Journal of the Acoustic Society of America, Vol. 111, No. 2, February 2002, pp. 1086-1101

An Overlapping-Feature Based Phonological Model Incorporating Linguistic Constraints: Applications to Speech Recognition

Jiping Sun and Li Deng*

University of Waterloo, Waterloo, Ont, Canada

*current address: Microsoft Research, Redmond, WA 98052

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 that represents long-span contextual dependency in speech acoustics. In this model, high-level linguistic constraints are incorporated in automatic construction of the patterns of feature overlapping and of the hidden Markov model (HMM) states induced by such patterns. The main linguistic information explored includes word and phrase boundaries, morpheme, syllable, syllable constituent categories, and word stress. A consistent computational framework developed for the construction of the feature-based model and the major components of the model are described. Experimental results on the use of the overlapping-feature model in an HMM-based system for speech recognition show improvements over the conventional triphone-based phonological model.

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 the perspective of computational phonology, motivated by some recent versions of nonlinear phonology [2, 11]. The computational framework developed is based on sub-phonemic, overlapping articulatory features where the rule-governed overlap pattern is described mathematically as a finite-state automaton. Each state in the automaton corresponds to a feature bundle with normalized duration information specified [9, 5]. In this paper, we report our new development of the feature-based phonological model which incorporates high-level linguistic (mainly prosodic) constraints for automatic construction of the patterns of feature overlapping and which includes new implementation of the model. We also report positive results of experiment on use of the feature-based model as the HMM state topology generator for speech recognition.

In our feature-based phonological model, patterns of feature overlapping are converted to an HMM state-transition network. Each state encodes a bundle of overlapping features and represents a unique, symbolically-coded articulatory configuration responsible for producing speech acoustics based on that configuration. 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 immediately neighboring segments, this approach is expected to show advantages over the conventional context dependent modeling based on diphones or triphones. Use of diphone or triphone units necessarily limits the context influence to only immediately close neighbors, and demands a large amount of training data because of the large number of the units (especially triphone units) combinatorially generated. Such a drawback is completely eliminated in the overlapping-feature based model described in this paper.

The feature-based phonological model and the conventional, triphone-based model currently most popular in speech recognition [13] are alternative ways of representing words in the lexicon and their pronunciation using HMM states. Their differences can be likened to "atomic" units versus "molecular" units — fine versus coarse scales in representing the fundamental building blocks of speech utterances. Consequences of such a disparity are that the feature-based model provides the long-span context-dependency modeling capability while the triphone model provides only the short-span one, and that the feature-based model is much more parsimonious and economical in lexical representation than the triphone model. This latter advantage is due to the fact that several distinct phones may share common features while feature overlapping concerns only the spreading of such features with no identity changes. As a result, the triphone model has much greater training-data requirements than the feature-based model for speech recognizer construction.

The feature-based model further permits construction of language-independent recognition units and portability of speech recognizers from one language to another in a principled way [7], while the triphone model is not able to do the same. This is because articulatory features are commonly shared by different languages and play important mediating roles in mapping the underlying, perceptually defined phonological units to surface acoustic forms. A feature-overlapping model defined by general articulatory dynamics can potentially generate all possible transitory and allophonic states given canonical articulatory descriptions of phonemes and continuous speech contexts. The task of a training process against a particular language, on the other hand, is to determine a subset of feature-bundles employed by the language so that the underlying units can be correctly "perceived" by the listener in terms of feature-bundle sequences. Therefore, feature bundles derived from context-induced overlapping can form a universal set for describing all sounds in all languages at a mediating level between acoustic signals and the the lexical units. The main challenge for developing the feature-based phonological model is its implementation complexity, which is the main focus of this paper. To what extent the feature bundles obtained from one language's data is shared by another language is both a theoretical topic as well as an empirical issue, and demands further study beyond the scope of this paper.

