make word-based recognition possible at all. The most obvious knowledge source ignored is the lexicon. But also the constraints imposed to the search by the topology of the HMMs are ignored: Whereas a word hypothesis has to fully traverse all HMMs it includes in order to survive, the phonemes hypothesized by the cheap transcription algorithm do not.

Moreover, the effectiveness of the algorithm is restricted by the influence of two characteristics of the system: the vocabulary size and the beam width. The influence of the vocabulary size is obvious. Think of the extreme case when the lexicon contains only one word: The transcription would be constrained to the phonemes present in this one word, regardless of what was actually said. But in large vocabulary systems, the high phonemic diversity present in the lexicon lessens the importance of this drawback. The influence of the beam width is less evident. The beam width is the interval between the path with the highest probability and the pruning threshold. If the beam width is reduced to zero, that is, if the only path allowed is the best one, the transcription would be constrained to the phonetic representation of that best path. But since in large-vocabulary, continuous-speech recognition systems the beam width *has* to be fairly wide to achieve good recognition results, this problem is, again, of less practical importance. In spite of these drawbacks, the benefits of the method are so appealing that it is worthwhile investigating it further.

[1]The phoneme set used for this study includes the following symbols:

```
2: 6 9 ? @ C E I N O S U Y a b d e: f g h i: j k l m n o: p r s t u: v x y: z
```

3. EXPERIMENTS

We carried out experiments using the Kiel Corpus of Spontaneous Speech (Vol. I) [3]. The corpus is a subset of the Verbmobil corpus, and contains 21 dialogs in 383 turns (approx. 40 min of speech). The material was manually segmented and labeled at the phoneme level, and can therefore be used as a reference to match the recognizer's frame-wise phoneme transcriptions. For our tests we used a subset of the corpus, comprising 199 turns and approx. 17 min of speech.

First, we tested our recognizer using the official vocabulary of the Verbmobil Evaluation 1996 [5]. The vocabulary was enhanced with noise models and pronunciation variants, and contains approx. 6 000 entries. The OOV rate of the test set was 1.47%. A bigram language model was used during the search. On the test set we measured a perplexity of 51 and a word accuracy of 82.57%. In separate recognition runs, additional information intended to be used for phonetic transcription was extracted from the search. Two different parameters were investigated:

- the phonemes belonging to the best path in every frame (BPF), and
- the phonemes with the highest (product of HMM transition and emission) probability in every frame (BMF).

In both cases, no measurable difference in the run time was observed with respect to the original system.

3.1. Frame-Based Experiments

The two phoneme chains investigated were compared to the reference chains on a frame basis. Every frame with a hypothesis different from the spoken phoneme was counted as an error[2]. The Frame Error Rate (FER) was calculated as follows:

$$\mathrm{FER} = \frac{\mathrm{Count(Err)}}{\mathrm{Count(Ref)}} \qquad (1)$$

where Count(...) represents the number of frames containing incorrect phoneme hypotheses (Err) and the total number of frames in the test set (Ref).

Table 1 shows the results. As can be seen, the phonemes that belong to the best path per frame (BPF) provide the

[2]A few frames were not considered for evaluation, because the references were labeled with symbols for diphthongs which we do not consider as a separate acoustic model. Also, silence and non-speech segments in the references were skipped.

best transcription. This outcome is somehow intuitive: The best path per frame has a degree of inertia, that compensates the lack of constraints mentioned earlier. On the other hand, the best instantaneous phoneme (BMF) is not subject to any stabilizing force, and can therefore vary widely from frame to frame. Since the inferiority of the latter parameter was confirmed in further experiments, we abandon it at this point.

Best path per frame (BPF)	FER = 37.70 %
Best model per frame (BMF)	FER = 45.76 %

Table 1: Frame error rates (FER) for the best path per frame (BPF), and best model per frame (BMF).

3.2. Phoneme-Based Experiments

In order to obtain the actual phonetic transcription, the frame-based chains are smoothed and then collapsed. Collapsing consists of simply reducing successions of identical tokens to one token. For smoothing, a sliding weighted window was used: To decide on the identity of a phoneme, all symbols within an interval given by the window width are considered. The weighted frequency of every symbol inside the window is computed. The weight is determined by an exponential function of the distance of the symbol in question to the window center (symbols further apart contribute less). The symbol with the highest weighted frequency replaces the phoneme in the center of the window.

