AUTOMATIC DISCOVERY OF A PHONETIC INVENTORY FOR UNWRITTEN LANGUAGES FOR STATISTICAL SPEECH SYNTHESIS

Prasanna Kumar Muthukumar, Alan W Black

Language Technologies Institute, Carnegie Mellon University, Pittsburgh, USA

ABSTRACT

Speech synthesis systems are typically built with speech data and transcriptions. In this paper, we try to build synthesis systems when no transcriptions or knowledge about the language are available. It is usually necessary to at least possess phonetic knowledge about the language. In this paper, we propose an automated way of obtaining phones and phonetic knowledge about the corpus at hand by making use of Articulatory Features (AFs). An Articulatory Feature predictor is trained on a bootstrap corpus in an arbitrary other language using a three-hidden layer neural network. This neural network is run on the speech corpus to extract AFs. Hierarchical clustering is used to cluster the AFs into categories i.e. phones. Phonetic information about each of these inferred phones is obtained by computing the mean of the AFs in each cluster. Results of systems built with this framework in multiple languages are reported.

Index Terms— Speech synthesis, TTS without text, unlabeled speech corpora, articulatory features, neural networks

1. INTRODUCTION

For speech technology to be practical for the majority of languages in the world, we must deal with languages where there is no clear standard orthography. This work concerns building speech synthesis systems for languages where only audio is available and there is no readily available writing system. Such languages include those for which there is no writing systems at all, but also those for which only an ill-defined version exists (such as spoken dialects of Arabic and Chinese). This paper continues our existing work of discovering a consistent symbolic representation of speech suitable for synthesis. Of course this work will require further integration into other systems. For example, in one possible direction, we can envisage a cross lingual synthesis system that takes as input a standard written form in a source language and combines both translation and speech synthesis technologies to produce speech synthesis in the target unwritten language. Thus our discovered written form for the unwritten language must not only produce a symbolic phonetic form suitable to encode speech, but also be appropriately abstract enough that we can identify word-like forms that will allow translation from the typically better resourced written language. Thus we can create a synthesizer that takes Hindi text as input and produces Konkani speech, or a system that takes High German text as input but produces Bavarian Speech.

2. RELATED WORK

We have already utilized cross lingual phonetic recognition technology in finding an initial phonetically influenced symbolic representation of audio in our target unwritten language. This cross-lingual phonetic segmentation is adapted by retraining to allow those segmentations to better present the phonetic distinctions in the target language [1]. But the symbols assigned to these segmentations remain the same, and any phonetic distinctions in the target language that do not exist in the source language must be encoded in the contextual tree prediction within the statistical parametric synthesizer.

This work tries to address this limitation by providing a mechanism, that is phonetically inspired that will allow us to better rename these segments given by Automatic Speech Recognition.

To do this we appeal to the notion of articulatory phonetic features as defined in IPA [2], and work that has created multi-stream classifiers for these features in arbitrary speech [3]. Thus, going in the direction of a wider inventory of cross-lingual phonetic forms (e.g. as has been successful Globalphone type systems [4]) we make use of a lower level component of phonetic variation. Cross lingual articulatory features (AFs), have been used successfully in other work [5]. The second, and we believe novel, use of AFs is to use their predictive relationship with Mel Cepstral features to discover new clustering of the segmental units found by the first cross-lingual phonetic recognizer. Thus we treat the initial labelling as a broad phonetic class recognizer, but use the AF/MCEP relationship to find an appropriate number of new phonetically inspired labels for the initial segments.

3. GETTING FEATURES INDEPENDENT OF SPEAKER AND LANGUAGE

A typical speech synthesizer is built using a speech corpus and transcripts that correspond to the recorded speech where the synthesizer is made to learn the mappings between the text and the sounds of the speech signal. In our task, the absence of text information is a somewhat bewildering issue, because this is equivalent to a machine learning problem with only output labels and no input data. We attempt to solve this by inverting it so that we have input data but no output labels so that this becomes an unsupervised learning problem.

At the very least, training a synthesizer would require a phonetic transcription of the corpus. For this task, we assume

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

that we know *nothing* about the language of the speech corpus. So, not only do we lack a phonetic transcription, but we also know nothing about what the phones of the language are. Our first task is therefore to try to automatically discover the phonetic set of our speech corpus. It is extremely difficult to discover the *true* phones of a language without careful linguistic analysis. So, for this paper, we will focus on getting a "phone-like" unit of speech that is consistent throughout the corpus.

