Improving automatic phonetic segmentation for creating singing voice synthesizer corpora

Varun Jewalikar

MASTER THESIS UPF / 2009

Master in Sound and Music Computing

Master thesis supervisor :
Jordi Bonada

Co-supervisor :
Merlijn Blaauw

Department of Information and Communication Technologies
Universitat Pompeu Fabra, Barcelona

Abstract

Phonetic segmentation is the breakup and classification of the sound signal into a string of phones. This is a fundamental step for using a corpora for a singing voice synthesizer. We propose improvements to an existing automatic phonetic segmentation method by adding more relevant descriptors to the computed feature set and by using a different regression model.

We start with a short introduction to singing voice synthesizers and how their corpora are created. We discuss the importance of automatic phonetic segmentation for these corpora. We briefly review and critique works relevant to phonetic segmentation of both speech and singing voice. This is followed by an introduction to score predictive modelling and how it will benefit with some fundamental modifications.

A detailed description of how score predictive modelling is adapted for our corpora and how it is implemented is presented. The corpora contains sentences sung by a professional female singer in Spanish and also contains accurate manual phonetic segmentation information. This corpora is divided into a train set and a test set (in a 3 to 1 ratio respectively). Relevant audio features are extracted and these serve as the backbone for training and testing of the machine learning models. A score function is calculated for candidate boundaries in the train set. The score and features for the train set are used for training random forest regression models. These trained models (called score predictive models) are used for predicting improved phoneme boundaries, around boundaries predicted by Hidden Markov Models (HMMs) for the test set. These predicted boundaries are then evaluated against the manually labelled boundaries (true boundaries) and boundaries previously found using HMMs (baseline).

The results obtained are promising and justify our modifications of using a large feature set and a different regression model. A number of interesting possibilities for future works are presented. We conclude with a summary of the work, conclusions and contributions.

Acknowledgements

I would like to thank Xavier Serra and the staff at the MTG for giving me an opportunity to be a part of this masters program. Above all, I would like to thank my supervisors Jordi Bonada and Merlijn Blaauw for their invaluable technical contributions, advise and for being very patient with me.

I would also like to express my sincere gratitude towards all the teachers at the UPF for making me more knowledgeable and fuelling my curiosity about sound and music computing.

I want to thank all my classmates and friends for the endless hours spent discussing, helping each other and partying. Without them this journey wouldn't have been so fulfilling.

My biggest thanks go to my family and Cristina for supporting me unconditionally.

Contents

List of Figures

1 Introduction

Singing voice synthesis strongly relies on corpus-based methodologies and, therefore, on the availability of good singing voice corpora. In order for a corpus to be really useful, apart from the audio itself, it should contain information about its contents (labels) and about the time alignment between labels and the audio. This thesis focuses on the problem of improving automatic phonetic segmentation for a singing voice synthesizer corpora, that is, the problem of automatically locating the boundaries between the sounds corresponding to the phones that make up a fragment of audio. The phone sequence is considered as given information.

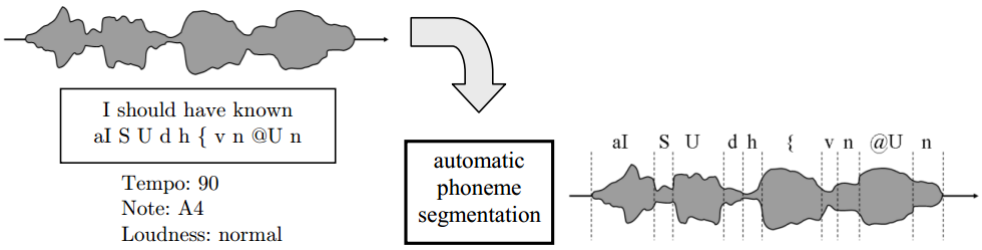

Figure 1: Illustration of phonetic segmentation

Phones represent the acoustic realisations of the smallest meaningful units of speech (phonemes), by the concatenation of which any other speech unit (syllable, word, phrase, etc.) can be built. The singing voice synthesis method that this thesis is concerned with uses diphone concatenation (Figure 2). A diphone is an adjacent pair of phones and using recorded diphones for speech/singing voice synthesis sounds more natural (than using recorded phones) because the pronunciation of each phone varies depending on the surrounding phones.

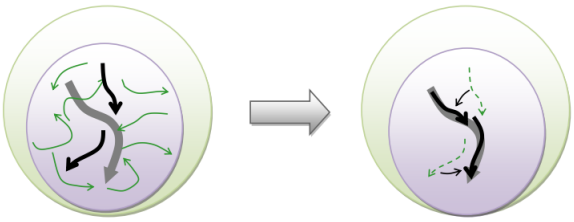

Figure 2: Diphone concatenation

On the left side we see the target trajectory (wide arrow) and the available diphone samples (narrow arrows). The two selected samples are drawn in black with wider width. On the right, we see how these samples are transformed and concatenated to approximate the target trajectory.

Phonetic segmentation is an important primary step for the automated generation of a diphone inventory since it decides the central boundaries of diphones and also the minima/maxima of their edge boundaries. The most precise way to obtain this information is manually [7]. But the slow pace of manual labelling often creates a bottleneck. Even well trained and experienced phonetic labellers working with a familiar voice using efficient speech display and editing tools on a modern workstation require about 200 times real time to segment and align speech utterances. Also, if several transcribers are used for manual labelling, there is the problem of inconsistency [4]. The need for searching for alternatives for manual labelling is evident.

In automatic speech recognition the use of Hidden Markov Models (HMMs) has avoided the need for manual phonetic segmentation. HMMs produce a segmentation which, although less precise than a manual segmentation, seems to be precise enough to train the automatic speech recognition systems. This is because HMM training is an averaging process that tends to smooth segmentation errors. These automatic speech recognition systems can be adopted for automatic phonetic segmentation by restricting their language model to the known input sentences. But the phonetic segmentation produced with this technique is not precise enough for speech or singing voice corpora. This is because unless automatic speech recognition systems are trained on segmented (not only labelled) speech and unless proximity to the boundaries is part of the optimality criterion, these systems may put boundaries at quite different locations such as, for vowel-voiced fricative boundaries, the onset of frication instead of the formant structure [5]. Also, automatic speech recognition based systems require large amounts of language-specific training data which is generally not available for singing voice synthesis systems. Thus, there is a need for finding other automatic phonetic segmentation techniques spe-

cific for singing voice corpora.

Generally, methods for phonetic segmentation for speech corpora involve two steps [8]. First, we perform a rough phonetic segmentation by forced alignment of the Viterbi search using HMMs with Mel-scale frequency cepstral coefficients (MFCCs). Then, we apply a boundary refinement procedure as a post-processor to fine-tune the results obtained by the HMM. This two step approach imitates human labellers who first try to find coarse phoneme segments and then zone-in on the exact phonetic boundaries. Intuitively, it is possible to perform segmentation of singing voices by the same scheme.

Lately, this boundary refinement approach has been adapted specifically for singing voice synthesizer corpora in the form of score predictive modelling [21]. This approach has been tested on a Chinese singing voice corpora and relies heavily on the tonal aspects of this language. We propose to adapt the fundamental idea from this approach for a singing voice synthesis corpora which is different from one used in the original work and suggest modification of a few key aspects of this approach.

1.1 Goals

The major goals of this thesis are:

- To write a concise review of the state of the art in the field of automatic phonetic segmentation for creating singing voice synthesizer corpora.
- To adapt an existing state of the art technique (score predictive modelling) for a different type of corpora.
- To change two fundamental aspects of score predictive modelling i.e. to increase the size of the feature set and to use a different regression modelling approach.
- To evaluate the results of these changes on a Spanish female singing voice synthesizer corpora.
- To implement the framework in an optimised manner with open source toolkits and distribute it freely.
- To discuss the results and suggest possible avenues for future work.

2 State of the Art

We first begin with a brief overview of the research in automatic phonetic segmentation of speech corpora and then a review of automatic phonetic segmentation specifically for singing voice synthesizer corpora.

2.1 Speech segmentation

As mentioned before, the most common approach used is to modify an existing HMM based phonetic recogniser for the task of segmentation [9]. The language model of the recogniser is restricted to the known input sentences and this is termed as forced alignment. The best result (90% of boundaries within 20 ms of human labelled ones) obtained with this approach is reported in [10]. Several modifications for this have been developed. In [12] a pre-segmentation technique is used followed by HMMs and cepstral coefficients to align the spectrally stable segments to phones. [13] combines HMMs with heuristic rules. HMMs are combined with speech synthesis and neural networks in [6].

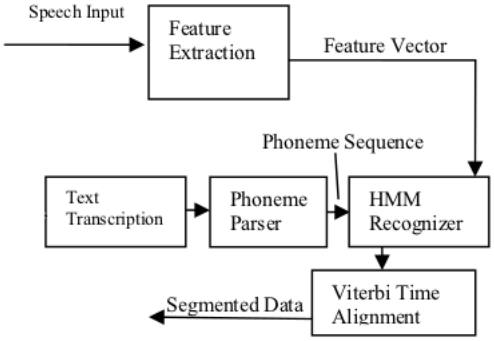

Figure 3: HMM based phonetic segmentation

[14] suggests moving away from the 'data driven' solutions to this problem and integrating expert knowledge. They develop a representation where each segment of the speech signal is a transition between two targets, which renders many of the effects of co-articulation irrelevant. Multi-step Adaptive Flux Interpolation (MAFI) is used, which provides an algorithm for describing extended speech segments in terms of an initial parameter vector, a target and a duration. The terminal observation vectors could be MFCCs, power spectral densities or even auditory representations. It is demonstrated that an appropriate choice of parameters for MAFI leads to segment boundaries which are similar to manually labelled

phoneme boundaries. This is essentially a pre-segmentation into spectrally stable elements.