After the phoneme chains were smoothed and collapsed, we measured the Phoneme Error Rate (PER), which is calculated as usual:

$$\mathrm{PER} = \frac{\mathrm{Count(Ins)} + \mathrm{Count(Sub)} + \mathrm{Count(Del)}}{\mathrm{Count(Ref)}} \qquad (2)$$

where Count(...) represents the number of insertions (Ins), substitutions (Sub), and deletions (Del) among the hypothesized phonemes, when aligned to the reference phonemes (Ref) in the test set[3]. A phoneme error rate slightly over 33% was obtained.

Next, we experimented with BPF chains containing a variable number of alternative phonemes per frame n, with $n \geq 1$. Using a frame-based phoneme language model, the most likely phoneme sequence ($n = 1$) can be extracted similarly to the way the best word sentence is extracted from a word graph [4]. This sequence is different to the one belonging to the best path in every frame (BPF), because the context is also considered for the decision on the identity of the current phoneme.

This chain was then smoothed and collapsed, and finally aligned to the spoken phoneme references. Figure 3 shows the results. A very significant improvement of the phoneme accuracy was obtained: The phoneme error rate was reduced from ≈33% to ≈29%. It is interesting to note that as the number of alternatives presented to the language model grows, the importance of smoothing decreases: In this case, the language model does the smoothing.

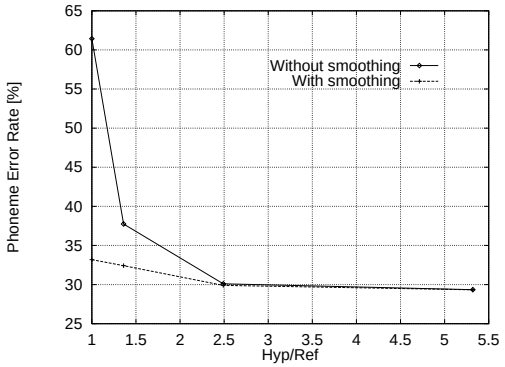

Figure 3: Phoneme error rates of the most likely phoneme sequence. This phoneme sequence is extracted from a graph structure containing a number of alternatives per spoken frame (*Hyp/Ref*).

3.3. Baseline System

For comparison purposes, the Daimler-Benz speech recognizer was also run in phonetic recognition mode. In this mode, phonemes are treated as if they were words. These "words" can be either context-independent (CI) or context-dependent (CD) HMM recognition units. In the former case, the lexicon consists of the 36 phonetic symbols used throughout this chapter, plus two models for silence and noise. In the latter case, we used approx. 1500 different phonemes in context. In addition, we also performed experiments with phoneme bigram backoff language models, which were used for constraining the search. The context-independent bigram language model was trained with the half of the Kiel Corpus of Spontaneous Speech not used for testing[4] . For the context-dependent bigram language model the whole Verbmobil corpus (excluding development and test sets) was used. Table 2 summarizes the characteristics of the language models. It is interesting to note that the language model of context dependent units (CD) is much more robust, even though its vocabulary is much larger. This is due to the strong contextual constrains imposed by context dependent units.

The results obtained for different system configurations are shown in the upper part of Table 3. As expected, the performance of the recognizer with context-dependent units (CD) is significantly better than with context-independent units (CI). Further, Table 3 clearly shows the importance of using a phoneme bigram language model. The language model

[3] In order to calculate *frame* error rates, no alignment is necessary because the strings to be compared have the same length. For *phoneme* error rates, the alignment with the smallest Levenstein distance is used.

[4] Tests with other training sets were performed as well. The bigram trained with the actually spoken phoneme strings from the Kiel Corpus of Spontaneous Speech yielded the lowest test set perplexity and was therefore chosen for recognition experiments.

LM	No. of turns	No. of tokens	Vocab. size	PP
CI	184	20 K	38	13
CD	9.6 K	1 M	1.5 K	4

Table 2: Characteristics of the language models used for the baseline phoneme recognizers.

is specially advantageous for large vocabularies, as in the context-dependent case. The best performance obtained is ≈28% phoneme error rate using context-dependent units and a bigram language model. This outcome is comparable to the ones reported by other Verbmobil partners [7].

