

Improving Phone Label Alignment Accuracy by Utilizing Voicing Information

Dyre Meen, Torbjørn Svendsen

Jon Emil Natvig

Department of Electronics and Telecommunications
Norwegian University of Science and Technology
N-7491 Trondheim
Norway
{dyre,torbjorn}@iet.ntnu.no

Telenor Research and Development
Snarøyveien 30
N-1330 Fornebu
Norway
jon-emil.natvig@telenor.com

Abstract

The perceived quality of a concatenative speech synthesizer depends on an accurate alignment of unit labels and the corresponding waveform. This paper proposes to improve an HMM-based phone label alignment system by utilizing voicing information.

Two different approaches have been investigated: (1) incorporating voicing information into the feature vector and (2) acoustic post-processing based on changes in the estimated voicing state. Experiments on a Norwegian speech corpus show that both methods may improve alignment accuracy and that combined use may be beneficial. Informal listening tests indicate that the methods can yield improved quality for unit selection synthesis.

1. Introduction

Identifying the temporal positions of the phonemic events in a spoken utterance is desirable in many speech processing applications that utilize speech databases. Annotated speech databases are used for training of speech recognizers, evaluation of phone recognition performance of basic speech recognition algorithms, concatenative speech synthesis and speech science. If both the phone identity and their positions are unknown, the task is essentially phone recognition. When the orthographic transcription is known, but the actual pronunciation and phone durations unknown, the task is phonemic transcription and segmentation. If both the orthographic and the phone transcription are assumed known we are addressing the problem of aligning the phone sequence with the speech waveform, phone label alignment. This paper concerns phone label alignment in the context of unit selection speech synthesis.

In unit selection speech synthesis, a (large) speech waveform database is searched in order to extract a sequence of units that minimizes some objective cost function. Regardless of the unit type, unit selection synthesis requires that the speech database contains accurate information about the position of the units contained in the database.

Manual labeling is laborious, costly and prone to inconsistencies, and it is desirable to automate the process of labeling either fully or to the extent that human effort is only required to verify the results of the automated process and correct a (presumably) small number of alignment errors. It has been demonstrated that automatic segmentation can be introduced without degrading the resulting synthesis quality [1].

Most current methods for automatic segmentation are based upon hidden Markov models (HMM) and forced Viterbi decoding of the hypothesized unit string (see e.g. [2], [3], [4]). Some

systems employ post-processing techniques to refine the phone boundaries found by the HMM decoding, relying on e.g. systematic bias removal[5], evaluation of spectral change in a window centered around the hypothesized boundaries[3] or applying statistical boundary models to the a small region surrounding the HMM boundaries[4].

It should be noted that the problem of segmentation for text-to-speech (TTS) purposes differs from automatic speech recognition in several aspects. Firstly, the main purpose is not to identify the phone identities but to determine the positions of the boundaries between phones. Secondly, the recorded speech database can be used in its entirety for training or adapting the acoustic models based on its real or hypothesized phonemic content. This implies that accepted truths about the best choices for model topology, type of feature extraction, frame shifts, etc. need not be the same for the segmentation task as it is for automatic speech recognition (ASR). Dines et.al.[5] have shown that increased performance can be achieved by increasing the number of states and decreasing the number of mixture components relative to models optimized for ASR.

In section 2, we give a short overview of HMM based label alignment and briefly discuss issues and methods that can be applied in order to improve performance. In section 3, we present experiments, followed by a comparison of the results in section 4 and some words about perceptual evaluation in section 5.

2. HMM-based Label Alignment

A simplified diagram of an HMM-based system for phonemic segmentation is shown in figure 1. Acoustic models are trained using all available data. This may include any suitable available databases in addition to the material that is to be processed. The additional training material may be hand labelled. For the material to be processed, however, only orthographic transcriptions are available. For phonemic segmentation, the system should be capable of selecting between pronunciation variants. The purpose of the current work is to improve the alignment accuracy. Thus, we assume perfect prediction of phone strings, leaving out errors due to insertions, deletions and substitutions.

The segmentation process is essentially the same as the decoding process in ASR with the important distinction that the unit sequence is (assumed) known so the aim of the decoding is to determine the time instances of transitions between units. Finally, a post-processing step may be applied in order to fine-tune the boundaries.

