PARSING SPEECH INTO ARTICULATORY EVENTS

Kadri Hacioglu, Bryan Pellom and Wayne Ward

Center for Spoken Language Research
University of Colorado at Boulder
E-mail: {hacioglu,pellom,whw}@cslr.colorado.edu

ABSTRACT

In this paper, the states in the speech production process are defined by a number of categorical articulatory features. We describe a detector that outputs a stream (sequence of classes) for each articulatory feature given the Mel frequency cepstral coefficient (MFCC) representation of the input speech. The detector consists of a bank of recurrent neural network (RNN) classifiers, a variable depth lattice generator and Viterbi decoder. A bank of classifiers has been previously used for articulatory feature detection by many researchers. We extend their work first by creating variable depth lattices for each feature and then by combining them into *product lattices* for rescoring using the Viterbi algorithm. During the rescoring we incorporate language and duration constraints along with the posterior probabilities of classes provided by the RNN classifiers. We present our results for the *place* and *manner* features using TIMIT data, and compare the results to a baseline system. We report performance improvements both at the frame and segment levels.

1. INTRODUCTION

The linear symbolic representation of speech at the lowest symbolic level using phonemes is very common in state-of-the-art speech recognizers. This is known as the "beads-on-a-string" representation. The drawbacks of this representation have been reported in [1-3]. A natural extension of this representation is "beads-on-multiple-strings" which suggests a nonlinear multi-dimensional symbolic representation. In the latter, the first issue is the decision on the nature of features ("strings") and the set of classes ("beads") for each feature. The second issue is the accurate detection of the classes along each dimension. Many different symbolic feature representations and ways of detecting the feature classes have been reported in [4-14]. According to the "beads-on-multiple-strings" approach, a segment of speech is classified into a number of broad classes in multiple dimensions. We associate the dimensions with articulatory features and the classes with their values. In doing so, a representation of the speech frame, or segment, is obtained as an articulatory feature vector (or state). In turn, a word can be represented by a sequence of feature vectors. Our goal is the accurate detection of the feature streams from the input speech for subsequent word recognition.

The speech recognition framework on which our current work is based is in strong agreement with the detection-based framework for speech recognition and understanding presented in [17]. Also, the lattice rescoring strongly overlaps with the notion of event lat-
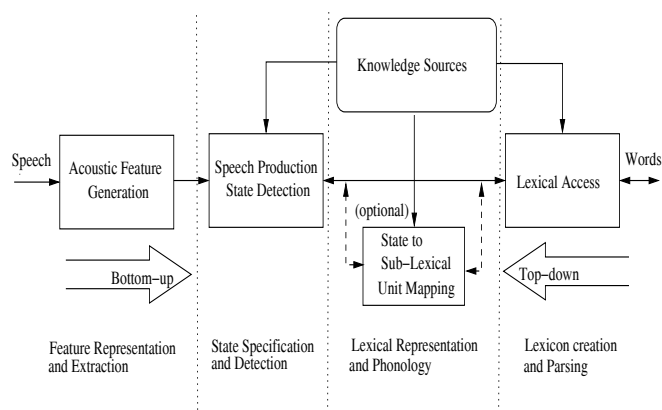

Fig. 1. Speech recognition framework

tices advocated in [18, 19]. The framework is illustrated in Figure 1. It points to four research areas. These are (i) acoustic feature representation and extraction, (ii) state specification and detection, (iii) lexical representation and phonology and (iv) lexicon creation and parsing. We currently work on the state specification and detection problem.

We propose a detector which is a bank of recurrent neural networks (RNNs) followed by a *product lattice* rescoring unit. The outputs of RNNs are the posterior probabilities of classes for each articulatory feature given the acoustic representation in MFCCs. RNNs have been extensively used for the articulatory feature detection in [7, 14]. We extend their work by generating lattices of feature classes for each feature stream. These lattices can be either independently or jointly rescored for better performance. The posterior probabilities along with language and duration constraints within and across the feature streams are used during rescoring. The final output of the system is a sequence of classes at frame level for each feature stream (tagged with posterior probabilities as indicators of the reliability of the information/evidence passed to higher levels). Articulatory events, or segments of speech, are created by concatenating the frame level repeating classes. We present results that show improved performance.

The paper is organized as follows. In Section 2, we present the framework for our ongoing research toward a system that uses articulatory features in speech recognition. We discuss the articulatory feature representation that we are currently considering and an implementation of a detector for it. Experimental results are presented in Section 3. Conclusions are made in the final section.