Similar to the above, in [15] a multi-level description of speech segments which contains both coarse and fine information in one uniform structure, called dendrogram is used. It is shown that dendrograms can capture more than 96% of acoustic phonetic events of interest with an insertion rate of less than 5% [19]. This is a multi level pre-segmentation strategy. Segmentation is carried out by searching for the best path through the segmentation graph and using information about the phonemic contents of speech (for eg: maximum/minimum length of phonetic segments). This approach has the advantage of being very fast, allows integrating additional heuristic information and requires no training. This could be followed by an alignment of segments and phones [12].

[16] simulates a human expert spectrogram reading process and performs assumption-based inference with certainty factors. This gives accuracy comparable to human labellers. But the inherent drawback is the availability and cost of this expert segmentation knowledge and the time taken to input these rules into the system.

Neural networks were adapted for phoneme event detection in (Figure 4) [17]. First a preprocessing of the speech signals with warped linear predictors is carried out. Warped linear prediction is a modification of the ordinary linear prediction (representing future values of a discrete-time signal as a linear function of previous samples) in order to implement the warped frequency scale (Bark scale) of human auditory perception. This representation is as compact and powerful as MFCCS with the added advantage that the normalized output can be directly used as input for neural networks. This is fed to a set of diphone event detectors composed of multilayer feed-forward neural nets (multilayer perceptrons). And finally a rule-based parser is used for matching the given transcription and the diphone event sequence from diphone detectors. They performed well in general (1-2% coarse labelling errors and on average 10ms deviation of boundary positions for the Finnish language) but have problems with some phonetic transitions like vowel-liquid and slow transition diphones inside dipthongs.

Another common approach is the alignment of the recordings to the same utterance produced by a speech synthesizer using DTW [7]. This technique is more robust to effects of co-articulation as compared to HMM based approaches but lacks speaker independence.

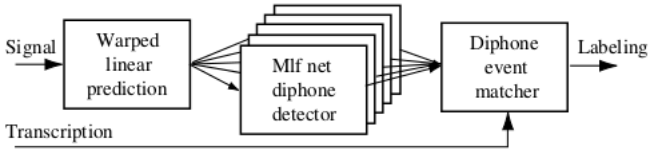

Figure 4: Neural networks based segmentation

Among the features used we can mention amplitude [12], short time energy contour [7], [15], energy in different frequency bands [12], spectral variation functions (SVFs) [7] and f0 contour [18].

2.2 Techniques adapted for singing voice synthesis corpora

Most approaches use two steps, an initial coarse estimation followed by refining the boundaries.

2.2.1 Initial estimate using HMMs

HMMs are stochastic state machines where the current state is not directly observable; an HMM emits an observable symbol per state. The probability of an HMM emitting a symbol is modelled by a mixture of Gaussian distributions, as described in the equation

$$b_j(x) = \sum_{m=1}^{M} C_{mj} N[x, u_{mj}, U_{mj}]$$

Where x could be the feature extracted from the audio e.g. MFCC, C_{mj}, u_{mj} and U_{mj} are the coefficient, mean vector and covariance for mixture component m in state j.

HMMs are typically created using an iterative training method called the Baum-Welch algorithm, which uses a set of training data to estimate the HMM model parameters. Starting with a prototype HMM, the Baum-Welch algorithm adjusts these parameters to maximise the likelihood of observing the data.

Generally, MFCCs with Cepstral Mean Normalization (CMN) and normalized log energy, as well as their first and second order differences are used as feature vectors. HMM topology is another important consideration. A common configuration would be 5 states, with transitions from left to right and no skips. Output probability distributions can be modelled with varying number of diagonal covariance Gaussians (1 to 6) [8].

It is common to use context independent HMMs for segmentation even though they are worse at modelling spectral movements in phonetic transitions as compared to context dependent HMMs. But they have the advantage of more precise segmentations over context dependent HMMs. Context-dependent HMMs are always trained with realizations of phones in the same context. For that reason, the HMMs do not have any information to discriminate between the phone and its context. As a result the HMM (particularly the lateral states) can end up modelling part of other phones or not all the phone. Context-independent HMMs, on the other hand, are trained with realizations of phones in different contexts. For that reason they should be able to discriminate between the phone to model (invariable in all the training examples) and its context (which varies) [20]. The main difficulty in phonetic segmentation for context-dependent HMMs when compared to context-independent HMMs is non-stationary phones. These phones clearly pose a greater challenge than that posed by stationary phones to keep the alignment between phones and context-dependent HMMs [8].

Each phoneme can be modelled by an individual HMM. The probability of the input feature vector matching the HMM is used to identify the words sung. A baseline system using HMMs for singing voice synthesis has been tested in [21] for Mandarin. Just by itself, it does not have very promising results (only 50% boundaries within 20ms of human labelled ones). This proves that there is a need to explore other techniques or combining it with complementary approaches.

2.2.2 Initial estimate using DTW

The singer recording the singing voice synthesis corpus is sometimes required to follow a melodic score. This could be used to perform phonetic segmentation by aligning the singer's pitch information with the corresponding melodic information using Dynamic Time Warping (DTW). DTW is an algorithm for measuring similarity between two sequences which may vary in time or speed. The sequences are "warped" non-linearly in the time dimension to determine a measure of their similarity independent of certain non-linear variations in the time dimension. For example: for a phrase which is sung, suppose that the input pitch (semitone) vector is represented by $t_{(i)} = 1, 2, ...M$ and the referenced pitch vector is represented by $r_{(j)} = 1, 2, ...N$. The vectors can be of different length. Then a M∗N DTW table is constructed using the following recurrence:

$$D_{(i,j)} = |t_{(i)} - r_{(j)}| + min\{D_{(i-2,j-1)}, D_{(i-1,j-1)}, D_{(i-1,j-2)}\}$$

The optimum end point can be found as the minimum element of the last col-

umn. The corresponding best alignment path can be found by backtracking from this point. This approach has been tested for a singing voice synthesis in [21] and independently, its accuracy is not high (70% of the boundaries within 20ms of human labeled ones).

For the corpora used for this thesis, the singer is not supposed to follow a melodic score but is supposed to sing one syllable per metronome click at a constant pitch. In case of automatic phonetic segmentation for speech corpora, a speech synthesizer is used to produce a synthetic reference signal from the phonetic transcription derived from the text (Figure 5). The speech signal is then temporally aligned on this reference, in which the phonetic segmentation is known, using DTW with feature vectors (like MFCCs) [6][11].

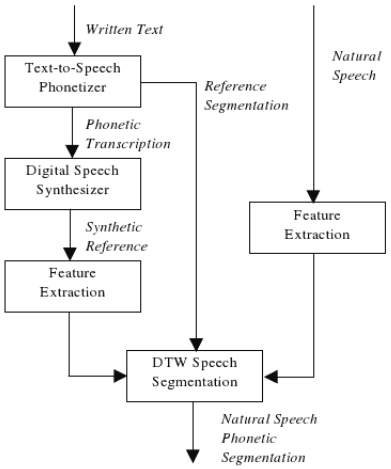

Figure 5: DTW based speech segmentation

Previously, DTW has been used extensively for melody recognition [1][2]. The experiments presented in [3] suggests that the accuracy of DTW alone is not very high but its combination with HMMs results in good performance. The reason for this is that if two neighbouring phonemes have same pitch then it can't be handled by DTW but in this situation HMMs have high performance given that the two phonemes have different pronunciation. On the other hand, co-articulation between two syllables usually poses a difficult problem for HMMs, but it is never a problem for DTW as long as they have different pitch. Hence the performance of the two techniques seems complementary and can be integrated for higher performance.

2.3 Boundary refinement

2.3.1 Hybrid approach

The central idea is to use a "Divide and Conquer" approach to choose different features and techniques (rule based or statistics based based) for refinement of boundaries for different phonetic transition categories. The phonetic categories according to one scheme [3] could be fricative, affricate, unaspirated stop, aspirated stop and periodic voiced. There is scope for trying out other ways of classifying phonemes.
The training data for each phonetic transition category is then split into two categories, "correct" and "wrong" according to the distance to the true manually labelled boundary. The classification based method adopts the k nearest neighbour strategy. A fixed search range is then used for boundary refinement. The "Periodic voiced to Periodic voiced" transition category has a bad performance using the above statistical method and hence this case is handled by heuristic rules [3] (thus using the "Divide and Conquer" idea).
Hard classification of boundaries (into "correct" or "wrong") seems unnatural and it would be more desirable to have a soft classification of the boundaries (a continuous number between 0 and 1). Also using a fixed search range across all categories is too assertive and a dynamic search range seems a plausible option.
This approach is proposed and tested in [21] and shows that there is still scope for improvement (80% boundaries within 20ms of human labeled boundaries). These drawbacks are addressed in the next subsection.