The most obvious way to discover phones would be to try some sort of unsupervised clustering; but what are the *features* that should be clustered? The usual speech features used for most tasks are signal processing features such as Linear Prediction Coefficients[6] or Mel Frequency Cepstral Coefficients[7]. These features do an excellent job of representing the speech signal. However, none of these let us separate speaker specific characteristics, emotion specific characteristics, and language specific characteristics. So, doing an unsupervised clustering on one of these features might not give us clusters that are phonetically relevant. Ideally, we would want to apply our clustering process on features that contain little speaker or language specific information.

There has been work in the speech recognition community on trying to find such features. One such set of features which are particular useful for our task are called **Articulatory Features** (AFs). Vowels in human speech can be viewed in a chart indexed by the frequency of the first two formants. We can identify vowels as being in two dimensions, high to low(F1), and front to back(F2). Other ways of describing vowels include nasality, length, stress, and tone. Consonants too can be broken down into a set of features that distinguish stops, fricatives, affricates etc... Articulatory Features describe the phones in terms of parameters like these. These are described in detail in [8], [9], [10], [11], and [3]. These should not be confused with the features described in [12] which describe how the articulators of speech actually move.

Various combinations of these articulatory features define phones in a language. While different languages and dialects differ in which combinations form phones, the articulatory features themselves are independent of language. This is because the articulatory features are based on the phonetics of human speech production, not on the phonology of specific languages. Articulatory features are also, for the most part, independent of speaker. For this paper, we use the 26 binary articulatory features described in [9].

4. TRAINING AN ARTICULATORY FEATURE DETECTOR

Articulatory feature extraction is done by bootstrapping from a speech corpus for which we *do* have transcriptions at the text or the phonetic levels. This bootstrap corpus can be in any arbitrary language, or even a mixture of several languages. It is desirable though to have a corpus with several hours of speech from multiple speakers. For our task we used the Wall Street Journal corpus[13], which has multiple speakers, but is all standard US English.

A simple table lookup is used to obtain articulatory feature information corresponding to each phone in our bootstrap corpus. While extracting features from data with phone labels is trivial, extracting these directly from the speech signal requires a more sophisticated approach.

Mel Cepstral Coefficients[14] are used as input features from which Articulatory Features are predicted. The mapping from one to the other is learned using a three hidden layer feedforward network with sigmoid activations using the QuickNet toolkit[15]. 26 neural networks were trained in this manner (one for each AF).

4.1. Testing articulatory features

Each of the neural networks we trained has an accuracy between 85-99% on cross-validation sets. However, we wanted to do some experiments on real data to verify that our predicted AFs were reasonable. To do this, we used the EHMM labeling tool as described in [16] to do a forced alignment between phone labels and articulatory features in the RMS voice of the CMU Arctic database[17]. EHMM gives us an alignment that maximizes the likelihood of the phone labels having generated the articulatory feature data. A synthesizer built using this alignment would only give us sensible results if the articulatory feature extraction works reasonably well. So, this is an independent way of evaluating the quality of our articulatory feature extraction. The synthesizer is compared to one built with alignments obtained using Mel Cepstral features. The ClusterGen speech synthesizer is used in all our experiments[18]. Mel Cepstral Distortion (MCD) on a separate test set is used as an objective metric for evaluating the quality of speech synthesis. This experiment was also replicated on a Hindi speech corpus[19]. The results obtained are shown in table 1

Table 1. MCD Scores

MCD Scores	English	Hindi
EHMM on MCEPs	5.1610	5.255
EHMM on AFs	5.4752	5.665

The synthesizer built with alignments using AFs does not perform as well as the one that used MCEPs. However, the difference in MCD between the two is small enough that it indicates that the articulatory features obtained through the methods described above are reasonable.

5. PHONE SEGMENTATION

Obtaining phonetic information about a speech corpus not only requires a consistent phone representation but also involves *segmenting* the speech corpus at the phone level. The task of doing phone segmentation would be greatly simplified if we had an approximate estimate of the number of phones for each utterance in our corpus. We obtain a first approximation of this number by running a Automatic Speech Recognizer in phone recognition mode. For our task, we used the CMU Sphinx speech recognition system[20] in **all phone** mode and the iterative procedure developed in [21] to get initial phone transcripts. We must remember that an English

phone recognizer gives us *English* phones and so is unlikely to produce a good phonetic transcription of our foreign language corpus. For example, if we were to run this recognizer on a Hindi corpus, the recognizer would not distinguish between the aspirated and unaspirated stops in Hindi. Or if it were run on a Japanese corpus, the recognizer would try to make distinctions between the liquids 'L' and 'R' while the language itself does not.