Table 1. Feature system based on articulatory phonetics

Feature	Categories
Phonation	+voice, -voice
Manner	approximant, fricative, nasal, stop, vowel
Place	labial, labiodental, dental, alveolar, velar, glottal, high, mid, low
FrontBack	front, back
Rounding	+round, -round

2. ARTICULATORY FEATURE BASED APPROACH

2.1. Recognition Framework

In this section we describe a framework for our ongoing research toward a speech recognition system based on articulatory features. The framework is illustrated in Figure 1. In the standard top-down approach, the lexicon is accessed to hypothesize words with a number of different pronunciations; each word is a sequence of selected sub-lexical units. Then the sub-lexical units are associated with their respective "states" (or acoustic models). A score is obtained depending on the degree of match between the acoustic observations and the acoustic models in that state. In the bottom-up approach, the input speech is segmented into a sequence of "states" (or constituents) and then combined into words using phonotactics directly on the states or on another (optional) intermediate representation (like phonemes or syllables). A hybrid approach is also possible; one can perform a top-down inference with constraints induced in a bottom-up manner. That is, the speech states detected in a bottom-up manner can guide the top-down search for words. In this paper, we focus on the state specification and detection problem illustrated in Figure 1.

2.2. A Multi-Valued Multi-Stream State Representation

We define the state of speech sound production process by using categorical features as

$$s_t = [f_1 f_2 \cdots f_L] \tag{1}$$

where s_t is the state at the t-th frame, f_i is the i-th feature, with M_i categories, and L is the number of features. We are considering an articulatory feature representation for Equation (1).

The dimensions and their categorical values are determined by articulatory phonetics (AP). In AP, we are mainly concerned with how people make the speech sounds of language using human vocal organs (or articulators). Some examples of the articulators are vocal cords, velum, tongue, teeth and lips. Speech sounds can be described in several dimensions; namely, phonation, manner of articulation, place of articulation, front/back and rounding. The speech sound phonation depends on the state of the vocal cords and shape of the oral tract. It can be voiced,unvoiced or mixed. The place of articulation is the place where the articulators obstruct the air stream. The manner of articulation refers to the way the articulation is accomplished. For example, the velum couples the nasal tract to the oral tract for nasal sounds. The speech sound characteristics also strongly depend on the shape and the position
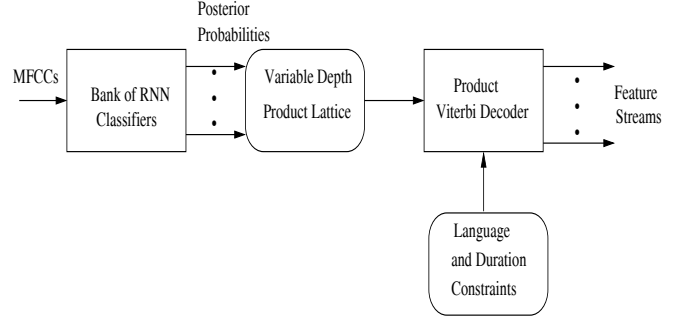
Fig. 2. An implementation of the second stage in Figure 1.

of the tongue. The tongue can be raised, and moved forward and backward, and can be made to touch a place in the oral tract or to change the volume of the oral tract for creating different sounds. Therefore, the shape and position of the tongue, and its place of articulation can be used to categorize speech sounds. For example, the consonants can be described by their *place* and *manner* of articulation, and their phonation. On the other hand, the vowels can be described by their height, backness and rounding. The rounding feature describes whether the sounds are generated when the lips are rounded or unrounded. In this paper, we use the feature system shown in Table 1 [14].

2.3. Detector Implementation

In this section we describe the implementation of the speech production state detector module shown in Figure 1. A more detailed block diagram of the detector is exhibited in Figure 2. Recall that the state was defined in terms of the articulatory features. The state detector includes a bank of RNN classifiers, one for each feature. We used the NICO toolkit to train RNNs [20, 21]. The choice of RNNs is motivated by (i) their ability to implicitly incorporate context in detection by recurrent connections, (ii) their decent performance as demonstrated in [5, 14] and (iii) the fact that the outputs can be interpreted as the posterior probabilities of classes. Although their training time is very slow as compared to hidden markov models, our pilot experiments have shown that their training is very fast when compared to support vector training while using all TIMIT data. The outputs of the RNNs are used to create variable depth lattices. That is, for each feature a variable number of class assignments with their scores are retained at each frame depending on an empirically optimized threshold on the posterior probabilities. To improve the performance of the feature detection we rescore lattices using

- the posterior probabilities from the RNNs
- durational models
- n-gram feature models