2.3.2 Score predictive modelling (SPM)

A score function for rating the candidate boundaries is selected. Then the candidate boundaries are scored using this function and the score along with the acoustic features of these boundaries are used as the training set. A regression (Support Vector Regression) approach is used to construct the SPM based on a supervised classification approach.

The scores of the initial boundaries from both DTW and HMMs are calculated and the boundary with the higher score is preserved. A dynamic search range is used to determine the suitable candidates around the preserved boundary and their scores are calculated using the SPM. The boundary with the highest score is selected as the final refined boundary. This approach has better performance (95% of boundaries within 20ms of human labelled ones) as compared to the hybrid approach stated above.

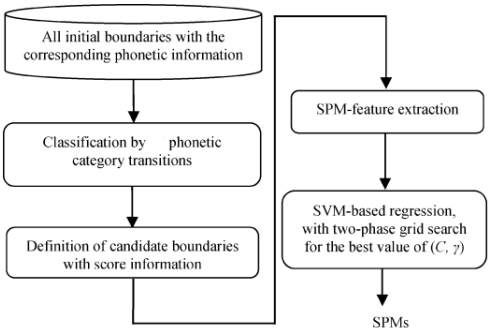

Figure 6: Construction of SPMs

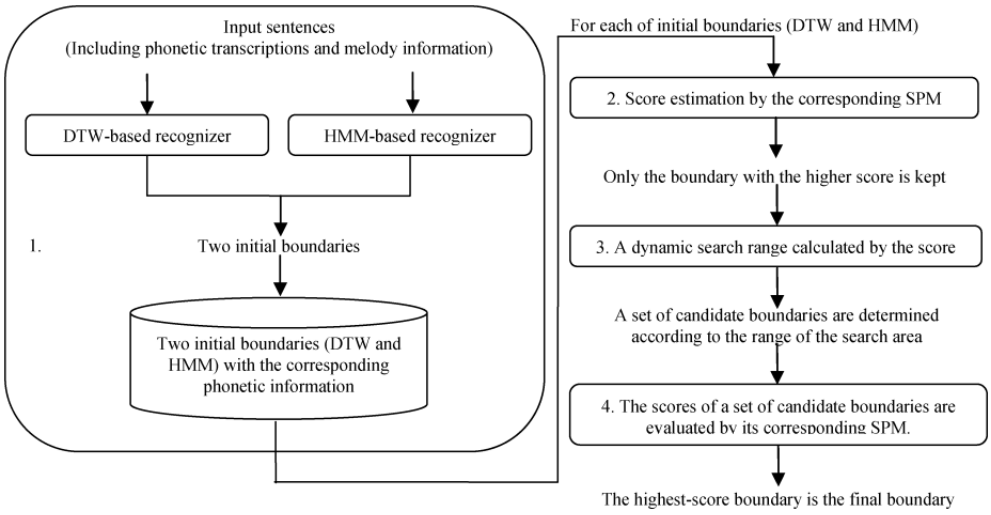

Figure 7: Boundary refinement with HMMs and DTW

3 Adapting a boundary refinement approach for a new singing voice synthesizer corpora

The approach which seems most suitable for our corpora is that of Score predictive modelling [21]. Firstly, this approach has been developed and tested for a singing voice synthesizer corpora. The fundamental idea (of score prediction for boundaries) is language independent. Hence, it can be easily adapted to our corpora which is in a different language as compared to the corpora the original work was based on. One caveat is that it depends on the tonal characteristics of Mandarin

but the fundamental idea of assigning scores to candidate boundaries doesn't rely on this. Moreover, it seems to have a high accuracy for boundary refinement.

While recording our corpora, the singer is supposed to sing the sentences at a constant pitch. This effectively renders the use of DTW for predicting initial boundary estimates useless. Instead, we choose to only use HMMs for initial boundary estimates and propose another approach for future work which is independent of HMMs altogether.

The fundamental idea is to use a machine learning approach to boundary refinement. There is a choice between using classification or regression models. Classification will output 'correct' or 'wrong' for each candidate boundary depending on the distance from the true boundary. It is more intuitive to have a fuzzy/soft classification i.e. a continuous number between 0 and 1 predicting the distance from the true boundary. This is called regression modelling. This gives more insight about the distance from the true boundary and can be a better guide for selecting amongst the candidate boundaries. Our method consists of five important parts: corpora preprocessing, score function calculation, feature extraction, training, testing and evaluation.

Figure 8: Block diagram of score predictive modelling

3.1 Corpora preprocessing

We have restricted the testing and evaluation to a female singing voice database. This database is in Spanish. It consists of 123 audio files (*.wav) each containing one sentence recorded by a trained female singer. Each sentence has a constant pitch and tempo. These sentences are generated so as to cover all the diphoneme combinations needed for singing voice synthesis in an optimum way [23]. Each audio file starts and ends with silence. Each sentence is recorded in 4 pitch ranges: low pitch, mid pitch, high and head pitch. This gives us a collection of 492 ($4*123$)

audio files.

Each audio file has a corresponding file (*.seg) [figure 9]. This contains timestamps of the start and end position of each phoneme of the sentence contained in the audio file. This is annotated by expert labellers. Whenever we refer to true boundary, it refers to this manually annotated end boundary of the phoneme. Subsequently, for each of these audio files we also have the phoneme boundaries obtained by using HMMs. These boundaries are used as initial estimates (baseline) for improvements during the testing phase.

```
nPhonemes 24
articulationsAreStationaries = 0
phoneme     BeginTime       EndTime
=================================
Sil     0.000000        2.449705
Sil     2.449705        2.589025
o       2.589025        3.036009
e       3.036009        3.268209
G       3.268209        3.320454
s       3.320454        3.407528
n       3.407528        3.488798
u       3.488798        3.633923
B       3.633923        3.680363
T       3.680363        3.819683
j       3.819683        3.877732
a       3.877732        4.080907
l       4.080907        4.138957
m       4.138957        4.231837
e       4.231837        4.481451
x       4.481451        4.614966
o       4.614966        4.876190
r       4.876190        4.969070
d       4.969070        5.015510
i       5.015510        5.241905
tS      5.241905        5.416054
o       5.416054        5.996553
Sil     5.996553        6.135873
Sil     6.135873        7.471020
```

Figure 9: Example of manual annotation for an audio file

The phoneme space is divided into unvoiced plosives (p,t,k), voiced plosives (b,d,g,B,D,G), unvoiced affricates (tS), unvoiced fricatives (f,T,s,x), nasals (m,n,J), liquids (l,r,rr,L), semi vowels (j,w,I,U), vowels (a,e,o,u,i) and Silence (Sil). Each phoneme boundary is categorised as a transition from one of the above categories to another. For instance, a transition from 'o' to 'e' is categorised as a Vowel to Vowel transition. We want to have a score prediction model for transition from each of the above category to every other. Obviously, not all of these transitions are possible because of the semantics of language.

At the end of this stage we have the audio data and the time stamps of all the boundaries for each of the possible transition categories.

3.2 Score function calculation

For each of the candidate boundaries a score function is calculated. As the distance between candidate and true boundary increases the score decreases. The score function is a normal distribution where x ($0 < x < 250ms$)is the distance of the candidate from the true boundary in milliseconds, μ is the mean and σ is the standard deviation.

$$Score(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu/\sigma)^2}{2}}$$

A sufficiently wide curve (250 ms wide) is obtained by using $\sigma = 40$. The score is then scaled between 0 to 1.

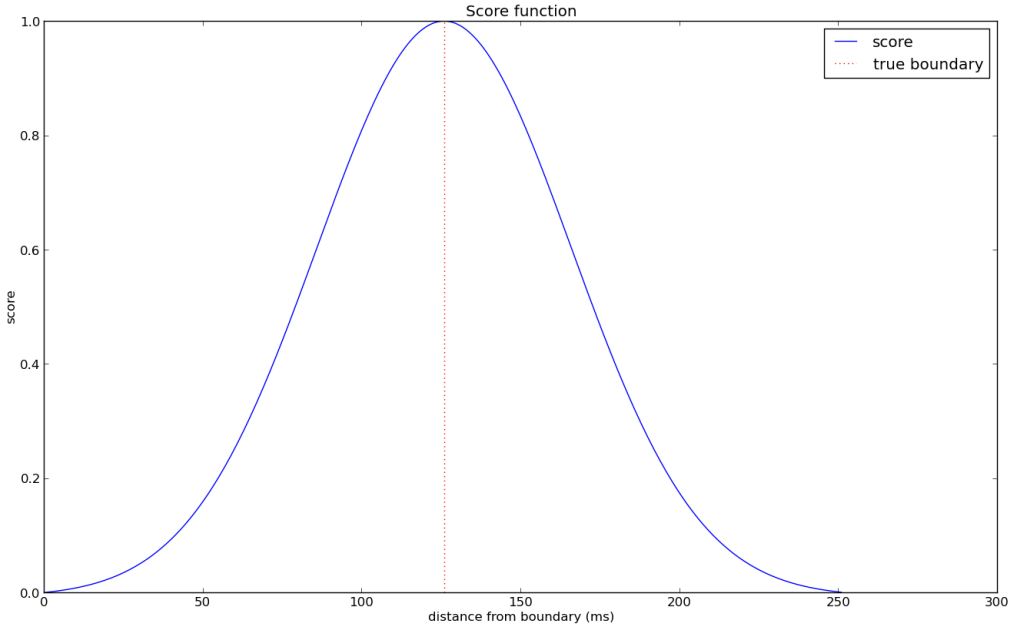

Figure 10: Score function

3.3 Feature extraction

For every phoneme boundary (during training and testing phase), we select 125 frames on either side. Each frame is 50ms wide (frame size) and the distance between adjacent frames is 1ms (hop size).