In our previous work, the feature overlapping rules were constructed based only on the information about the phoneme (i.e., segment) identity in each utterance to be modeled [9, 8, 6]. It is well established [2, 3, 4, 11] that a wealth of linguistic factors beyond the level of phoneme, in particular prosodic information (syllable, morpheme, stress, utterance boundaries, etc.), directly control the low-level feature overlapping. Thus, it is desirable to use such high-level linguistic information to control and to constrain feature overlapping effectively. As an example, in pronouncing the word *display*, the generally unaspirated /p/ is constrained by the condition that an /s/ precedes it in the same syllable onset. On the other hand, in pronouncing the word *displace*, *dis* is a morphological unit of one syllable and the /p/ in the initial position of the next syllable subsequently tends to be aspirated.

In order to systematically exploit high-level linguistic information for constructing the 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 sufficiently comprehensive to cover a wide variety of utterances (including spontaneous speech) so as to be successful in speech recognition. Development of such a methodology is the major thrust of the research reported in this paper.

2 A General Framework of Feature Overlapping

2.1 Use of High-Level Linguistic Constraints

Our general approach to pronunciation modeling is based on the assumption that high-level (e.g. prosodic) linguistic information controls, in a systematic and predictable way, feature overlapping across feature dimensions through long-span phoneme sequences. The high-level linguistic/prosodic information used in the current implementation of the feature-based model for constraining feature overlapping includes

- Utterance, word, morpheme and syllable boundaries. (Syllable boundaries are subject to shifts via resyllabification.)
- Syllable constituent categories: onset, nucleus and coda.
- Word stress and sentence accents.

Morpheme boundary and syllabification are key factors in determining feature overlapping across adjacent phonemes. For example, aspiration of voiceless stops in *dis-place* and in *mis-place* versus non-aspiration of the stop in *di-splay* are largely determined by morpheme boundary and syllabification in these words. In the former case, overlapping occurs at the Larynx tier (See Section 2.2 for the definition of articulatory feature tiers). Utterance and word boundaries condition several types of boundary phenomena. Examples of the boundary phenomena are glottalized word onset and breathy word ending at utterance boundaries, and the affrication rule at word boundaries (e.g., compare *at right* with *try*) [12]. Likewise, association of a phoneme with its syllable constituent influences pronunciation in many ways. For example, stops are often unreleased in coda but not so in onset. An example of the effect of word-stress information on feature overlapping is the alveolar-flap rule which only applies to the contextual environment where the current syllable is unstressed and the preceding syllable is stressed within the same word.

This kind of high-level linguistic constraints is applied to our framework through a predictive model which parses the training sentences into accent groups at the sentence level and syllabic components at the lexical level. The accent group identification is mainly through part-of-speech tagging information. The syllabic component identification is mainly through a context-free grammar parser based on rules of syllable composition by phonemes (see Appendix 1). After this analysis, a sentence is represented by a sequence of symbolic vectors, each containing the phoneme symbol and its syllabic, boundary and accent information which governs the pronunciation of each phoneme in continuous speech. For example, the utterance "The other one is too big" will be represented as:

[dh ons ub] (ons = syllable onset, ub = utterance beginning)

[iy nuc we ust] (nuc = syllable nucleus, we = word end, ust = unstressed)

[ah nuc wb] (wb = word beginning)

[dh ons]

[ax nuc we ust]

[w ons wb]

[ah nuc ust]

[n cod we] (cod = syllable coda)

[ih nuc wb ust]

[s cod we]

[t ons wb]

[uw nuc we str] (str = stressed)

[b ons wb]

[ih nuc str]

[g cod ue] (ue = utterance end)

The the above and throughout this paper, we use the ARPAbet symbols to represent phonemes. In the later part of the paper we will explain how high-level information constrains feature overlapping, and thus influences speech recognition model building.

2.2 Feature Specification for American English

We use a consistent feature specification system for transforming segment symbols to feature bundles, which is carried out after syllable parsing and before the application of feature overlapping rules. This system is characterized by the following key aspects:

- Five feature tiers are specified, which are: Lips, Tongue-Blade, Tongue-Dorsum, Velum, and Larynx.