While the recognizer will make errors in assigning labels to the phones in an utterance, we are at this stage caring more about the segmentation rather than the labels of those segments. Doing a forced alignment using the EHMM tool between the phone labels and the file containing the audio of the utterance will give us a fairly good estimate of the segment boundaries. In our earlier work, [1], we took the provided phone names as is, here we are discovering a better set.

6. CLUSTERING AND LABELING PHONES

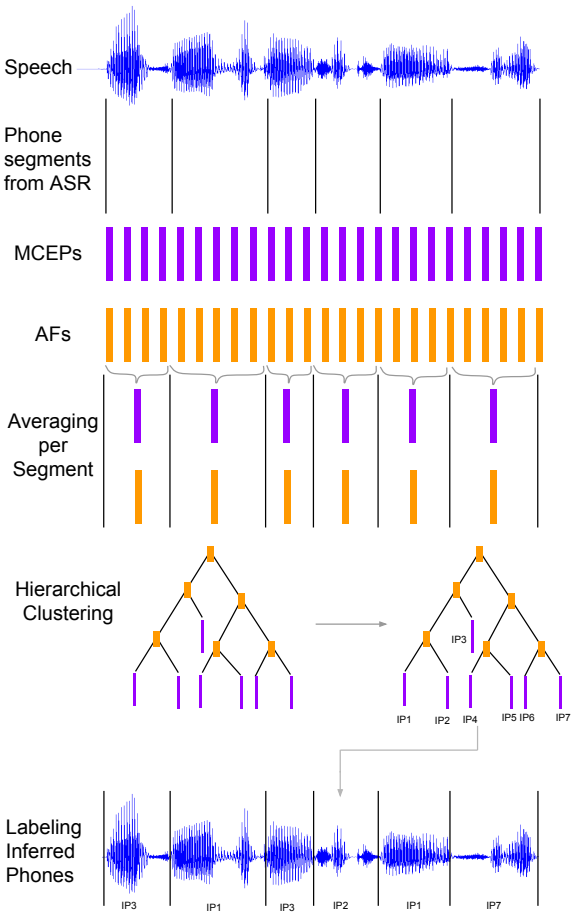

Fig. 1. Clustering and Labeling Framework

Once phone segmentation is done, we have to face the difficult problem of assigning labels to these phones. The MCEP parameters that we extract from the speech represent the speech signal in 25ms frames at 5ms shifts. The AFs which are extracted from the MCEPs also are of the same time intervals. However, the phone segments themselves are much longer than the length of a frame. For each phone segment, we *average* the MCEPs and the AFs in that segment so that we have an average MCEP vector, and an average AF vector for that phone segment. Our task is now to find a set of phone labels which gives a consistent description of these average vectors in our hypothetical database. Before we consider possible schemes of label assignment, we have to tackle the issue of evaluating how good our labeling schemes are. It is reasonable to assume that the more consistent and accurate the phone labels are, the easier it is for the synthesizer to model a voice based on that data. A good set of phone labels will therefore result in a good MCD score.

The most straightforward way to automatically assign a consistent set of phone labels is to use some sort of clustering method. As mentioned earlier, it is advantageous to cluster the Articulatory Features, rather than the MCEPs themselves, because of their speaker and language independent characteristics. We build a Classification And Regression Tree (CART)[22] that forms a hierarchical clustering of the AF vectors, where the leaf nodes contain Gaussians modeling the MCEP vectors that correspond to the AF questions that are asked at higher levels of the tree. The clustering algorithm works by minimizing the variance of the MCEP coefficients at the leaf nodes by asking questions about the AF vectors. Overfitting is prevented by specifying a minimum number of MCEP vectors from the training data that must correspond to each leaf of the tree (also called the **stop size**).

Once we have built a tree with AF questions and MCEP leaves using our entire corpus, each leaf will now be a fairly reliable cluster that corresponds to an acoustically derived phone. This tree is then used to assign labels to the phone segments in our corpus. We call these labels **Inferred Phones** or **IPs** because they are inferred from purely acoustic information with no human involvement. One important issue in doing this is deciding the stop size of the tree i.e. deciding how many leaves and therefore how many phones we should infer from the corpus. There is no clear answer to this problem because it is difficult even for linguists to give a definitive number for phones of a language. We will instead report results that we obtained with a range of phoneset sizes. In our experience, nearly all languages in the world have between 20-200 phones. So, we use that as bounding range within which we will do our experiments.