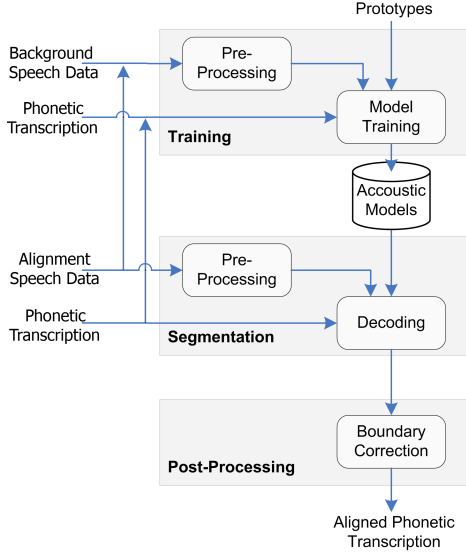


Figure 1: HMM-based alignment system

2.1. Utilizing Voicing Information

Voicing is a strong cue to detecting transitions between voiced and unvoiced segments. Exploiting explicit voicing information should potentially improve the detection of phone boundaries where the voicing state changes. We have considered two strategies:

1. Extending the feature vector with voicing information
2. Using voicing in a cost function in a post-processing unit

Both approaches depend on an available voicing graph. This may be accomplished by processing a laryngograph signal or by using existing algorithms for extracting voicing information.

In our work we regard that a phone belongs to one of three classes; voiced (V), unvoiced (U) or other (N) (see Table 1). For segmentation purposes, we wish to include information on the state of voicing at either side of a phone boundary. We define the voicing state context of two consecutive phones, p_n and p_{n+1} , as

$$C(p_n, p_{n+1}) \in \{VV, VU, VN, UV, \dots\} \quad (1)$$

2.1.1. Extending the Feature Vector

The observation probability $b_j(\mathbf{o}_t)$ for state j at time instance t may be regarded as the joint probability of S independent data streams given by [6]

$$b_j(\mathbf{o}_t) = \prod_{s=1}^S \left[\sum_{m=1}^{M_s} c_{j sm} \mathcal{N}(\mathbf{o}_{st}; \boldsymbol{\mu}_{j sm}, \boldsymbol{\Sigma}_{j sm}) \right]^{\gamma_s^j} \quad (2)$$

where M_s is the number of mixture components in stream s , $c_{j sm}$ is the weight of the m 'th component and $\mathcal{N}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is a multivariate Gaussian with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The exponent γ_s^j is a state dependent stream weight which may be used to emphasize particular streams.

Using this representation, the voicing information may be regarded as an independent stream, s_2 , in parallel with the original feature vector, s_1 . For segmentation purposes, the voicing information will provide most additional information at phone

boundaries where there is a change in voicing state. Because of this, it may be beneficial to emphasize s_2 more in VU and UV transitions and less in VV and UU transitions.

To accomplish this, it is necessary to make the phone models dependent on voicing context, which in turn increases the required number of models in the inventory. In case of scarce training material, this may be countered for by using tied model parameters and allowing only the stream weights to differ.

2.1.2. Using Voicing Information in Post-Processing

A post-processing stage may be constructed by utilizing an appropriate cost function which is sensitive to phone boundaries and move each boundary to the most prominent peak within a predefined window around the boundary proposed by the HMM decoding.

The approximated time derivative of the voicing information may serve as an adequate cost function for detecting VU and UV boundaries. Final boundary positions can be found by using a dynamic programming (DP) search for the most prominent peaks subject to the condition that there can be only one boundary within each window and that overlapping windows cannot share boundaries. If no prominent peaks are found, the HMM boundary is kept unchanged.

At this stage the data rate of the voicing function may be independent of the HMM feature analysis, thereby making it possible to mitigate problems due to the limited time resolution of the HMM alignment process.

3. Experiments

The experiments were performed on the Norwegian speech database PROSDATA [7]. PROSDATA was originally intended for studies of Norwegian prosody and is a collection of 502 news sentences read by a female speaker. The studio recording was digitized at a sampling rate of 16kHz. The database has been manually segmented in terms of phonemes, syllables and words. The phonemic segmentation was undertaken by a single expert labeler. The total number of phones in the corpus is 39710. The distribution of phones in the PROSDATA database is uneven, some of the 50 phones have as few as 2-15 occurrences. This makes reliable estimation of model parameters difficult for some of the infrequent phones. Table 1 summarizes the phone-sets V, U and N for PROSDATA.

Symbol	Phones (SAMPA)
V	b d g v m n N l i i: e e: { { : A A: y y: 2 2: O O: u u: } } : { i 2y A } Ai Oy } i ui rn rl @ @rn
U	p k t f s S C h r t
N	r rd rL

Table 1: Phone sets; Voiced(V), Unvoiced(U) and Other(N)

3.1. Evaluation

There are no “true” boundary positions in the continuous sequence of phone realizations. Thus, evaluating the performance of an automatic system should ideally have been done by evaluation of the resulting synthesis quality in a unit selection TTS. In this study, however, we have chosen to evaluate the system performance by comparing with the manual segmentation. Moreover, since the “true” (hand made) phone string is fed into the system, as depicted in figure 1, the evaluation will give an upper

bound for the alignment performance. In a practical system the phone string has to be estimated, thereby introducing errors due to insertions, deletions and substitutions.