- The feature specification of segments is context independent; it shows canonical articulatory properties coded in symbolic forms. (The total repertoire of the feature values we have designed is intended for all segments of the world languages. For a particular language, only a subset of the repertoire is used.)

- Open (underspecified) feature values are allowed in the feature specification system. These underspecified feature values may be partially or fully filled by temporally adjacent (specified) features during the rule-controlled feature overlapping process.

The feature specification system we have worked out for American English has the following specific properties. A total of 45 phonemes are classified into 8 categories: stops, fricatives, affricates, nasals, liquids, glides, (monophthong) vowels and diphthongs. Each phoneme is specified with a five-dimensional feature bundle, corresponding to the five articulators: Lips, Tongue-blade, Tongue-body, Velum, and Larynx. The values for each dimension are symbolic, generally concerning the place and manner of articulation (which are distinct from other phonemes) for the relevant articulator. The feature values for any (canonically) irrelevant articulator are underspecified (denoted by the value "0").

Continue with the above example. After the phonemes are replaced by articulatory features [before overlapping], the utterance "The other one is too big" becomes (the explanations of the prosodic symbols are given in the example on Section 2.1.):

[dh(0 ClsDen 0 0 V+) ons ub] (ClsDen = dental closure, V+ = voiced)

[iy(0 0 D.iy 0 V+) nuc we ust] (D.iy = tongue dorsum position of /iy/)

[ah(0 0 D.ah 0 V+) nuc wb]

[dh(0 ClsDen 0 0 V+) ons]

[ax(0 0 D.ax 0 V+) nuc we ust]

[w(Rnd.u 0 D.w 0 V+) ons wb] (Rnd.u = lip rounding of /u/)

[ah(0 0 D.ah 0 V+) nuc ust]

[n(0 ClsAlv 0 N+ V+) cod we] (ClsAlv = alveolar closure, N+ = nasal)

[ih(0 0 D.ih 0 V+) nuc wb ust]

[s(0 CrtAlv 0 0 V-) cod we] (CrtAlv = alveolar critical, V- = unvoiced)

[t(0 ClsAlv 0 0 V-) ons wb]

[uw(Rnd.u 0 D.uw 0 V+) nuc we str]

[b(ClsLab 0 0 0 V+) ons wb] (ClsLab = labial closure)

[ih(0 0 D.ih 0 V+) nuc str]

[g(0 0 ClsVel 0 V+) cod ue] (ClsVel = velum closure)

Some further detail is given here on the featural representation of the segments. Generally, we use a single feature bundle to represent a segment in its canonical state. This, to some extent, ignores some finer structures. For example, the stops have at least two distinctive phases: the closure phase and the release phase. To account for this, finer structures are needed and they are modeled by the derived feature bundles. For instance, the release phase of the stops is represented by a derived feature bundle between the stop and an adjacent segment. The derived feature bundle for the release phase still contains such a feature as ClsAlv (e.g. for /t/) or ClsLab (e.g. for /p/), but it is understood differently as will be illustrated in the examples of the subsection 2.4.

2.3 A Generator of Overlapping Feature Bundles

An overlapping feature bundle generator is a program which 1) scans the input sequence of feature bundles with high-level linguistic information; 2) matches them to corresponding overlapping rules; 3) executes overlapping (or mixing) operations specified in the overlapping rules during two separate, leftward-scan and rightward-scan processes; The execution starts from the right-most phoneme for the leftward-scan process, and it starts from the left-most phoneme for the rightward-scan process. and 4) integrates the results of leftward-scan and rightward-scan to produce a state-transition network. A block diagram of the overlapping feature bundle generator is shown in Figure 1.

PLACE Figure 1 AROUND HERE

Our feature-overlapping rules contain two types of information (or instruction): possibility information and constraint information. The possibility component specifies what features can overlap and to what extent, regardless of the context. The constraint component specifies various contexts to constrain feature-overlapping. Below we give some examples of possibility and constraint:

- Possibility of Velum Feature Overlapping: A velum lowering feature can spread left and right to cause the phenomenon of nasalization in some phones, such as vowels.