Assuming that the features are independent, we rescore them independently. However, this is not a realistic assumption, and we might get some improvement by rescoring them jointly. We do this by rescoring a cartesian product of the lattices (or product lattice) taking into account the durational and language constraints on
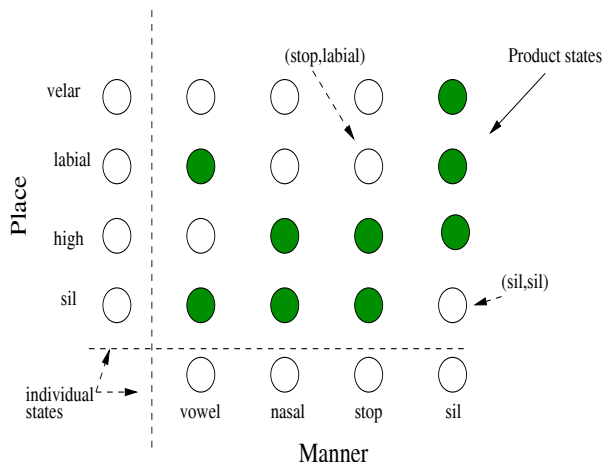
Fig. 3. A slice of the product lattice at a certain frame. Here, the individual streams have 4 classes retained. The total number of product states is 16. However, some of them are restricted depending on the degree of asynchrony allowed.

product states. A time slice of the product lattice (of degree 2) is illustrated in Figure 3 for the *manner* and *place* features. We have the additional *sil* class for both streams. At each frame, the number of possible product states is equal to the product of the number of classes retained for each feature at that frame. However, one can exclude certain product states by limiting the asynchrony between the streams. For example, the *sil* class in one stream can only co-occur with the *sil* class in another stream. In the figure, the states that are not allowed to co-occur are illustrated by the dark circles. A simple extension of the Viterbi algorithm for a single stream is used to search for the best path through the product lattice, without explicitly creating the product lattice. The posterior probability of the product state is assumed to be the multiplication of the posterior probabilities of the component states.

The output of the lattice rescoring is a sequence of classes at a frame level for each stream. These frame level sequences are then converted into segments by collapsing the sequence of identical classes. Both outputs are tagged with posterior probabilities to indicate the confidence on the detected event for subsequent processing.

3. EXPERIMENTS

3.1. Data

TIMIT data was used in all experiments. It is a very high quality corpus labeled at phoneme and word levels. All but SA files were used for training and testing. We reserved a randomly selected set of 100 sentences out of 3696 training sentences as a validation set. Experimental results were reported on the core test data containing 1344 sentences. The mapping from phonemes to features were made using the table presented in [7]. The full set of the TIMIT phonemes were used in the experiments.

Table 2. Baseline System Results

	Manner	Place
Frame Error	15.5%	28.4%
Segment Error	35.7%	57.1%

Table 3. Independent lattice rescoring results

	Manner	Place
Frame Error	15.0%	27.5%
Segment Error	22.4%	32.5%

3.2. Baseline System

The speech waveforms were parameterized using MFCCs. The length of the analysis frame was set to 25ms. The analysis frames were overlapped and shifted by 10ms. Each frame is represented by 12 MFCCs and energy plus velocity and acceleration coefficients. A context of five frames centered at the current frame was used. This amounts to a 195-dimensional input vector. For each feature an RNN is trained with one hidden layer consisting of 200 hidden units with recurrent connections. The number of training iterations was set to 60. We did not carry out any extensive optimization of the learning parameters, architecture and context using the validation data. So, we believe that there is still some room for improvement by optimizing the RNN training. The 1-best outputs were obtained by picking the maximum posterior probability at the output of the RNNs for each feature. The standard NIST Sclite scoring tool which counts insertions, deletions and substitutions is used to compute the segment error. The baseline results are shown in Table 2 for the *manner* and *place* features.

3.3. Independent Lattice Rescoring

For each feature a lattice with a varying depth was created by posterior probability pruning. All outputs above a validation set optimized threshold were retained (0.7-0.8). In cases where there were no outputs above the threshold (relatively high entropy frames) all the outputs were retained in the lattice. Each lattice was rescored independently using their own bigram feature model (trained using the sequence of classes in TIMIT training data) and gamma distributed duration model using the Viterbi algorithm. The weights for the log-linear combination of the knowledge sources were optimized using the validation set. The results are exhibited in Table 3. For the *manner* feature, the error reduction is 3.2% relative at the frame level and 37% at the segment level. For the *place* feature the improvement is 3.2% relative at the frame level and 43% relative at the segment level.

3.4. Product Lattice Rescoring

Here we combined the two lattices in Section 3.3 into a product lattice and utilized bigram and durational constraints on the product states for rescoring. In the product lattice, the states (or nodes) are all the possible pairs of *manner* and *place* features allowed by the

Table 4. Product lattice rescoring results.

	Manner	Place
Frame Error	13.5%	27.0%
Segment Error	19.6%	31.4%

asynchrony constraints as illustrated in Figure 3. The results are shown in Table 4. The further improvement for the *manner* feature is 10.6% relative at the frame level and 12.5% relative at the segment level. The respective further improvements for the *place* feature are 2.2% and 3.4%, respectively.