6.1. Experiments and results

We ran experiments on four languages, Hindi, Dari, Iraqi, and English. The results obtained for Hindi, Dari, and Iraqi are shown in Figure 2. Comparisons are made between ClusterGen synthesis systems built with various numbers of Inferred Phones and the baseline system which was built with phones from ASR. To make a fair comparison of the inferred phones to the ASR phones, we assumed that we did not possess any phonetic feature information about the ASR phones either. Empirically we observed that this assumption increased the

baseline MCD by about 0.07. As can be seen, the IP systems in Hindi and Iraqi perform better than the baseline system. In Dari, the IP systems come close to the baseline but do not perform better. However, the trend of the Dari IP systems is similar to that of the other languages. As a comparison with

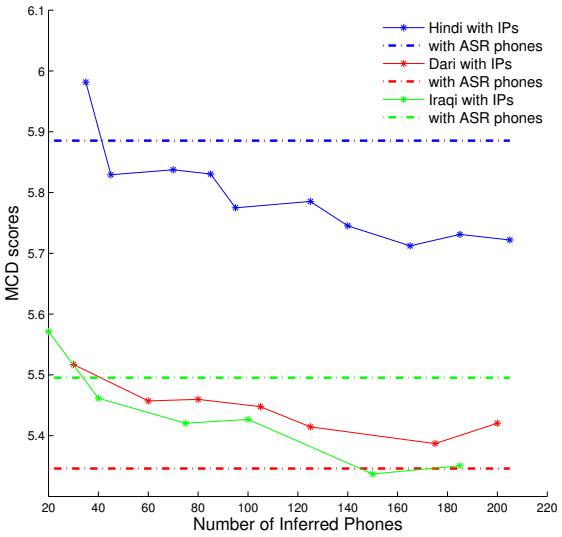

Fig. 2. MCD Scores for Inferred Phones

a language that we have much more experience with, we also tested an English system using the true phonetic transcription as the baseline to look at the difference in performance between using knowledge-based phonetic transcriptions compared with inferred phones. The phonetic transcription was obtained using the CMUDICT phonetic lexicon[23] on the original text. The baseline MCD was 5.702. The best IP system, with 190 inferred phones, had an MCD of 5.904. This result is quite promising considering that a completely data-driven phoneset only has a 0.2 MCD difference from the baseline system which had the best possible phonetic transcription.

7. GETTING PHONETIC FEATURES FOR INFERRED PHONES

One potential problem that usually occurs with a completely data driven phoneset, as opposed to using phone recognition, is that it is usually impossible to get any phonetic information about inferred phones.

Our technique has a big advantage here since articulatory features are fundamental to our clustering process. By looking at the articulatory features that correspond to our inferred phones, it is straightforward to derive phonetic information for these.

Each of the leaves of our CART that we had used for clustering is a cluster that corresponds to a particular Inferred Phone. The mean AF of all the AF vectors in that cluster gives us a good estimate of the phonetic information about that Inferred Phone. This information can be used by the synthesizer in prediction.

7.1. Experiments and Results

We repeated our experiments on Hindi, Dari, and English with the inferred phonetic information. For our baseline, we take the best performing system from the previous set of experiments. For Hindi, this would be the IP system with 165 phones. For Dari, the baseline would be the same as the previous one. For English, it would again be the system with perfect phonetic transcriptions. Figure 3 shows the results in Hindi and Dari. In both languages, the IP system is able to perform better than the baseline systems.

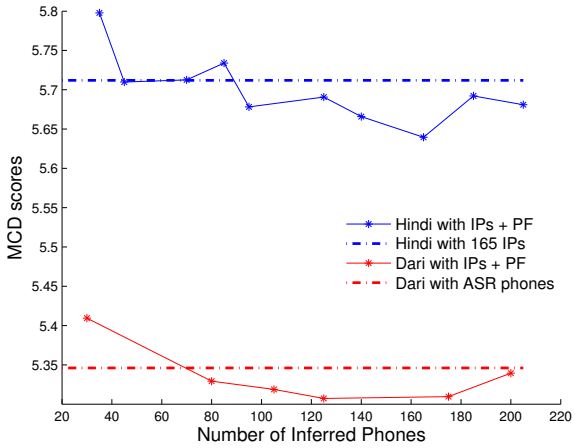

Fig. 3. IPs with phonetic information added

In the English system, the baseline MCD was 5.702. After adding inferred phonetic information, the best IP system, with 190 phones, had an MCD of 5.812. This brings our system even closer to the best possible phonetic transcriptions.