System	+RTF	PER [%]
Baseline: CI, no LM	≈ 0.4	48.03
Baseline: CI, bigram	≈ 0.5	42.17
Baseline: CD, bigram	≈ 6	27.79
BPF	≈ 0	33.19
BPF, frame trigram	≈ 0.2	29.33

Table 3: Overhead real time factor (+RTF) and phoneme error rate (PER) for different transcription methods. The baseline systems are phoneme recognizers with different configurations. The cheap transcription algorithm is evaluated with and without a language model.

3.4. Comparison of Performance

Finally, we performed a comparative analysis of the computational resources needed by the different transcription methods presented above. For all methods tested, we measured the *elapsed time* needed to complete a recognition run. Then, we normalized them by the duration of the test set to obtain the *real time factor (RTF)*. We report here the overhead RTF (+RTF), which represents the additional time required to generate a phonetic transcription. All experiments were conducted on a Digital AlphaServer 8200 5/300 (SPECfp95 11.7, as reported in `http://open.specbench.org`). There, our word recognizer runs in 1.3 times real time (RTF ≈ 1.3).

Table 3 summarizes our measurements. The cheap transcription algorithm allows the input utterance to be transcribed with no (BPF) or very small computational overhead (BMF with frame trigram). In both cases, the phonetic transcriptions are more accurate and less expensive than those obtained using context-independent models with a phoneme recognizer. The lowest error rates are still obtained with a phoneme recognizer using context-dependent units and a phoneme bigram language model. Yet this performance gain is very costly (note that +RTF ≈ 6) due to the high confusability of the vocabulary, which makes pruning techniques ineffective. For comparison, we also tuned the recognizer parameters to allow more pruning. We could accelerate the context-dependent baseline recognizer from +RTF ≈ 6 to +RTF ≈ 3 with a minor performance degradation (PER = 30.97%). Still, the transcriptions provided by the cheap transcription algorithm are a better compromise between performance and computational costs.

4. CONCLUSIONS

We presented a method we dubbed the "cheap transcription algorithm", which can be used to transcribe utterances during recognition, so that the phonetic representation of the input is available as a by-product of the (word-based) search. Using this method, the computational cost of the word-based search remains almost unaltered. As a bonus, the method does not require any changes in the system lexicon, since the transcription is extracted directly from the models competing in the search. Further, the phonetic transcriptions obtained using the cheap transcription algorithm are almost as accurate as the ones obtained by a phoneme recognizer at a much higher cost.

ACKNOWLEDGMENTS

The authors gratefully thank Gunnar Evermann and Henrik Heine (University of Hamburg) for sharing their ideas and contributing to this work with interesting discussions. We also thank Matthias Pätzhold and Adrian Simpson (IPdS, University of Kiel) for their support with the Kiel Corpus of Spontaneous Speech.

REFERENCES

1. Ayman Asadi, Richard Schwartz, and John Makhoul. Automatic modeling for adding new words to a large-vocabulary continuous speech recognition system. In *Proc. ICASSP'91*, pages 305–308, 1991.
2. Katunobu Itou, Satoru Hayamizu, and Hozumi Tanaka. Detection of unknown words and automatic estimation of their transcriptions in continuous speech recognition. In *Proc. ICSLP'92*, Banff, Alberta, Canada, October 1992.
3. Klaus J. Kohler. Labelled Data Bank of Spoken Standard German: The Kiel Corpus of Read/Spontaneous Speech. In *Proc. ICSLP'96*, Philadelphia, USA, 1996.
4. Thomas Kuhn, Pablo Fetter, Alfred Kaltenmeier, and Peter Regel-Brietzmann. DP-Based Wordgraph Pruning. In *Proc. ICASSP'96*, volume 2, page 861, Atlanta, USA, 1996.
5. Jörg Reinecke. Evaluierung der signalnahen Spracherkennung im Verbundprojekt Verbmobil (Herbst 1996). Verbmobil Memo 113, Technische Universität Braunschweig, Germany, 1996. available at `http://sbvsrv.ifn.ing.tu-bs.de/eval.html`, in German.
6. Bernd Suhm and Monika Woszcyna. Detection and transcription of new words. In *Proc. EUROSPEECH'93*, pages 2179–2182, 1993.
7. Henrik Heine Uwe Jost and Gunnar Evermann. What is wrong with the lexicon—an attempt to model pronunciations probabilistically. In *Proc. EUROSPEECH'97*, Rhodos, Greece, 1997.
8. Sheryl R. Young. Detecting Misrecognitions and Out-Of-Vocabulary Words. In *Proc. ICASSP'94*, pages II–21–II–24, Adelaide, Australia, 1994.