- Possibility of Lip Feature Overlapping: A lip rounding feature can spread mainly to the left to cause the phenomenon of lip-rounded allophones.

- Possibility of Tongue Body Feature Overlapping: A tongue body feature can spread to cause such phenomenon in stops as advanced or retracted tongue body closures (as in /g iy/ versus /g uh/).

- Possibility of Larynx Feature Overlapping: A voicing/unvoicing feature can spread to cause such phenomena as voiced/unvoiced allophones.

- Possibility of Tongue Tip Feature Overlapping: The tongue tip feature of /y/ can spread into the release phase of a stop to cause the phenomenon of palatalization (as in "did you").

- Constraint rule: A stop consonant blocks feature spreading of most features, such as lip feature, larynx feature, etc.

- Constraint rule: A vowel usually blocks tongue body features from spreading through it.

The above spreading-and-blocking model can account for many types of pronunciation variation found in continuous speech. But there are some other common phenomena that cannot be described by feature spreading only. The most common among these are the reductive alternation of vowels (into schwa) and consonants (flapping, unreleasing, etc.). Therefore, our model needs to include a control mechanism that can utilize high-level information to "impose" feature transformation in specific contexts. We give some examples below:

- Context-controlled transformation: A stop consonant undergoes a flap transformation in such contexts as: [V stressed] * [V unstressed] (where '*' marks the position of the consonant in question).

- Context-controlled transformation: A stop consonant deletes its release phase in a coda position.
- Context-controlled transformation: A vowel undergoes a schwa transformation in an unstressed syllable of an unaccented word in the utterance.

The output of the generator is a state-transition network consisting of alternative feature bundle sequences as the result of applying feature-overlapping rules to an utterance. This structure directly corresponds to the state topologies of hidden Markov models of speech. Each distinctive HMM state topology can be taken as a phonological representation for a word or for a (long-span) context-dependent phone. The HMM parameters, given the topology, are then trained by cepstral features of the speech signal. In the following subsection, we give two examples of applying the feature-overlapping rules (details will be presented in Section 3), and show the results in the form of the constructed overlapping feature bundles.

2.4 Examples: Feature Bundles Generated by Applying Feature Overlapping Rules

We present two examples to illustrate typical applications of the feature overlapping rules utilizing high-level linguistic information before details of the rules are formally described. The first example shows how the words *display* and *displace* are endowed with different feature structures in the stop consonant /p/, despite the same phoneme sequence embedding the /p/. The difference is caused by different syllable structures. After syllable parsing and feature overlapping, the results in feature bundles, accompanied by the spectragrams of the two words, are shown in Figure 2. Due to different syllable structures: (/d ih s . p l ey s/ versus /d ih . s p l ey/), different overlapping rules are applied. This simulates the phonological process in which the phoneme /p/ in *displace* tends to be aspirated but in *display* unaspirated.

PLACE Figure 2 AROUND HERE

The two relevant feature bundles are shown in the figure by the dashed vertical lines. The difference lies in the voicing feature at the larynx feature tier. The aspiration is indicated by a V- feature in the feature bundle of the word *displace* between /p/ and /l/. Phonologically, this is called delayed voicing in the onset of /l/. In the model, this is realized through asynchronous leftward spreading of the tongue blade and larynx features of /l/, which overlap with the features of /p/.

The second example (Figure 3) shows the word *strong*, which contains several feature overlaps and mixes. (Feature mixes are defined as feature overlaps at the same feature tier). Some of them have variable durations (in lip-rounding and nasalization), represented by the dashed boxes. Such variability in the duration of feature overlapping gives rise to alternative feature bundle sequences. By merging identical feature bundles, a network can be constructed, which we call the “state transition network”. Each state in the network corresponds to a feature bundle. The network constructed by the overlapping feature bundle generator for the word *strong* is shown in Figure 4, where each state is associated with a set of symbolic features. The branches in the network result from alternative overlapping durations specified in the feature overlapping rules.