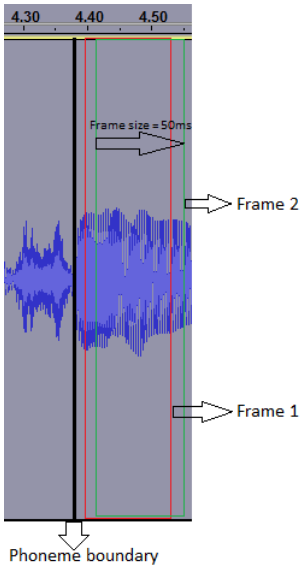

Figure 11: Example of candidate frames

Every such frame is a candidate boundary. To characterise each frame a number of features are calculated. From the initial review of the literature [21] we decided to use zero-crossing rate, log energy, bisector frequency [3], entropy [22], pitch, MFCCs (13 values) and their delta features. This gave a 36 dimensional feature vector for each candidate boundary.

Later on, we realised that the system could handle a larger feature set and the results could improve with this extended feature set. The new feature set consists of a 302 dimensional vector for each candidate boundary. This consists of a large number of tonal features, spectral features and their delta values. It also includes many features which remain constant for all frames around a given phoneme boundary. Log attack time, max to total energy ratio, loudness, effective duration, etc are a few such features. A complete list of all the features extracted can be found in the appendix.

3.4 Training of score prediction models

It refers to the training of machine learning models to be used for score predictive modelling. Support vector regression has been used for this previously [21]. Initially, we used support vector regression but the training time was very high. For one phoneme transition category, the time required for training was 120 minutes.

This was when a small subset (40%) of the boundaries were used. Considering that each pitch range has around 69 transition categories and there are 4 different pitch ranges, this approach won't scale very well. The major reason is that support vector regression has complexity $O(n^3)$ in the number of training points. Hence, we decided to try another regression model.

3.4.1 Random Forest Regression

This is an ensemble learning method, which generates many classiers and aggregates their results. Two well-known methods are boosting [24] and bagging [25] of classication trees. In boosting, successive trees give extra weight to points incorrectly predicted by earlier predictors. In the end, a weighted vote is taken for prediction. In bagging, successive trees do not depend on earlier trees each is independently constructed using a bootstrap sample of the data set. In the end, a simple majority vote is taken for prediction.
Random forests add an additional layer of randomness to bagging [26].In addition to constructing each tree using a different bootstrap sample of the data, random forests change how the regression trees are constructed. In standard trees, each node is split using the best split among all variables. In a random forest, each node is split using the best among a subset of predictors randomly chosen at that node. This somewhat counter intuitive strategy turns out to perform very well compared to many other classifiers, including discriminant analysis, support vector machines and neural networks, and is robust against overtting [26]. In addition, it is very user-friendly in the sense that it has only two parameters (the number of variables in the random subset at each node and the number of trees in the forest),and is usually not very sensitive to their values. This saves the time spent in grid searching for optimal parameter values. This is a large saving considering the different number of cases involved, as explained earlier.

The random forests algorithm can be specified as follows:

1. Draw n_{tree} bootstrap samples from the original data.

2. For each of the bootstrap samples, grow an unpruned classication or regression tree, with the following modication: at each node, rather than choosing the best split among all predictors, randomly sample m_{try} of the predictors and choose the best split from among those variables. (Bagging can be thought of as the special case of random forests obtained when $\mathrm{m}_{try} = \mathrm{p}$, the number of predictors.)

3. Predict new data by aggregating the predictions of the n_{tree} trees (i.e., majority votes for classication, average for regression).

An implementation of this algorithm available as a part of the scikit-learn package [28] is used, it is the backbone of this thesis. Important input parameters for this implementation are:

1. n estimators: The number of trees in the forest. A fixed value of 100 is used for this.

2. max features: The number of features to consider when looking for the best split. This is set to the number of input features i.e. 302.

3. max depth: The maximum depth of that each tree can take. This is set to 5000.

4. min samples leaf:The minimum number of samples in newly created leaves. A split is discarded if after the split, one of the leaves would contain less then min samples leaf samples. This is set to 10.

Since this is the training phase, we use the manually annotated phoneme boundaries to generate the candidate frames for training the models. Features are extracted from these as previously explained. This gives us a vector of 302 features and 1 score value for each frame. These are used as input for the random forest regression. The output of this phase is a model which predicts a score for a vector of 302 features for a given frame of audio data.

The curve below shows that the score predicted by trained models follows the true score closely. Each time the true boundary curve touches the x axis, it signifies the start of a new phoneme boundary. Ideally, the two curves should overlap but the aim is to replicate the shape of the curve so that the maxima of the predicted score function occurs very close (within 10ms) to the true boundaries. Due to the nature of random forest regression and the large number of features provided, this shape is achieved without doing a grid search on the parameter values. The true test of these models is how well they predict the score during the testing phase (unseen data).

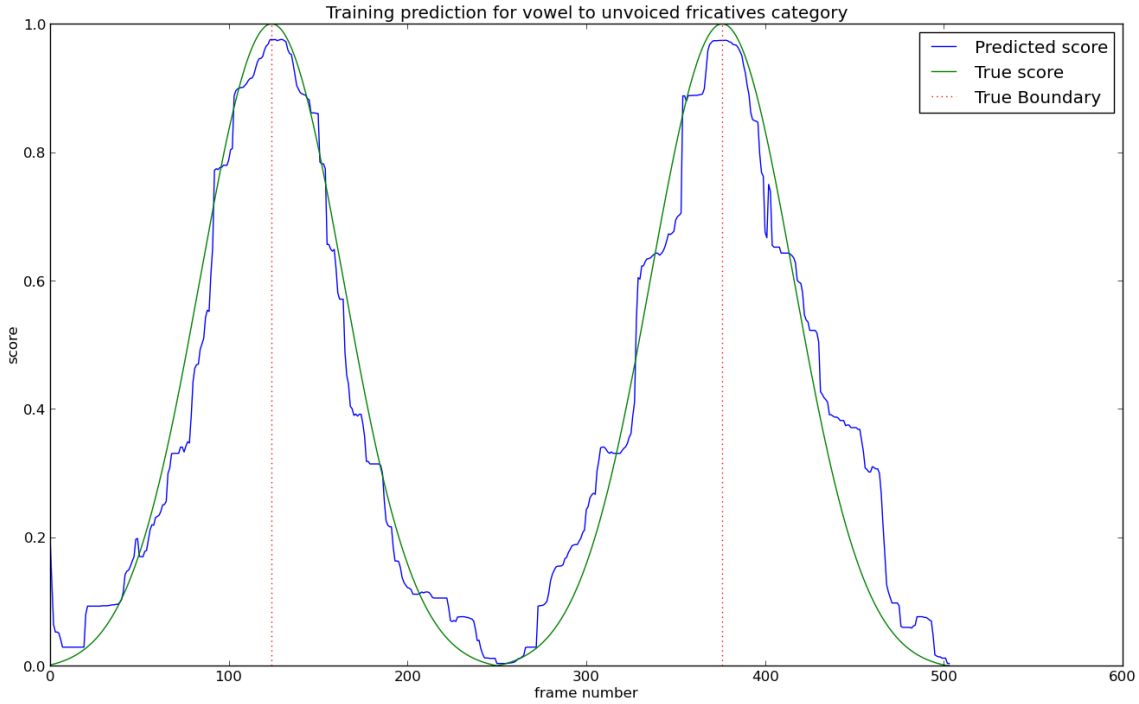

Figure 12: Predicted score during training for two boundaries of vowels to unvoiced fricatives transition

3.5 Testing phase

The purpose of this phase is to use the models generated previously to predict scores for candidate frames around boundaries predicted by HMM (inaccurate boundaries). The assumption here is that HMM boundaries are worse than manually annotated boundaries and hence can be refined. This is true because the HMM models used to generate these boundaries are trained for speech, so they don't capture the nuances of the singing voice. Using the predicted score, the frame with the highest score is chosen as the refined boundary.

Feature extraction is performed on the candidate frames around HMM boundaries. These features serve as input for the score prediction models previously generated. This process is carried out for each phoneme transition category.