3.2. The Baseline System

Our baseline system is based on the Hidden Markov Model toolkit (HTK) [6].

Every 5 msec the pre-processor extracts a 39 dimensional feature vector from a 15 msec Hanning windowed version of the speech signal. The vector consists of 13 MFCC coefficients (including c_0) appended with delta and acceleration coefficients.

The phone model set consists of 52 context-independent left-right HMMs. Based on initial experiments a model topology with 7 states and 2 Gaussian mixtures was found to be best suited for our task. This confirms the findings of Dines et al[5]. A supervised flat start embedded training paradigm was used to compute the final model inventory.

3.3. Extending the Feature Vector

The scalar voicing stream, s_2 , was constructed using the measure:

$$v(t) = P_{voicing}(t) + w \cdot k_1(t) \quad (3)$$

where $P_{voicing}(t)$ and $k_1(t)$ is an estimate of the probability of voicing and the first reflection coefficient at time t , respectively, estimated by the ESPS program *formant*. w is a weighting constant. The reflection coefficient is related to voicing through correlation.

It is possible to construct models that favors one or another stream at a particular state. In order to weight s_2 according to voicing state context, $C(p_n, p_{n+1})$, each voiced model, $p \in V$, were cloned to three versions: $p_{C(V,V)}$, $p_{C(V,U)}$ and $p_{C(V,N)}$. Problems due to the increased number of models were mitigated by tying all parameters of the cloned versions, except for the stream weights. Likewise, each unvoiced model, $p \in U$, were cloned to $p_{C(U,V)}$, $p_{C(U,U)}$ and $p_{C(U,N)}$. The remaining models $p \in N$, were kept unchanged. The input phone level transcription was also modified in order to reflect the extended set of models.

Then, the voicing stream weight for the j 'th state of the n 'th phone model, p_n , in an utterance was set according to

$$\gamma_2^j(p_n, p_{n+1}) = \begin{cases} const & , 1 \leq j < J_n \\ f(C(p_n, p_{n+1})) & , j = J_n \end{cases} \quad (4)$$

where J_n is the number of states in the model and $f(.,.)$ is a function taking the voicing context into account, giving relatively more weight to UV and VU transitions than other transition types.

Initial training was done with equal weights for all states in the voicing stream. Then the extended phone model set with context dependent stream weights were built followed by several iterations of Baum Welch (BW) re-estimation. During re-estimation the weights were kept constant.

3.4. Post-Processing

A post-processor utilizing voicing was implemented and tested on both the baseline and the feature-extended system. This was done by defining a cost-function, $d(t)$, given by

$$d(t) = |P_{voicing}(t) - P_{voicing}(t-1)| \quad (5)$$

This measure tends to give sharp sharp peaks whenever the voicing state changes. This cost function was used as described in section 2.1.2 in order to adjust the UV and VU boundaries.

4. Results

The comparisons have been done in terms of the percentage of the phone boundaries that are within a specified maximum distance from the manual boundaries, the coincidence rate, c_s , where s is the maximum allowed deviation in ms. Also, the mean, μ_{err} , and standard deviation σ_{err} of the time difference between manual and automatically found boundaries are computed. In addition, the number of gross errors, N_{gr} , is counted. A gross error occurs when there is no overlap between the manual and automatic phone placements.

Table 2 illustrates results for all boundaries. The context dependent weighting experiment (Ctx) outperforms the baseline (Bsl) for all evaluation parameters. However, the strategy depends on estimated weights, thereby increasing the total number of system parameters. The post-processing unit, on the other hand, does not depend on these weights. Processing of the baseline (Bsl+Post) results in an increase from 67.5% to 74.2% of boundaries within 10 ms from the reference labelling.

Experiment	N_{gr}	μ_{err}	σ_{err}	c_{10}	c_{20}
Bsl	67	-2.12	12.5	67.5	90.4
Ctx	60	-0.56	11.8	73.9	92.5
Bsl+Post	69	-0.60	12.0	74.2	91.9
Ctx+Post	63	-0.01	11.7	75.2	92.4

Table 2: Segmentation performance for all (38968) boundaries illustrating the number of gross errors, mean and standard deviation of the alignment error and coincidence rates for 10 and 20 ms deviation.

The results for VU boundaries are summarized in table 3. As can be seen, the baseline suffers from a larger mean deviation error, which decreases the coincidence rates for the 10 and 20ms error bounds.

By extending the feature vector with a voicing stream and using context dependent weighting, both the mean and standard deviation of the alignment error drops. For the 10ms error bound, the coincidence rate increases from 46.5% to 80.5%.