PLACE Figures 3 and 4 AROUND HERE

Generally, a derived feature bundle with overlapping features from adjacent segments represents a transitional phase (coarticulation) between phonemes in continuous speech. Overlapping in real speech can pass several phonemes and our feature-overlapping model effectively simulates this phenomenon. For example, in *strong* /s t r ao ng/, the lip rounding feature of /r/ can spread through /t/ to /s/, and the nasal feature of /ng/ can also pass through /ao/ to /r/, as is shown in Figure 3. This ability to model long-span phonetic context is one of the key characteristics of this model.

3 Implementation of the Feature-Overlapping Engine

3.1 The Demi-Syllable as the Organizational Unit in Formulating Feature-Overlapping Rules

Based on the information obtained by the syllable parser and the feature specification (including underspecification) of phonemes, demi-syllables are constructed, which are operated upon by the feature-overlapping rules (formally defined below) to generate transition networks of feature bundles. A demi-syllable in our system is a sequence of broad phoneme categories encompassing the phonemes in either syllable-onset plus nucleus, or nucleus plus syllable-coda formations, together with high-level linguistic information. When a syllable has no onset or coda consonants, that demi-syllable will be only a vowel. The broad phonetic categories we have used are defined as follows:

- V – vowel,
- GLD – glide,
- LQD – liquid,
- NAS – nasal,
- AFR – affricate,
- FRI1 – voiced fricative,
- FRI2 – voiceless fricative,
- STP1 – voiced stop,
- STP2 – voiceless stop.

Other elements included in a demi-syllable are related to the higher-level linguistic information. These include:

- ons – syllable onset,
- nuc – syllable nucleus,

- cod – syllable coda,
- ub – utterance beginning,
- ue – utterance end,
- wb – word beginning,
- we – word end,
- str – stressed syllable in the utterance,
- ust – unstressed syllable.

For instance, the demi-syllables of the utterance "The other one is too big", including high-level linguistic information, are as follows:

[FRI1 ons ub] [V nuc we ust] (dh-iy)

[[V, nuc, wb]] (ah)

[FRI1 ons] [V nuc we ust] (dh-ax)

[GLD ons wb] [V nuc ust]] (w-ah)

[V nuc ust] [NAS cod we]] (ah-n)

[V nuc wb ust] [FRI2 cod we] (ih-s)

[STP2 ons wb] [V nuc we str] (t-uw)

[STP1 ons wb] [V nuc str]] (b-ih)

[V nuc str] [STP1 cod ue] (ih-g)

Demi-syllables split a full syllable (one with both onset and coda consonants) into two halves. The purpose of this splitting is to make a small set of units for practical rule development. Contextual constraints specified in the phonological rules are defined on the demi-syllables. After parsing all the 6110 words in the TIMIT corpus dictionary, we obtained 291 distinct word-based demi-syllables (that is, without specifying utterance boundaries and utterance accents, which can be included in later rule development). This is a compact set, facilitating the development of the overlapping rule system which we now describe in detail.

3.2 Overlapping Phonological Rule Formulation

This subsection gives a detailed description of the phonological rules for articulatory feature overlapping. Appendix 2 presents a logical basis of our feature-overlapping system in the form of a temporal logic. This logic is based on autosegmental and computational phonological theories, presented in [1, 2, 11] and elsewhere. The phonological rules have been formulated systematically based on the behavior of articulatory features, especially under the influence of high-level linguistic structures. The phonological rules are used to map any utterance from its demi-syllable representation into its corresponding feature bundle network (i.e. the state transition graph).

The data structure of feature-overlapping rules consists of "overlapping patterns" and "overlapping operators". Each overlapping pattern is defined with respect to a demi-syllable and contains the names of a number of overlapping operators. The demi-syllable, as illustrated in the last subsection, contains both segmental information (broad phonetic categories) and high-level linguistic information (boundaries, accents and syllable constituents). The construction of overlapping patterns starts from the 291 word-based demi-syllables. Based on the temporal logic and particular phonological knowledge concerning coarticulation and phonetic alternations, necessary boundary and accent requirements are added. Further, a number of overlapping operators' names are added to form an overlapping pattern. Each operator corresponds to a broad phonetic category in the demi-syllable.

