USE OF HIGH-LEVEL LINGUISTIC CONSTRAINTS FOR CONSTRUCTING FEATURE-BASED PHONOLOGICAL MODEL IN SPEECH RECOGNITION

Jiping Sun and Li Deng

Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

ABSTRACT

Modeling phonological units of speech is a critical issue in speech recognition. In this paper, we report our recent development of an overlapping feature-based phonological model which gives long-span contextual dependency. We extend our earlier work by incorporating high-level linguistic constaints in automatic construction of the feature overlapping patterns. The main linguistic information explored includes morpheme, syllable, syllable constituent categories and word stress markers. We describe a consistent computational framework developed for the construction of the feature-based model, and discuss use of the model as the HMM state topology for speech recognizers.

1. INTRODUCTION

Modeling phonological units of speech, also referred to as pronunciation or lexicon modeling, is a critical issue in automatic speech recognition. Over the past several years, we have been studying this issue from computation phonology perspectives, motivated by some recent versions of nonlinear phonology theory [2, 6]. The computational framework developed is based on sub-phonemic, overlapping articulatory features where the rule-governed overlapping pattern can be described mathematically as a finite-state automaton and each state in the automaton corresponds to a feature bundle with relative timing information specified [4, 5]. In this paper, we report our new development on the feature-based phonological model in incorporating high-level linguistic (mainly prosodic) constraints for automatic construction of the feature overlapping patterns aimed at automatic speech recognition.

In our feature-based phonological model used for speech recognition, feature overlapping patterns are converted to an HMM state-transition network. Each state is a (overlapped) feature bundle and represents a unique, symbolically-coded articulatory configuration responsible for producing speech acoustics from that state. When the features of adjacent segments (phonemes) overlap asynchronously in time, new states are derived which model either the transitional phases between the segments or the allophonic alternations caused by the influence of context. Since feature overlapping is not restricted to immediate neighboring segments, this approach is expected to show advantage over the conventional context dependent modeling based on diphones or triphones which limits the context influence to only immediately close neighbors.

In our previous work, the feature overlapping patterns were constructed based only on the information about the phoneme (i.e., segment) identity in each utterance to be modeled. It is apparent that a wealth of linguistic factors beyond the level of phoneme, such as prosodic information (syllable, morpheme, stress, utterance boundaries, etc.), directly control the low-level feature overlaps. Thus, it is desirable to use such high-level linguistic information to control and to constrain the feature overlaps effectively. As an example, in pronouncing word *display*, the generally unaspirated /p/ is constrained by the condition that a /s/ precedes it in the same syllable onset. On the other hand, in pronouncing word *displace*, *dis* is lexically constrained as a morpheme of one syllable and /p/ in the initial position of the next syllable subsequently tends to be aspirated.

In order to systematically exploit the high-level linguistic information for constructing overlapping feature-based phonological model in speech recognition, we need to develop a computational framework and methodology in a principled way. Such a methodology must be sufficicently comprehensive to cover a wide variety of utterances (including spontaneous speech) dealt with in speech recognition. This forms the focus of the research reported in this paper.

2. USE OF LINGUISTIC CONSTRAINTS

The high-level linguistic/prosodic information to be used for constraining feature overlapping patterns includes utterance, word, morpheme, syllable, syllable constituent categories, and word stress markers. All such information is first obtained by a recursive transition network-based morphological parser [3], together with a pronunciation dictionary. The parser output in the form of constituent tree is further transformed into subsegmental feature structures [1, 4] while keeping the high-level information. As an example, this process is illustrated by parsing a one-word utterance: *display*. The parse tree in Fig. 1 denotes that word *display* consists of 2 syllables. The category 'medcv' is used for dealing with multiple syllables recursively; 'medc' and 'medc3-1' are categories of syllable medial consonant clusters. The category 'init' denotes word initial syllable onset. A square bracket specifies a segment type, such as c (consonant), stp (stop), fri (fricative), etc., providing phonotactic information for separating syllable medial consonant clusters into coda and onset. This tree structure is transformed into a subsegmental feature structure illustrated in Fig. 2.

The subsegmental feature structure can be viewed as an autosegmental structure [1, 6] with skeletal and articulatory-feature tiers and a prosodic structure placed on top of it.