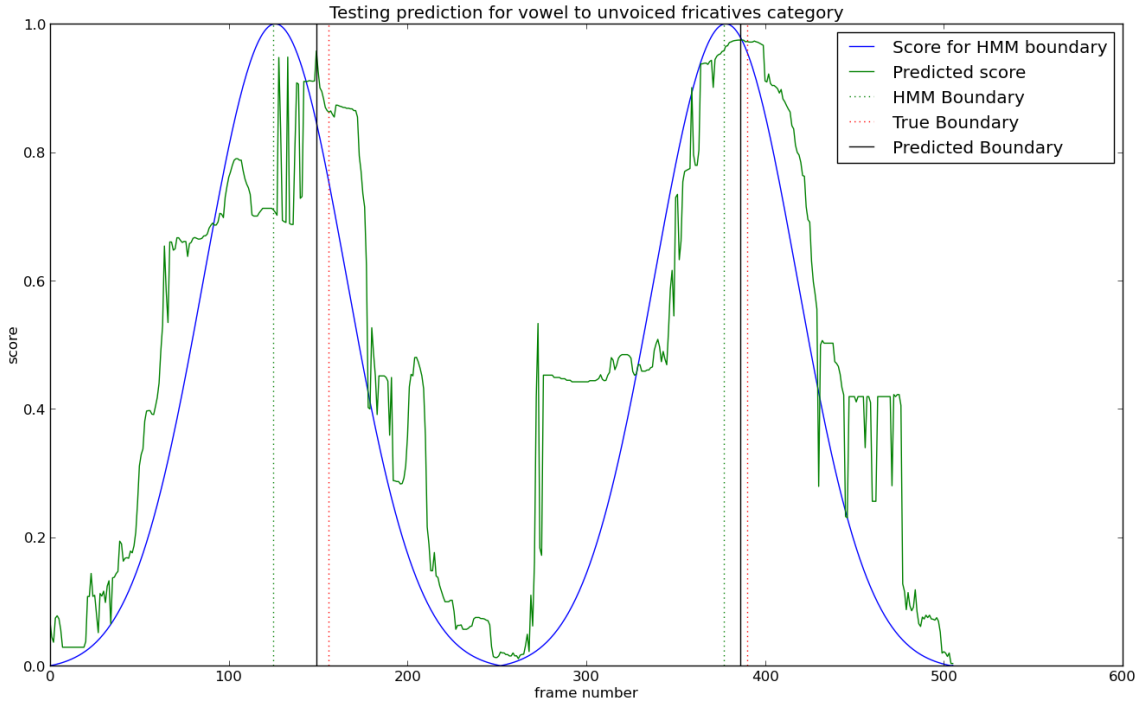

Figure 13: Predicted boundaries for vowel to unvoiced fricative transition

It can be seen in the above figure that the predicted boundaries for these two cases are closer to the true (manually annotated) boundary as compared to the HMM boundaries. These boundaries are said to be improved or refined by our method.

3.6 Evaluation

A generic evaluation strategy like cross validation is not insightful in our case. This is because the aim is to only follow the shape of the score function approximately using the regression models rather than exact replication. Moreover, even this is not strictly necessary because the only important factor is that the maxima of the predicted score is close to the true boundary.

Instead of cross validation, all the boundaries are split into a test set and train set. These sets are mutually exclusive. The train set contains 67 % of all the boundaries in each transition category and the testing set contains the rest of the 33 % of the boundaries. In the training phase the train set is used for modelling

and during the testing phase the test set is used for evaluation.

Evaluation is carried by plotting percentage segmentation accuracy curves which are very common for speech segmentation. These tell us what percentage of the predicted boundaries are within a certain distance of the true boundary. This is plotted for the boundaries predicted by score predictive modelling and the HMM boundaries. If the curve for the SPM boundaries stays above the curve for HMM boundaries it is judged as an improvement.

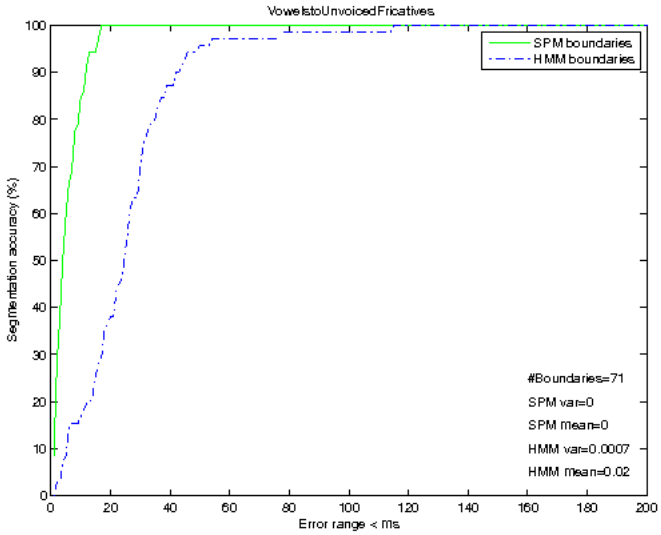

Figure 14: Segmentation accuracy for vowel to unvoiced fricative transition for mid pitch range

In the above figure, we see that the segmentation accuracy for the boundaries predicted by the score predictive modelling is higher than the HMM boundaries. This means that the SPM boundaries are closer to the true boundary as compared to the HMM boundaries. Also, the variance and mean of errors of SPM and HMM boundaries are shown on the right hand side of the figure.

4 Results

All the results stated below make use of the extended feature set containing 302 features unless stated otherwise.

4.1 Using models from the same pitch range

This section details results obtained when using the score predictive models trained and tested with audio from the same pitch range. First we present a case where a large improvement is obtained by using score predictive modelling (SPM) and later a case with low improvement.

4.1.1 Case where SPM improves over HMM

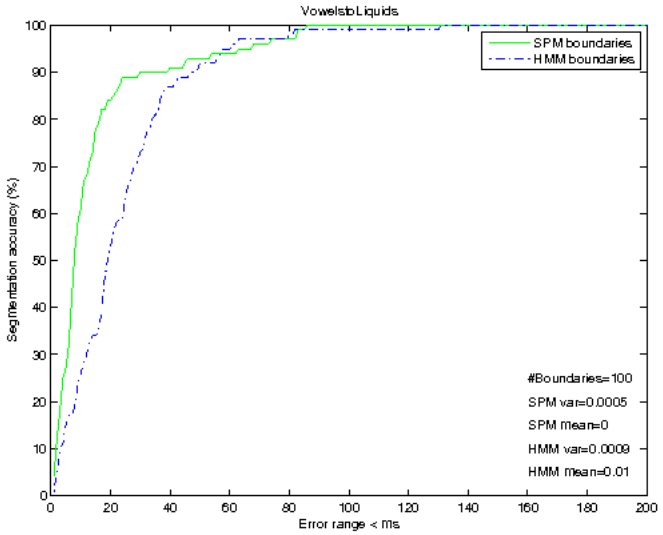

Figure 15: Segmentation accuracy for low pitch range

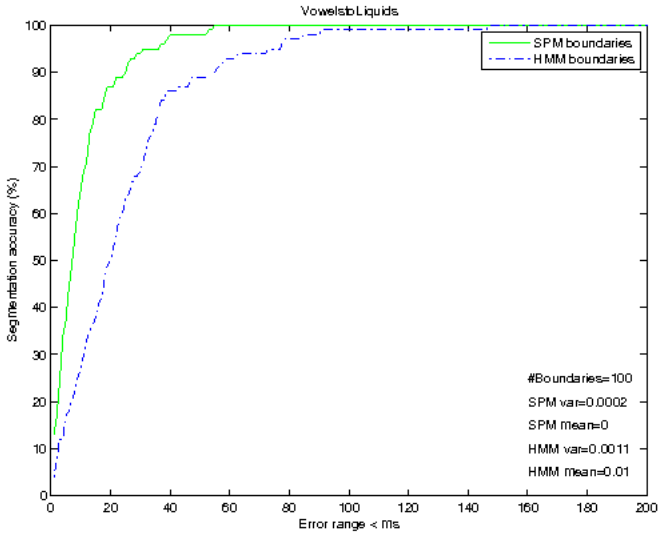

Figure 16: Segmentation accuracy for mid pitch range

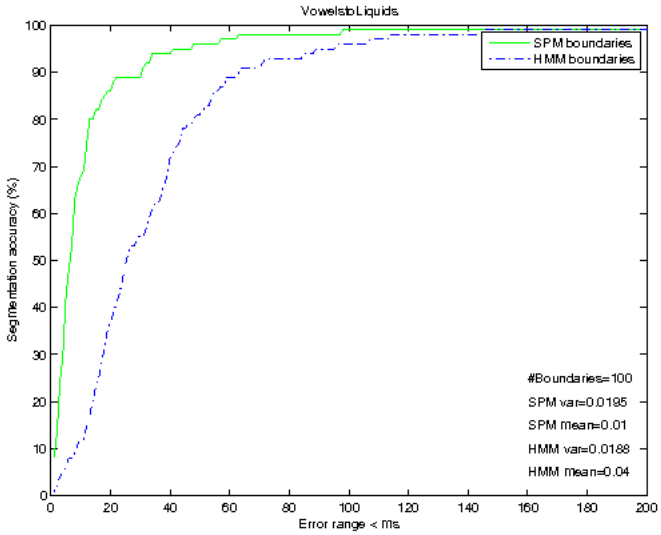

Figure 17: Segmentation accuracy for high pitch range

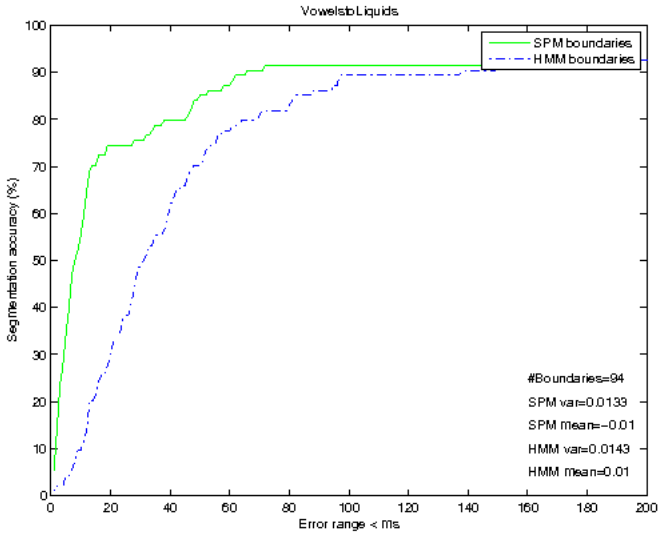

Figure 18: Segmentation accuracy for head pitch range

The above four figures show a case where SPM has higher segmentation accuracy as compared to HMM for vowels to liquids transition category (category with highest number of training boundaries).