4. CONCLUSIONS

We have presented some experiments towards parsing speech into articulatory-feature based events. The speech-production state has been categorized using a number of multi-valued articulatory features. We have addressed the state detection problem and explored some lattice rescoring methods to improve the performance of the baseline detector. We have shown the impact of feature-level language and duration constraints on the performance. We have also shown that the lattice rescoring using inter-feature constraints through a *product lattice* yields further improvement.

5. REFERENCES

[1] M.A. Huckvale, "Exploiting speech knowledge in neural nets for recognition," *IEEE Transactions on Pattern Analysis and Machine Inteligence*, vol. 9, no. 1, pp. 1–14, 1990.

[2] M. Ostendorf, "Moving beyond the 'beads-on-a-string' model of speech," *Proc. of IEEE ASRU Workshop*, pp. 79–84, Keystone, Colorado,1999.

[3] Li Deng, "Switching dynamic system models for speech articulation and acoustics," *M. Johnson, M. Ostendorf, S. Khudanpur, and R. Rosenfeld (eds.) IMA Volume 138: Mathematical Foundations of Speech and Language Processing*, pp. 115–134, 2003.

[4] M.A. Huckvale, "Phonetic characterization and lexical access in non-segmental speech recognition," *Proc. 13th International Congress of Phonetic Sciences*, vol. 4, pp. 280–283, August 1995.

[5] S. King, T. Stephenson, S. Isard, P. Taylor, and A. Strachan, "Speech recognition via phonetically featured syllables," *5th International Conference on Spoken Language Processing*, pp. 1013–1017, Sydney, Australia, 1998.

[6] K. Kirchhoff, *Robust Speech Recognition Using Articulatory Information*, Ph.D. thesis, University of Bielefeld, 1999.

[7] S. King and P. Taylor, "Detection of phonological features in continuous speech using neural networks," *Computer Speech and Language*, vol. 14, no. 4, pp. 333–353, 2000.

[8] E. Eide, "Distinctive features for use in an automatic speech recognition system," in *Proceedings of the 7th EUROSPEECH*, Aalborg, Denmark, September 2001.

[9] C A. Juneja and Espy-Wilson, "Segmentation of continuous speech using acoustic phonetic parameters and statistical learning," in *Proceedings of the ICNIP*, 2002.

[10] S. Chang, S. Greenberg, and M. Wester, "An elitist approach to articulatory-acoustic feature classification," *Proceedings of the 7th EUROSPEECH*, pp. 1729–1733, September Aalborg, Denmark, 2001.

[11] F. Metze and A. Waibel, "A flexible stream architecture for asr using articulatory features," in *Proceedings of the 7th ICSLP*, Denver, Colorado, September 2002.

[12] B. Launay, O. Siohan, A.C. Surendran, and C.-H. Lee, "Towards knowledge based features for HMM based large vocabulary automatic speech recognition," in *International Conference of Acoustics, Speech, and Signal Processing*, Orlando, Florida, 2002.

[13] S. Stuker, T. Schultz, F. Metze, and A. Waibel, "Multilingual articulatory features," in *International Conference of Acoustics, Speech, and Signal Processing*, Hong Kong, April 2003.

[14] M. Wester, "Syllable classification using articulatory acoustic features," in *Proceedings of the 8th EUROSPEECH*, Geneva, Switzerland, September 2003.

[15] M. Tang, S. Seneff, and V.W. Zue, "Modeling linguistic features in speech recognition," *Proceedings of the 8th EUROSPEECH*, pp. 2585–2588, September Geneva, Switzerland,2003.

[16] Tarek-Abu Amer and Julie Carson-Berndsen, "Hartfex: A multi-dimensional system of HMM based recognizers for articulatory feature extraction," in *Proceedings of the 8th EUROSPEECH*, Geneva, Switzerland, September 2003.

[17] B.H. Juang, "Detection based processing for speech recognition and understanding," in *NSF Symposium on Next Generation ASR*, Atlanta, GA, October 2003.

[18] Kai Hubener and Julie Carson-Berndsen, "Phoneme recognition using acoustic events," in *Proceedings of the International Conference on Spoken Language Processing*, Yokohoma, Japan, September 1994.

[19] C.-H. Lee, "On automatic speech recognition at the dawn of the 21st century," *IEICE Transactions on Information and Systems*, vol. E86-D, no. 3, pp. 377–396, March 2003.

[20] N. Strom, "Phoneme probability estimation with dynamic sparsely connected artificial networks," in *The Free Speech Journal*, Issue No. 5, 1997.

[21] N. Strom, "The NICO toolkit for artificial neural networks," in *http://www.speech.kth.se/NICO*, 1996.