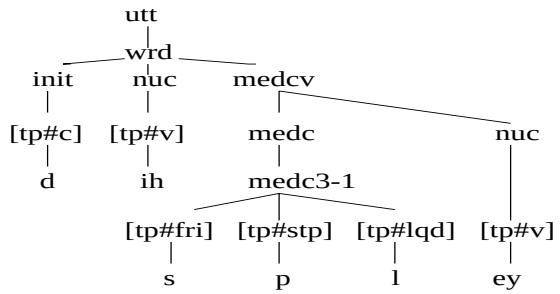

Figure 1: Parse tree for word *display*

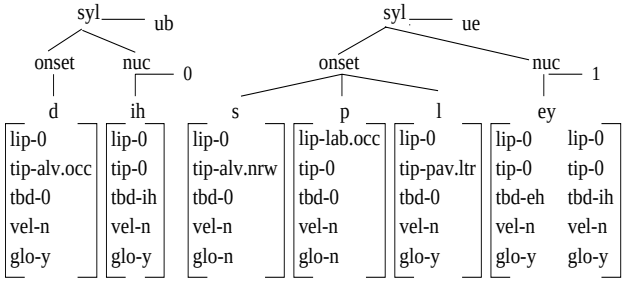

Figure 2: Subsegmental feature structure for *display*

Overlapping is realized by computationally implemented phonological rules, incorporating high-level information. We define a temporal feature logic [1] as the metalanguage for the formulation of phonological rules.

A Temporal Feature Logic

A temporal feature logic for constraint-based approach to feature overlapping is a language $\mathcal{L}(\mathcal{X}, \mathcal{P}, \mathcal{T}, \mathcal{C})$ where

$\mathcal{X}$ is a set of variables: $a, b, c, x, y, z...$, etc..

$\mathcal{P}$ is a prosodic structure: $\{syl, sylconst, seg, boundary, stress\}$.

$\mathcal{T}$ is a tier structure: $\{seg, articulator, feature\}$.

$\mathcal{C}$ is a set of logical connectors: $\{\delta, \prec, \circ, \bowtie, =, \neg, \vee, \wedge, \forall, \exists, \rightarrow, \equiv, (,), \top, \bot\}$, where δ, $\prec$, $\circ$, and $\bowtie$ are "dominance", "precedence", "overlap", and "mix", respectively, etc..

The prosodic structure:

1. $\forall xy,\ syl(x) \wedge x\,\delta\,y \rightarrow sylconst(y) \vee boundary(y)$
   Syllable dominates syllable constituent or boundary.
2. $\forall xy,\ sylconst(x) \wedge x\,\delta\,y \rightarrow seg(y) \vee stress(y)$
   Syllable constituent dominates segment or stress.
3. $\forall x, boundary(x) \rightarrow x \in \{ub, ue, wb, we, mb, me\}$
   ub = utterance beginning
   ue = utterance end
   wb = word beginning
   we = word end
   mb = morpheme beginning
   me = morpheme end
4. $\forall x,\ sylconst(x) \rightarrow x \in \{onset, nuc, coda\}$
5. $\forall x,\ stress(x) \rightarrow x \in \{0, 1\}$

The tier structure:

1. $\forall x,\ seg(x) \rightarrow \exists y,\ x\,\delta\,y \wedge articulator(y)$
2. $\forall x,\ articulator(x) \rightarrow \exists y,\ x\,\delta\,y \wedge feature(y)$
3. $\forall x,\ articulator(x) \rightarrow x \in \{lip, tip, tbd, vel, glo\}$
4. $\forall x,\ feature(x) \rightarrow poa(x) \vee cdg(x) \vee shape(x)$ where
   poa = place of articulation
   cdg = constriction degree
   $shape$ = shape of lip, etc.

Fig. 2 shows how prosodic and tier structures are motivated by subsegmental feature structures.

Dominance, Precedence, Overlap, and Mix

The basic properties of $\delta, \prec, \circ$ are described in [1]. Some of the important ones are:

$\forall xy,\ x \prec y \rightarrow \neg\, x \circ y$ (mutual exclusion of δ and $\prec$)
$\forall wxyz,\ w \prec x \wedge x \circ y \wedge y \prec z \rightarrow w \prec z$
(transitivity of $\prec$ through $\circ$)
$\forall xy,\ x\,\delta\,y \rightarrow x \circ y$ (locality constraint)

The first two properties together form the "no-crossing constraint" [1]. We abandon the "linearity constraint" which requires events of the same sort be in precedence relation only. Instead, we define two dominance relations so that sister events are in either precedence or overlap relations, regardless of sort:

$\forall xy,\ x\,\delta\,y \equiv x\,\delta_{\prec_n}\,y \vee x\,\delta_{\circ_n}\,y\ (n \in N)$

and add two properties:

$\forall ax,\ a\,\delta_{\prec_n}\,x \rightarrow \exists y,\ a\,\delta_{\prec_n}\,y \wedge (x \prec y \vee y \prec x)$
x, y are in the same precedence group n.
$\forall ax,\ a\,\delta_{\circ_n}\,x \rightarrow \exists y,\ a\,\delta_{\circ_n}\,y \wedge x \circ y$
x, y are in the same overlap group n.