The overlapping operators are defined on the phonemes based on phonological theory, describing how their articulatory features may overlap in speech. When an overlapping pattern is applied, an operator name will point to the actual definition, which then is applied to the corresponding phoneme matching a broad phonetic category. One definition of an operator may be pointed to by more than one overlapping pattern. Thus, the overlapping operators realize the possibilities while the overlapping patterns realize the constraints on the possibilities. (The concepts of possibility and constraint were discussed in subsection 2.3.)

Let us denote a broad phone category in a demi-syllable by DSC (standing for demi-syllable constituent), then a phonological rule is described by a list of DSC's in a demi-syllable, together with all possible operators allowed to operate on each DSC. The overall data structure of a phonological rule is in this form:

[DSC-1: operator1.1, operator1.2, operator1.3 ... (high-level information)]

[DSC-2: operator2.1, operator2.2, operator2.3 ... (high-level information)]

[DSC-3: operator3.1, operator3.2, operator3.3 ... (high-level information)]

...

An operator describes how feature overlapping could happen on different articulatory tiers, as is described in phonological theory, such as "lip rounding", "jaw lowering", "palatalization", etc. Each operator consists of four components: 1) action, 2) tier-specification, 3) feature-value constraint, and 4) relative-timing. Below we discuss each of these components. First, there are three choices for describing an action:

- L or R: For leftward (look-ahead) or rightward (carry-over) feature spread from an adjacent phoneme onto an underspecified tier of the phoneme.
- M or N: For leftward or rightward mixture of a feature from an adjacent phoneme on the same tier.
- S: For substitution of a feature value by a different feature value.

Second, a tier-indicator specifies at which feature tier an action takes place. A tier indicator is given by an integer as follows:

- 1: the Lips tier,
- 2: the Tongue-Blade tier,
- 3: the Tongue-Dorsum tier,

- 4: the Velum tier,

- 5: the Larynx tier.

Third, a value constraint can optionally be given to stipulate that a feature spread from an adjacent phoneme must have a specified value. If this value constraint is not given, the default requirement is that on this tier of an adjacent phoneme there must be a specified feature in order for the operator to be applicable.

Fourth, a relative-timing indicator is used to specify the temporal extent of a feature spreading. In the current implementation of the model, we use four relative-timing levels: 25%, 50%, 75%, and 100% (full) with respect to the entire duration of the phoneme.

The reader may wonder how long-span effects are realized in this model. This is realized by full (100%) feature spreading. Once an adjacent phoneme's feature is spread to the entire duration of the current phoneme, that feature is visible to the adjacent phoneme on the other side and may spread further. For example, a nasal feature from a right adjacent phoneme may be allowed to spread to the full duration of a vowel. The phoneme to the left of the vowel can "see" this feature and may allow it to spread into itself. This is the mechanism used by the model to pass a feature over several phonemes until it is blocked.

The naming of an operator follows a syntax which reflects its internal definition. The syntax for an operator name is given as:

$$\text{Operator-Name} := Op\ N^{+}\ [\ @\ N^{+}\ ]$$

$$N := 1 \mid 2 \mid 3 \mid 4 \mid 5$$

where the numbers after 'Op' reflect the tier-indicators in the definition, and the optional numbers after the symbol @ stands for the tiers at which feature-value constraints are imposed.

A phoneme can be given a number of operators. Whether an operator is allowed to apply to a phoneme depends on whether it is listed in a DSC of an overlapping pattern. Furthermore, whether an operator listed in an DSC can be fired or not depends on if the conditions in the operator definition are met. For example, for the operator with the name Op2 to fire, the second tier of the adjacent phoneme must have a specified feature value. As another example, for the operator of the name Op12@2 to fire, the adjacent phoneme (whether it is to the left or right depends on the action type of the operator) must have specified features at tier 1 and 2 and the feature value at tier 2 must match the value specified in its definition.