4.1.2 Case where SPM shows low improvement over HMM

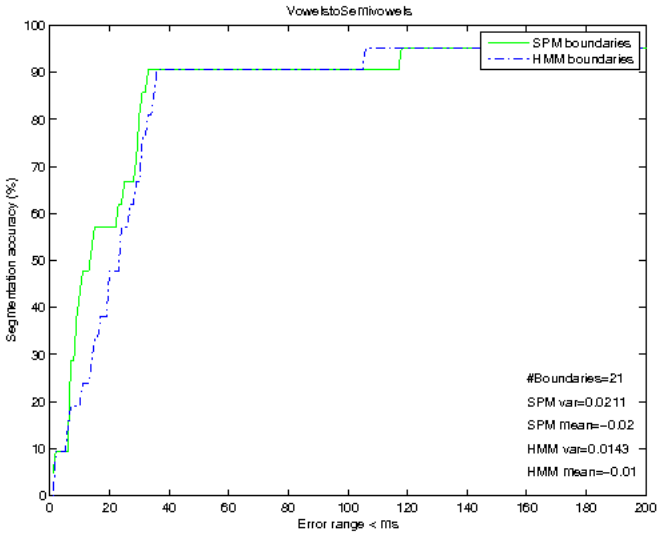

Figure 19: Segmentation accuracy for low pitch range

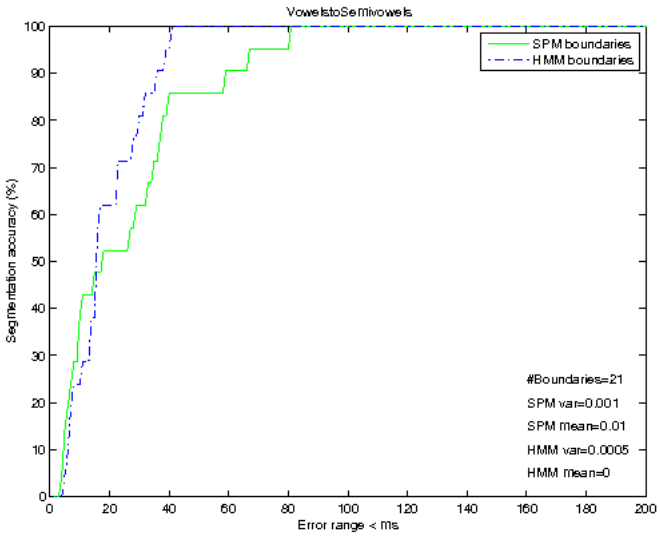

Figure 20: Segmentation accuracy for mid pitch range

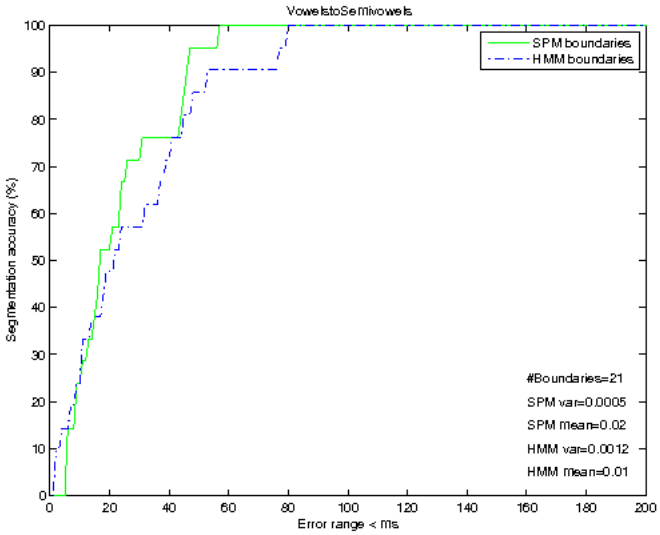

Figure 21: Segmentation accuracy for high pitch range

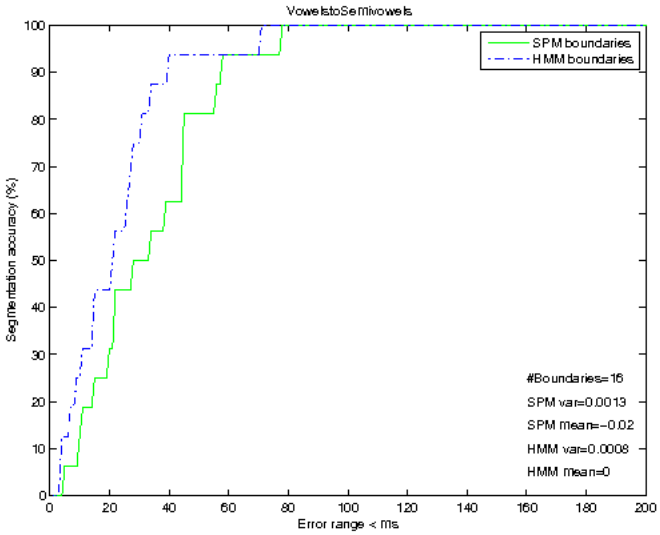

Figure 22: Segmentation accuracy for head pitch range

The above four figures show a case where SPM shows less improvement/degradation over HMM for vowels to semivowels transition category. This can partially be attributed to the lack of training data. The above case (vowels to semivowels) has only 63 boundaries for training as compared to 300 boundaries for training for the

previous case of vowels to liquids.

We observed that for most of the transition categories with number of training boundaries less than 100, no significant improvement is observed by using SPM. This can mainly be attributed to the lack of sufficient training data which is ultimately a property of the corpora. This can be used as a guideline while creating singing voice synthesizer corpora in the future, to ensure that the important transition categories have atleast a 100 training boundaries.

4.2 Using feature sets of different sizes

As mentioned earlier, we started with a feature set of 36 features. This feature list was taken from the original work [21] on score predictive modelling. Later, we scaled this feature set to include 302 features.

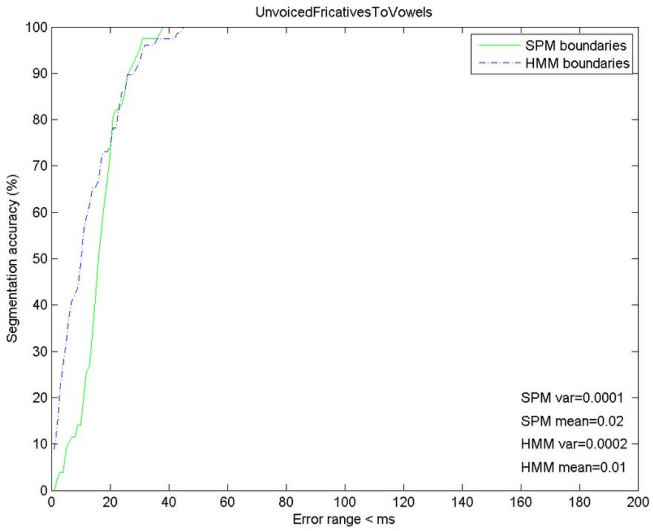

Figure 23: Segmentation accuracy for mid pitch range trained with 36 features

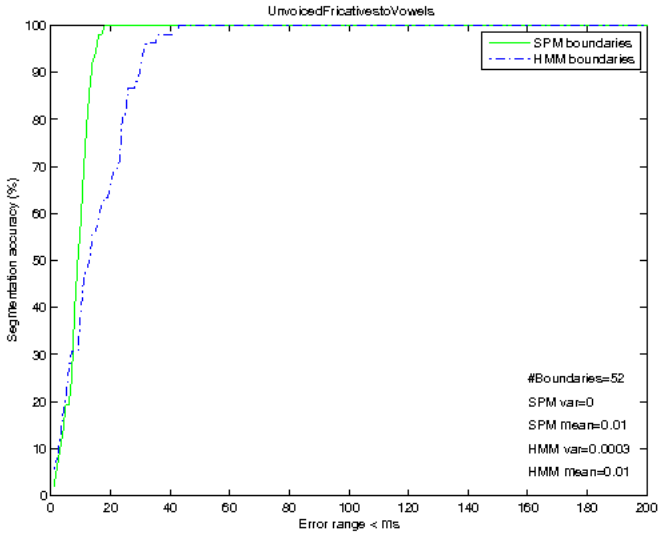

Figure 24: Segmentation accuracy for mid pitch range trained with 302 features

As can be seen above, the segmentation accuracy is improved by using a larger features set. This is at the cost of increased time for feature extraction, training and testing but the increase in processing time is not very significant.