8. DISCUSSION

In this paper we have provided a technique that can be used to automatically derive phones and phonetic features for those phones. One thing that needs consideration is that the phones were derived purely from acoustics. The improvements that we get in synthesis may be attributed to the transition to a representation that is closer to language as it is spoken as opposed to the way it is written. While the advantages of this direction are obvious, it is important to note that the gains will come at a cost. A Text-To-Speech synthesis system or synthesis in a Speech to Speech translation system must work from text rather than raw phones. At this point, the ease with which the inferred phones can be predicted from text is unknown. This is an issue that we plan to investigate in future work.

9. REFERENCES

[1] Sunayana Sitaram, Sukhada Palkar, Yun-Nung Chen, Alok Parlikar, and Alan W Black, "Bootstrapping text-to-speech for speech processing in languages without an orthography," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7992–7996.

[2] International Phonetic Association, *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*, Cambridge University Press, 1999.

[3] Sebastian Stüker, Tanja Schultz, Florian Metze, and Alex Waibel, "Multilingual articulatory features," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*. IEEE, 2003, vol. 1, pp. I–144.

[4] Tanja Schultz, "Globalphone: a multilingual speech and text database developed at Karlsruhe university.," in *INTERSPEECH*, 2002.

[5] Bajibabu Bollepalli, Alan W Black, and Kishore Prahallad, "Modelling a noisy-channel for voice conversion using articulatory features.," in *INTERSPEECH*, 2012.

[6] J. Makhoul, "Linear prediction: A tutorial review," *Proceedings of the IEEE*, vol. 63, no. 4, pp. 561–580, 1975.

[7] L. Rabiner and B.H. Juang, *Fundamentals of speech recognition*, Prentice hall, 1993.

[8] Florian Metze, *Articulatory features for conversational speech recognition.*, Ph.D. thesis, Karlsruhe Institute of Technology, 2005.

[9] Alan W Black, H Timothy Bunnell, Ying Dou, Prasanna Kumar Muthukumar, Florian Metze, Daniel Perry, Tim Polzehl, Kishore Prahallad, Stefan Steidl, and Callie Vaughn, "Articulatory features for expressive speech synthesis," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4005–4008.

[10] Katrin Kirchhoff, *Robust speech recognition using articulatory information*, Ph.D. thesis, Bielefeld University, 1999.

[11] Katrin Kirchhoff, Gernot A Fink, and Gerhard Sagerer, "Combining acoustic and articulatory feature information for robust speech recognition," *Speech Communication*, vol. 37, no. 3, pp. 303–319, 2002.

[12] Alan Wrench, "The MOCHA-TIMIT articulatory database," *Univ. of Edinburgh. Edinburgh, UK, Nov*, 1999.

[13] John Garofalo, David Graff, Doug Paul, and David Pallett, "CSR-I (WSJ0) complete, LDC93S6A," .

[14] Keiichi Tokuda, Takao Kobayashi, Takashi Masuko, and Satoshi Imai, "Mel-generalized cepstral analysis-a unified approach to speech spectral estimation.," in *ICSLP*, 1994, vol. 94, pp. 18–22.

[15] David Johnson et al., "ICSI Quicknet software package," 2004.

[16] Kishore Prahallad, Alan W Black, and Ravishankhar Mosur, "Sub-phonetic modeling for capturing pronunciation variations for conversational speech synthesis," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*. IEEE, 2006, vol. 1, pp. I–I.

[17] John Kominek and Alan W Black, "The CMU Arctic speech databases," in *Fifth ISCA Workshop on Speech Synthesis*, 2004.

[18] Alan W Black, "ClusterGen: A statistical parametric synthesizer using trajectory modeling," in *Proceedings of INTERSPEECH*, 2006, pp. 1762–1765.

[19] Kishore Prahallad, Naresh Kumar, Venkatesh Keri, S Rajendran, and Alan W Black, "The IIIT-H indic speech databases.," in *INTERSPEECH*, 2012.

[20] CMU Sphinx, "CMU Sphinx open source speech recognizer," http://cmusphinx.sourceforge.net/.

[21] Sunayana Sitaram, Gopala Krishna Anumanchipalli, Justin Chiu, Alok Parlikar, and Alan W Black, "Text to speech in new languages without a standardized orthography," in *Proceedings of 8th Speech Synthesis Workshop, Barcelona*, 2013.

[22] Leo Breiman, Jerome H Friedman, Richard A Olshen, and Charles J Stone, *Classification And Regression Trees*, CRC press, 1993.

[23] R Weide, "The Carnegie Mellon pronounciation dictionary [cmudict. 0.6]," 2005.