The use of the subscript n for the modified dominance relations helps to avoid violation of the "no-crossing constraint" by identifying groups of dominated events. The advantages of this modified dominance relation are

- Overlapping tiers dominated by a segment can be of the same (abstract) sort; and
- Events of the same sort and on the same tier are allowed to overlap, which defines the mix relation $\bowtie$:

$\forall axy, tier_i(x) \wedge tier_i(y) \wedge a\,\delta_{\circ_j}\,x \wedge a\,\delta_{\circ_j}\,y \rightarrow x \bowtie y.$

This means if events on the same tier overlap in time, they are said to mix with each other. As overlap can be either partial or complete, so is the mix relation. This mix relation is used to describe coarticulation involving the same dimension in the articulatory feature space (i.e., *co-production*).

3. IMPLEMENTATION

3.1. Phonological-Rule Formulation

Given the computational framework based on temporal feature logic outlined above, the phonological rules have been formulated systematically based on subsegmental feature bundles, with the high-level linguistic information providing constraint and control. An example is shown in Fig. 2, where each feature bundle consists of a 5-tuple component features, which are dominated by a segment and are initialized as simultaneous to each other (i.e., being synchronous "canonically"). The subsequent phonological rules will produce asynchrony among the component features using the constraints provided by the high-level linguistic information.

In our formulation of phonological rules, broad categories of segments are used instead of the segments themselves.

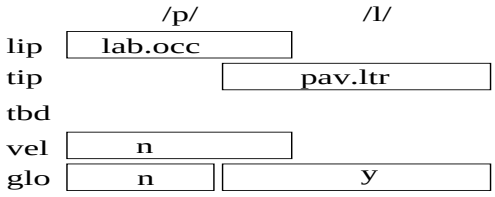

Figure 3: /p/ affected by overlapping type usp_cc2

These categories (for American English) are: 1) vowels (**v**); 2) voiced stops (**stp1**); 3) unvoiced stops (**stp2**); 4) voiced fricatives (**fri1**); 5) unvoiced fricatives (**fri2**); 6) nasals (**nas**); 7) onset liquids (**lqd1**); 8) coda liquids (**lqd2**); 9) glides (**gld**); 10) affricates (**afr**); and 11) aspiration (**h**). All phonological rules are formulated on the basis of *feature overlapping pattern*, and most of the rules have their application domain of a demi-syllable. As an initial experiment, we have parsed some 6000 words in the TIMIT dictionary and obtained about 300 "demi-syllable patterns", i.e. demi-syllables of segment categories. For each of these demi-syllable patterns, a feature overlapping pattern is described, which forms the main body and substance of a phonological "rule". As an example, the overlapping pattern that matches the syllable /s p l ey/ looks like:

```
[[we,onset,nuc,1],
  fri2 -> [cc1,cc2,cc3],
  stp2 -> [cc2,lpr,usp],
  lqd1 -> [cv, lpr,nas],
  v    -> [vc1,vc2,vc3,nas]]
```

The rule is in a list representation. The first sublist contains high-level structure information. The remaining of the list maps each segment category to the names of some *overlapping types*. An overlapping type describes what happens when a feature bundle is overlapped at some tier(s) by features of other segments in context. For example, "cc1" is the name for the overlapping type: *"anticipatory tongue tip feature overlapping in consonant cluster"*.

The overlapping types are described in a database of segments, in which each segment in English is given a broad category, an articulatory feature bundle, and a number of descriptions of overlapping types applicable to the segment in verious contexts. For example, /b/ and /p/ are described as:

```
segment(b, [tp#stp],
           [lip_occ,0,0,n,y],
            [cv,l#3,[0.25]],
            [lpr,m#1#rnd,[0.5,1]],
            [cc1,m#1,[0.25]],
            [cc2,l#2,[0.25]],  and
segment(p, [tp#stp,
           [lip_occ,0,0,n,n],
            [cv,l#3,[0.25]],
            [lpr,m#1#rnd,[0.5,1]],
            [cc1,m#1,[0.25]],
            [cc2,l#2,[0.25]],
            [usp,l#5#y,[0.25]],
            [usp_cc2,l#5#y,l#2,[0.25#0.25]],
```