It is observed that using the large feature set all the transition categories with greater than 100 training boundaries show an improved segmentation accuracy. If we don't consider cases which don't have significant number of training boundaries, this suggests an improvement with our method across all the transition categories which was not the case when using the smaller feature set. This is a very promising result as it justifies using a larger feature set with score predictive modelling.

4.3 Using models from different pitch ranges

This section details results obtained when using the score predictive models trained in one pitch range and tested with audio from another pitch range. The idea is to test for generalisation across pitch ranges. This might sometimes be desirable if there is not enough training data in a particular pitch range.

4.3.1 Case where this improves segmentation accuracy

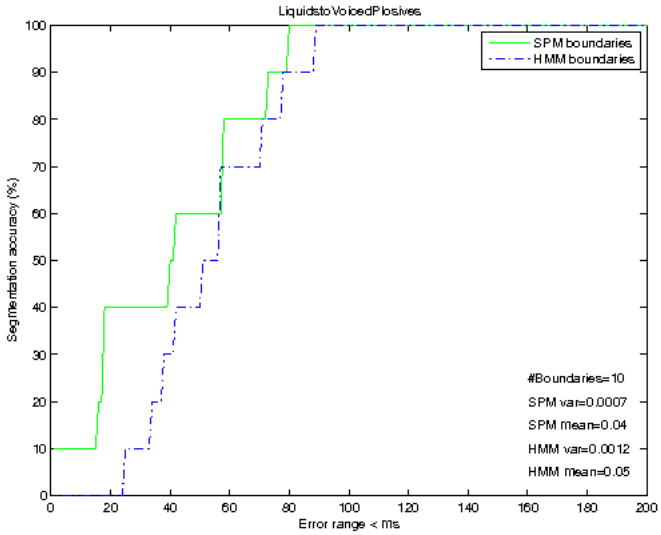

Figure 25: Segmentation accuracy for the high pitch range using models trained in the high pitch range

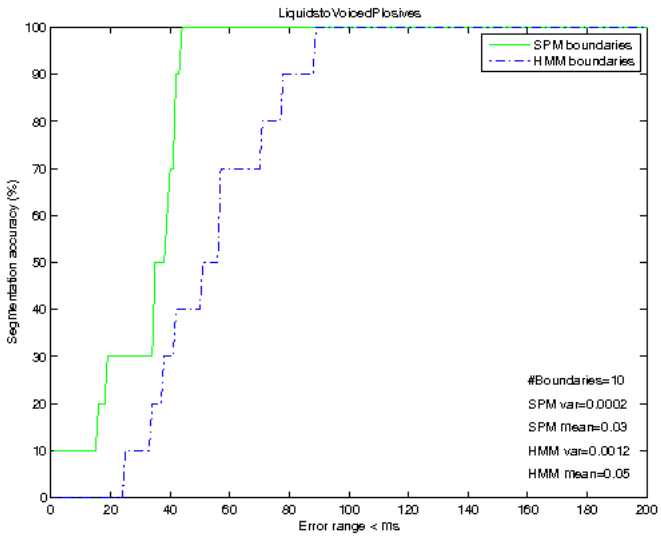

Figure 26: Segmentation accuracy for the high pitch range using models trained in the low pitch range

The accuracy in the second case with models trained in the low pitch range is higher than the first case. Surprisingly, this low pitch model doesn't give a good segmentation accuracy when tested in the low pitch range (where it was trained) as can be seen in the figure below.

4.3.2 Case where this reduces the segmentation accuracy

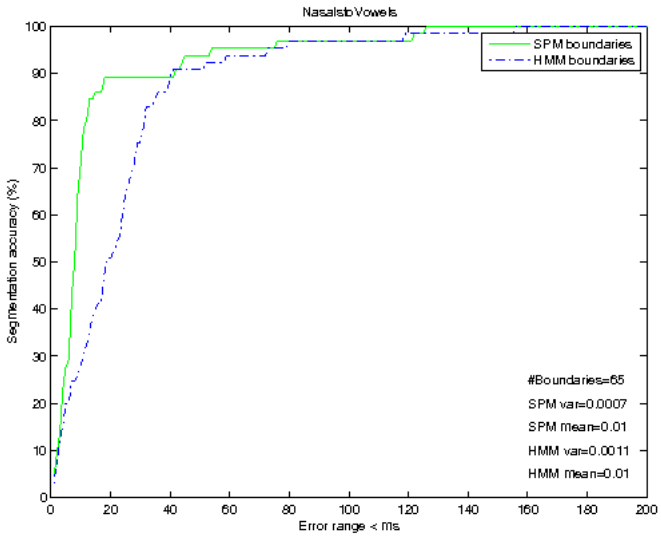

Figure 27: Segmentation accuracy for the high pitch range using models trained in the high pitch range

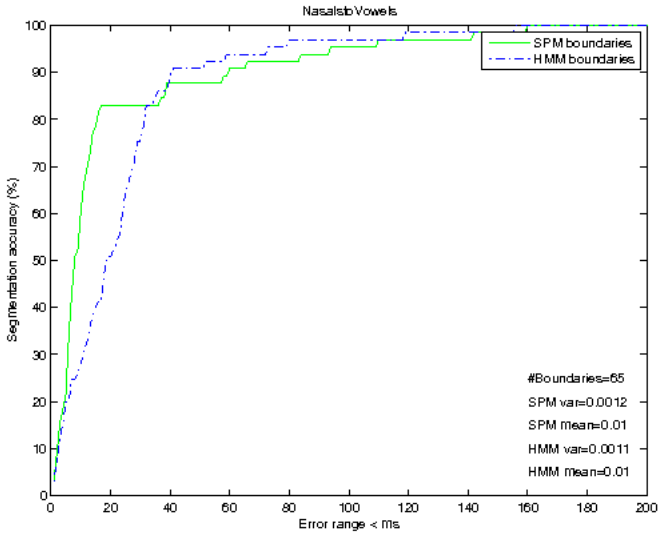

Figure 28: Segmentation accuracy for the high pitch range using models trained in the low pitch range

For the majority of transition categories the segmentation accuracy seems to decrease by using models trained in a different pitch range. Otherwise, no specific pattern is observed when using models trained in a different pitch range for testing i.e. we can not conclude whether it is beneficial for a particular transition category or whether it is more advantageous to use the models trained on a higher pitch range in a lower pitch range.

5 Future work

A number of different directions can be taken in the future using the work in this thesis as a starting point.

5.1 From boundary refinement to a new phonetic segmentation approach

An important drawback of our boundary refinement method is that it can only search for a more accurate boundary within 250ms of an already existing HMM boundary. If the true boundary is not within 250ms of the HMM boundary, then the boundary refinement result is not very relevant. This means that the initial HMM boundary is very inaccurate and any improvement on this is not meaningful. So, it would be beneficial to incorporate score predictive modelling in a method

which doesn't rely upon HMM for an initial boundary estimate.

Our corpora is recorded at a constant tempo. This means that vowel onsets and beat onsets are very close (ideally they should coincide). This can be used for initial segmentation into vowel to vowel, silence to vowel and vowel to silence. Then, phoneme duration likelihood information and the scores from score predictive modelling are used as criterion for guiding a dynamic programming algorithm which would optimally segment vowel to vowel segments into phonemes. This would turn our boundary refinement approach into a phonetic segmentation approach.

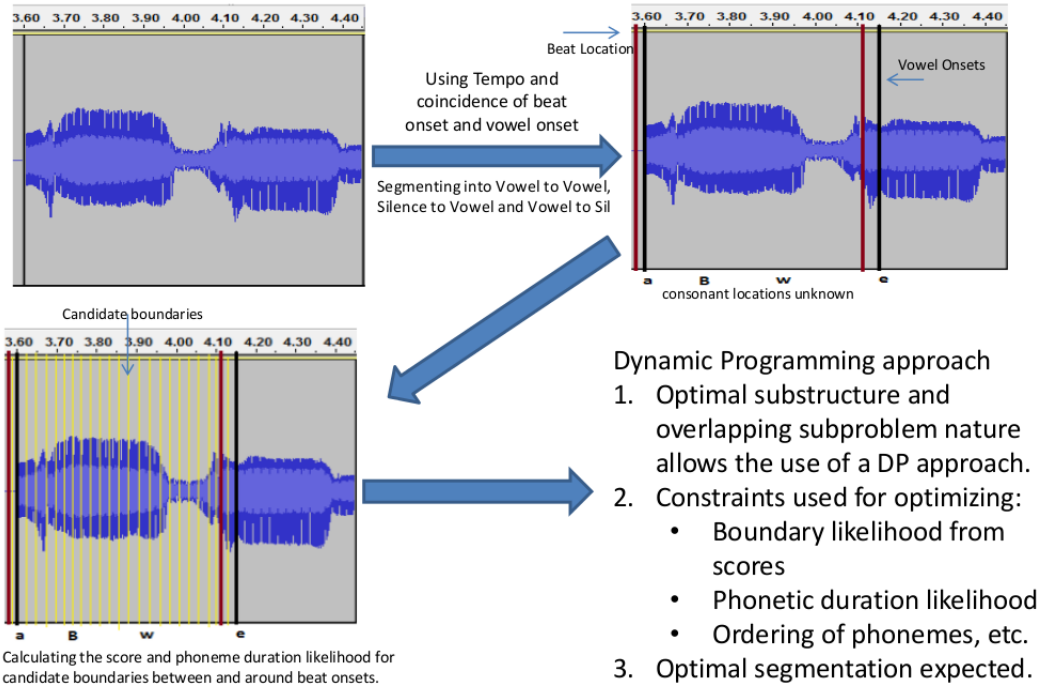

Figure 29: Dynamic programming approach for phonetic segmentation

5.2 Testing SPM for language independence

A recurring problem for using singing voice synthesizers in a variety of languages is the unavailability of manually segmented corpora for these languages. The main problem is the cost of manually segmenting recordings made by singers. A way to solve this could be to train the score predictive models with a few languages and use these conglomerative models for phonetic segmentation of new languages. That is to say that train the models with some languages for which manually segmented databases exist and use these models on the new languages for which these corpora don't exist. This is based on the assumption that languages have

phonetic similarities and hence, the segmentation models which are robust in the phonetic space of one language can be used directly in a similar phonetic space of another language.