Also, the post-processor seems to effectively decrease the mean alignment error, both when utilized on the baseline and the context-dependent stream weighting experiment.

Experiment	N_{gr}	μ_{err}	σ_{err}	c_{10}	c_{20}
Bsl	3	-9.91	9.03	46.5	87.1
Ctx	3	-2.85	7.77	80.5	97.1
Bsl+Post	3	-3.30	7.93	81.6	95.9
Ctx+Post	3	-1.19	7.14	87.4	97.9

Table 3: Segmentation performance for (7254) VU boundaries.

Table 4 illustrates the results for UV boundaries. As can be seen, the baseline produces almost no mean error and the coincidence rates are consequently far better than for VU boundaries, thereby diminishing the effect of using the two extended techniques.

5. Informal Listening Experiments

Our aim has been to improve the performance of automatic phone alignment for unit selection speech synthesis. The seg-

Experiment	N_{gr}	μ_{err}	σ_{err}	c_{10}	c_{20}
Bsl	8	0.13	9.31	83.77	95.60
Ctx	5	0.27	8.68	84.52	96.30
Bsl+Post	9	1.71	9.95	84.18	94.75
Ctx+Post	6	1.73	9.14	84.92	95.32

Table 4: Segmentation performance for (6949) UV boundaries

mentation accuracy with respect to manual segmentation is an indicator of improved performance, but the only true assessment of the quality is obtained by evaluating the synthesis quality obtainable. One may argue that optimal coupling techniques may counter inaccurate segmentation. This is to some extent true for alleviating join discontinuities, but it is also possible that segmentation errors where e.g. a part of the adjoining phone is being included in a phone may lead to noticeable errors due to erroneous duration estimation.

As a first step to verify the usability of the results we have built some TTS-voices based on the different segmentation experiments. The voices were built by means of Festvox[8] and the Festival TTS “clunits” framework[9]. The voices all had a fixed setup - except for the phone boundary positions.

A separate front-end developed at Telenor R&D[10] was used.

Informal listening experiments on the different voices do indicate that the quality differs and that the proposed methods are beneficial. In order to make conclusions, however, more testing is needed.

6. Conclusions

In this paper we have proposed using voicing as a mean for improving HMM-based phone label alignment. By extending the feature vector with a voicing stream and apply context-dependent weighting, the coincidence rate for 10 ms error bound increased from 67.5% to 73.9% when compared to the baseline. For VU boundaries the rate increased from 46.5% to 80.5%. A performance increase was also achieved by utilizing voicing in a post-processing unit. When using the baseline as initial estimates, we achieved a 74.2% coincidence rate for 10ms error bound. The VU boundary class achieved 81.6%.

The findings in this paper are currently being tested on segmentation of two new Norwegian speech corpora. Preliminary experiments, using voicing information parameters from PROSDATA, indicate a gain similar to the presented results.

7. Acknowledgements

This work has been supported by the FONEMA project (Methods for realistic Norwegian speech synthesis). The project is funded by the Research Council of Norway and is a collaboration between the Norwegian University of Science and Technology (NTNU) and Telenor R&D.

8. References

- [1] M.K. Makashay, C. Wightman, A.K. Syrdal, and A. Conkie, “Perceptual evaluation of automatic segmentation in text-to-speech synthesis,” *Proc. ICSLP-2000*, pp. 431–434, 2000.
- [2] T. Svendsen and K. Kvale, “Automatic alignment of phonemic labels with continuous speech,” *Proc. ICSLP-1990*, pp. 997–1000, 1990.
- [3] Y.-J. Kim and A. Concie, “Automatic segmentation combining an [hmm]-based approach and spectral boundary correction,” *Proc. ICSLP-2002*, pp. 145–148, 2002.
- [4] A. Sethy and S. Narayanan, “Refined speech segmentation for concatenative speech synthesis,” *Proc. ICSLP-2002*, pp. 149–152, 2002.
- [5] J. Dines, S. Sridharan, and M. Moody, “Automatic speech segmentation with hmm,” *Proc. 9th Australian Conference on Speech Science and Technology*, 2002.
- [6] Steve Young et.al, *The HTK Book (for HTK Version 3.2)*, 2002.
- [7] J. E. Natvig and P.O. Heggveit, “Prosdatabasen 2.0. a speech database for study of norwegian prosody [r&d] n 20/2000,” 2000.
- [8] A. W. Black and K. A. Lenzo, *Building Synthetic Voices*, 2003.
- [9] A. W. Black, P. Taylor, and R. Caley, *The Festival Speech Synthesis System*, 2002.
- [10] “Telenor talsmann®,” <http://www.telenor.no/fou/prosjekter/taletek/talsmann/>.