which are associated with four and six distinct overlapping types, respectively. Each overlapping type description consists of: 1) a name; 2) one or more operators; and 3) corresponding temporal indicator(s). The name of an overlapping type is used by *overlapping pattern(s)* described earlier to identify the needed operator(s). An operator contains: a) an action indicator; b) a tier indicator; and c) an optional value indicator. They are separated by the symbol '#'. (The tier indicator specifies at which tier the action takes place; e.g., '1' for Lip tier, '2' for Tongue-Tip tier, etc., and a value indicator limits the value of the overlapping features). A temporal indicator specifies the percentage of overlapping. Some overlapping-type names and action indicators are explained below:

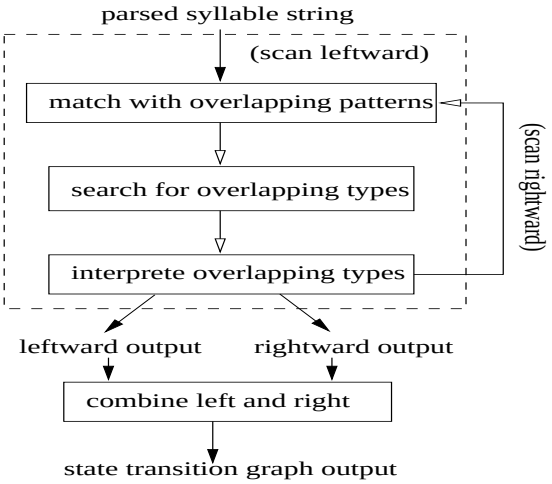

Figure 4: The overlapping feature generator

- Overlapping type name
  cv: consonant-vowel partial coarticulation
  lpr: lip-rounding
  cc1: consonant-consonant coarticulation on lip tier
  cc2: consonant-consonant coarticulation on tip tier
  usp: unaspirated voiceless stop
  usp_cc2: composition of usp and cc2
- Operator action indicator
  l: leftward overlap by feature from right context
  m: leftward mix with feature from right context

Fig. 3 illustrates the effect of overlapping type "usp_cc2" acting on /p/ followed by /l/.

3.2. Overlapping-Feature Generator

An overlapping feature generator is a program which 1) scans the input string of feature bundles with high-level linguistic information; 2) matches (assigns) them to appropriate overlapping patterns; 3) executes overlapping operations by interpreting the overlapping types specified in the patterns; and 4) integrates the results of 3) to produce a state-transition network (i.e., HMM state topology). A block diagram of the overlapping feature generator is shown in Fig. 4.

In the program implementing the feature generator, the following constraints are used to limit the solution space:

$$\forall xy,\ seg(x) \wedge seg(y) \wedge x \prec y \wedge operator(x, l\#n) \rightarrow feature(y, n, f) \wedge \neg f = 0$$

$$\forall xy,\ seg(x) \wedge seg(y) \wedge x \prec y \wedge operator(x, l\#n\#f1) \rightarrow feature(y, n, f2) \wedge is.a(f2, f1).$$

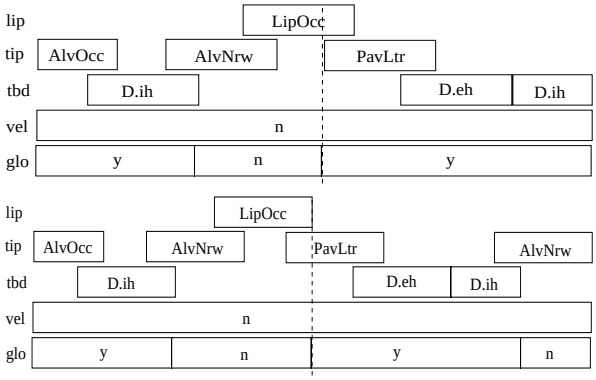

Figure 5: Feature overlaps for words *display* (upper) and *displace* (lower)

For rightward scanning, the rightward overlapping operators and the corresponding constraints are used. At the final stage of the generator, the constraint for integrating the leftward and rightward features with temporal overlap is expressed by

$$\forall xy, l.output(x) \wedge r.output(y) \rightarrow x \circ y \in f.output.$$

We have developed this overlapping-feature generator in Prolog, and are using the output of the generator as the HMM topology in speech recognition.