6 Conclusions

We started by introducing speech segmentation and then phonetic segmentation. Subsequently, we explained how phonetic segmentation is important for singing voice synthesis. A brief survey of methods commonly used for speech segmentation was presented with adaptations of these methods for segmentation of singing voice synthesis corpora. Eventually, we introduced the idea of boundary refinement and how it is relevant for this work and specially for our corpora.

This gave us a clear view of the problem this work tries to tackle. We then presented a new adaptation of an existing method called score predictive modelling. The various parts of this method were then explained detailing how the existing method has been adapted to suit the singing voice synthesis corpora at hand. Finally, we evaluated this method under various conditions.

6.1 Contributions

The major contributions of this thesis:

- **Review of works**: The starting sections provide a brief review of relevant works and a non-technical introduction to phonetic segmentation for singing voice corpora.

- **Adaptation of score predictive modelling**: We adapted an already existing method for a different type of singing voice synthesizer corpora and for a different language. The adaptation is also tried with different feature sets, something which was not attempted earlier. The increase in size of the feature set gives a large improvement in the results. Score predictive models were also trained in one pitch range and tested in another pitch range. Even though the results of this are not conclusive but it had not been attempted before.

- **Use of random forest regression models**: The original score predictive modelling approach uses support vector regression for modelling the scores from the feature set. This is computationally very expensive due to the $O(n^3)$ complexity of support vector machines (n is the number of data points). We decided to replace this with random forest regression models and this has given us a large improvement in the time taken for training and testing of the

models (almost 10 times quicker now). Also, due to the nature of random forest regression models, its parameters don't need much tweaking and this adds to the ease of implementation and use of our approach.

- **Framework implementation**: The feature extraction, training/testing, HMM boundary estimation and evaluation code is written using open source toolkits in Python and is available for anyone to use. Such a framework is not currently available and we hope that it could serve as a starting point for future work in this area.

- **New dynamic programming approach**: We suggest a way to turn score predictive modelling from a boundary refinement method to a phonetic segmentation approach. This would make it independent of HMMs and this approach might even be applicable to speech segmentation problems.

References

[1] J. S. Jang and M. Y. Gao,"A query-by-singing system based on dynamic programming", in Proc. Int. Workshop Intell. Syst. Resolutions, 2000, pp. 85-89.

[2] J. S. Jang and H. R. Lee, "Hierarchical filtering method for content based music retrieval via acoustic input", in Proc. ACM Multimedia, 2001, pp. 401-410.

[3] Lin, Cheng-Yuan, Kuan-Ting Chen, and J-S. Roger Jang. "A hybrid approach to automatic segmentation and labelling for Mandarin Chinese speech corpus" Proceedings of the 9th European conference on speech communication and technology (EUROSPEECH 2005). 2005.

[4] Makashay, Matthew J., et al. "Perceptual evaluation of automatic segmentation in text-to-speech synthesis." Proc. ICSLP. Vol. 2. 2000.

[5] van Santen, Jan, and Richard Sproat. "High-accuracy automatic segmentation." Proc. Eurospeech. Vol. 6. 1999.

[6] Malfrre, Fabrice, Olivier Deroo, and Thierry Dutoit. "Phonetic alignment: Speech synthesis based vs. hybrid hmm/ann." Proceedings of the ICSLP. 1998.

[7] Cosi, Piero, Daniele Falavigna, and Maurizio Omologo. "A preliminary statistical evaluation of manual and automatic segmentation discrepancies." Proc. of EUROSPEECH-1991 (1991): 693-696.

[8] Toledano, Doroteo Torre, LA Hernndez Gmez, and Luis Villarrubia Grande. "Automatic phonetic segmentation." Speech and Audio Processing, IEEE Transactions on 11.6 (2003): 617-625.

[9] Ljolje, Andrej, Julia Hirschberg, and Jan PH van Santen. "Automatic speech segmentation for concatenative inventory selection." Progress in Speech Synthesis, Springer (1996): 305-311.

[10] Angelini, Bianca, Claudia Barolo, Daniele Falavigna, Maurizio Omologo, and Stefano Sandri. "Automatic diphone extraction for an Italian text-to-speech synthesis system." In Fifth European Conference on Speech Communication and Technology. 1997.

[11] Malfrere, Fabrice, and Thierry Dutoit. "High quality speech synthesis for phonetic speech segmentation." Eurospeech97. 1997.

[12] Farhat, Azarshid, Guy Perennou, and Regine Andre-Obrecht. “A segmental approach versus a centisecond one for automatic phonetic time-alignment.” Third European Conference on Speech Communication and Technology. 1993.

[13] Chou, Fu-chiang, Chiu-yu Tseng, and Lin-shan Lee. “Automatic segmental and prosodic labelling of Mandarin speech database.” Proceeding of the fifth international conference on spoken language processing. 1998.

[14] Beet, S. W., and L. Baghai-Ravary. “Automatic segmentation: data-driven units of speech.” Proc. Eurospeech. Vol. 97. 1997.

[15] Gholampour, Iman, and Kambiz Nayebi. “A new fast algorithm for automatic segmentation of continuous speech.” Proc. ICSLP. 1998.

[16] Hatazaki, Kaichiro, et al. “Phoneme segmentation using spectrogram reading knowledge.” Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on. IEEE, 1989.

[17] Karjalainen, Matti, Toomas Altosaar, and Miikka Huttunen. “An efficient labelling tool for the QUICKSIG speech database.” Proceedings of the International Conference on Spoken Language Processing (1998): 1535-1538.

[18] Saito, Takashi. “On the use of F0 features in automatic segmentation for speech synthesis.” Proceedings of ICSLP. Vol. 7. 1998.

[19] Glass, James R., and Victor W. Zue. “Multi-level acoustic segmentation of continuous speech.” Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on. IEEE, 1988.

[20] Ljolje, Andrej, Julia Hirschberg, and Jan PH van Santen. “Automatic speech segmentation for concatenative inventory selection.” Progress in Speech Synthesis, Springer (1996): 305-311.

[21] Lin, Cheng-Yuan, and Jyh-Shing Jang. “Automatic phonetic segmentation by score predictive model for the corpora of mandarin singing voices.” Audio, Speech, and Language Processing, IEEE Transactions on 15.7 (2007): 2151-2159.

[22] Shen, Jia-lin, Jeih-weih Hung, and Lin-shan Lee. “Robust entropy-based endpoint detection for speech recognition in noisy environments.” ICSLP. Vol. 98. 1998.

[23] Bonada, Jordi, and Xavier Serra. “Synthesis of the singing voice by performance sampling and spectral models.” Signal Processing Magazine, IEEE 24.2 (2007): 67-79.

[24] Schapire, Robert E. “A brief introduction to boosting.” Ijcai. Vol. 99. 1999.

[25] Breiman, Leo. “Bagging predictors.” Machine learning 24.2 (1996): 123-140.

[26] Breiman, Leo. “Random forests.” Machine learning 45.1 (2001): 5-32.

[27] Liaw, Andy, and Matthew Wiener. “Classification and Regression by randomForest.” R news 2.3 (2002): 18-22.

[28] http://goo.gl/QOzxEA

Appendix A
Complete feature list and code repository

Predominant melody pitch, predominant melody confidence, after maximum energy to before maximum energy ratio, effective duration, long-term loudness, log attack time, maximum value index to total length of the envelope, temporal centroid to total length of envelope, spectral contrast, spectral valley, spectral complexity, dissonance, energy ratio of low frequency bands, energy ratio of middle-low frequency bands, energy ratio of middle-high frequency bands, energy ratio of frequency high bands, spectral flatness, spectral flux, spectral rolloff, spectral strong peak, high frequency content of the spectrum, equal tempered deviation, non tempered energy ration, non tempered peaks energy ratio, harmonic spectral centroid, harmonic spectral deviation, harmonic spectral spread, inharmonicity, linear predictive coefficients, linear predictive reflection coefficients, the log-energies in mel bands, the mel frequency cepstrum coefficients, frequency with maximum magnitude, odd to even harmonic energy ratio, tristimulus, zero crossing rate and their delta values.
Details about features: http://essentia.upf.edu/documentation/algorithms_reference.html

Code repository: https://github.com/neo01124/ModifiedSPM