4. ILLUSTRATIONS

In this section we will illustrate by examples some results in constructing the overlapping feature-based phonological model using the implementation described in Section 3. The first example is for words *display* and *displace*. After the stages of syllable parsing, overlapping-pattern matching, and overlapping-type searching/interpretation, the results of the feature overlaps for these two words are shown in Fig. 5. Importantly, due to the different syllable constituent assignment of /s/ (/d ih s . p l ey s/ vs. /d ih . s p l ey/) distinct overlapping types in separate overlapping patterns are specified for the two words. This gives rise to the correct phonological process that the /p/ in *displace* tends to be aspirated and the /p/ in *display* becomes largely unaspirated.

The second example comes from word *strong*, which contains several optional feature overlaps over a variable temporal extent involving lip-rounding and nasalization. Such variability gives rise to a complex network (i.e., state-transition graph) as output of the feature generator. This network is shown in Fig. 6, where each state (numbered numerically) contains a distinct set of symbolic features. These symbols, together with the related phonetic notations, associated with each state, are given below:

```
[1]  [0,AlvNrw,0,n,n]          /s/
[2]  [Rnd4,AlvNrw,0,n,n]       /s(rnd)/
[3]  [0,AlvNrwOcc,0,n,n]       /s-t/
[4]  [0,AlvOcc,0,n,n]          /t/
[5]  [Rnd4,AlvNrwOcc,0,n,n]    /s(rnd)-t/
[6]  [Rnd4,AlvOcc,0,n,n]       /t(rnd)/
[7]  [Rnd4,AlvOccNrw,0,n,y]    /t-r/
[8]  [Rnd4,AlvNrw,0,n,y]       /r/
[9]  [Rnd4,AlvNrw,0,y,y]       /r(nas)/
```

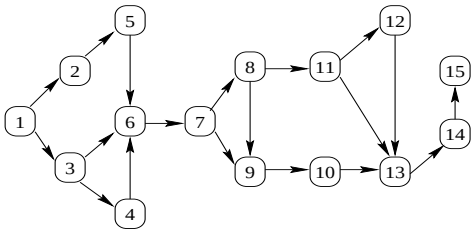

Figure 6: State-transitional graph for word *strong*

```
[10]  [Rnd4,AlvNrw,D.ao,y,y]  /r-ao(nas)/
[11]  [Rnd4,AlvNrw,D.ao,n,y]  /r-ao/
[12]  [Rnd6,0,D.ao,n,y]       /ao/
[13]  [Rnd6,0,D.ao,y,y]       /ao(nas)/
[14]  [Rnd6,0,D.aoVelOcc,y,y] /ao(nas)-ng/
[15]  [0,0,VelOcc,y,y]        /ng/.
```

5. DISCUSSION

In this paper, we report our recent theoretical development of an overlapping feature-based phonological model which has long-span contextual dependency. We extend the earlier work by incorporating high-level linguistic/prosodic constaints in automatic construction of the feature overlapping patterns. The linguistic information explored includes utterance, word, morpheme, syllable, syllable constituent categories and word stress markers. A consistent computational framework, based on temporal feature logic, has been developed for the construction of the phonological model.

One use of the feature-based phonological model in automatic speech recognition is to provide the HMM state topology for the conventional recognizers, serving as a pronunciation model that directly characterizes acoustic variability due to phonetic context and speaking style changes. Our future research will involve use of the feature-based model developed in this work to control the dynamic process of speech production at the phonetic level. Seamless interface between the feature-based phonological model and the dynamic phonetic model for the true speech process has the potential to overcome many of the critical limitations of the current HMM-based speech recognition technology which has virtually no structure of the speech process built into the speech model.

6. REFERENCES

1. S. Bird: "Computational Phonology – A constraint-based approach" Cambridge University Press, 1995.
2. C. P. Browman and L. Goldstein: "Articulatory Gestures as Phonological Units" Phonology (6), 1989, pp. 201-251.
3. K. W. Church: "Phonological Parsing in Speech Recognition" Kluwer Academic Publishers, 1987.
4. L. Deng and D. Sun: "A statistical approach to ASR using atomic units constructed from overlapping articulatory features," *J. Acoust. Soc. Am.*, Vol.95, 1994, pp. 2702-2719.
5. L. Deng. "Autosegmental representation of phonological units of speech and its phonetic interface," *Speech Communication*, Vol.23, No. 3, 1997,pp. 211-222.
6. J. A. Goldsmith: "Autosegmental & Metrical Phonology" Blackwell, 1990.