As an illustration, Figure 5 shows the result of applying an operator named $Op125@15$ to the feature bundle of /t/ when it is followed by /r/. The operator is defined as

(125@15, tier_1.L.rnd, tier_2.M, tier_5.L.V+, time:(.5,.25,.25; 1,.25,.25)).

According to this definition, the three tiers of the phoneme – Lips (1), Tongue-Blade (2) and Larynx (5) have actions M or L. Tiers 1 and 5 constrain the spreading feature values as *rnd* and $V+$ that come from a right neighbor. There are two alternative timing specifications (.5,.25,.25) and (1,.25,.25). Feature spreading at the three tiers will enter the feature bundle of /t/ in two possible ways: 1) Lips feature spreading to 50% of the entire duration, and

Tongue-Blade and Larynx features spreading to 25%, or 2) Lips feature spreading to the entire duration and the Tongue-Blade and Larynx feature spreading to 25%. As a consequence, two new feature bundles are derived. The two possible ways for state transitions are shown in Figure 5, which is automatically derived by a node-merging algorithm accepting parallel state sequences. Note how long-distance feature overlapping can be realized by the rule mechanism: Once a feature spreading covers an entire duration, this feature will be visible to the next phoneme. Now we give an example of a phonological rule, which is defined on the demi-syllable with high-level linguistic structure:

[FRI2 ons wb] [STP2 ons] [LQD ons] [V nuc str]

This demi-syllable can match the first four phonemes of the word *strong*. This rule is expressed as:

[FRI2 (Op2, Op3, Op13@1) ons wb]

[STP2 (Op2, Op125@15) ons]

[LQD (Op3, Op34@4) ons]

[V (Op3, Op34@4) nuc str]

Each DSC in this rule is given a number of operators which can operate on the phonemes that are matched by the demi-syllable. Notice the high-level linguistic structures (ons, wb, etc.) which constrain the application of the rule to certain prosodic context. In the current implementation of the feature-based model, we have the following operator inventory which consists of a total of 26 operators defined for the 44 English phonemes for the leftward scanning. A corresponding set of operators for rightward scanning are similarly defined. We list the leftward operators as follows:

1. `(Op1,1.M,(.25))` (transitional phase)
2. `(Op1,1.L,(.25))`
3. `(Op2,2.M,(.25))`
4. `(Op2,2.L,(.25))`
5. `(Op3,3.M,(.25))`
6. `(Op3,3.L,(.25))`
7. `(Op5,5.S,())` (glottal substitution)
8. `(Op2,2.S,())` (tongue blade substitution)
9. `(Op4,4.L.N+,(.5;1))` (nasalization)
10. `(Op12@1,1.L.rnd,2.M,(.5,.25;1,.25))` (transition with lip rounding)
11. `(Op13@1,1.M.rnd,3.L,(.5,.25;.25,.25))`
12. `(Op13@1,1.L.rnd,3.M,(.5,.25;1,.25))`
13. `(Op13@1,1.L.rnd,3.L,(.5,.25;1,.25))`

14. `(Op14@4,1.L,4.L.N+,(.25,.5;.25,1))` (transition with nasalization)

15. `(Op24@4,2.L,4.L.N+,(.25,.5;.25,1))`

16. `(Op34@4,3.M,4.L.N+,(.25,.5;.25,1))`

17. `(Op23@2,2.S.TapAlv,3.L,(.25,.75;1,.25))`

18. `(Op34@4,3.M,4.l.N+,(.25,.5;.25,1))`

19. `(Op34@4,3.L,4.L.N+,(.25,.5;.25,1))`

20. `(Op35@5,3.M,5.L.V+,(.25,.25))` (transition with unaspiration)

21. `(Op35@5,3.L,5.L.V+,(.25,.25))`

22. `(Op125@15,1.L.rnd,2.M,5.L.V+,(.5,.25,.25;1,.25,.25))` (more combinations)

23. `(Op134@14,1.M.rnd,3.L,4.L.N+,(.5,.25,.5;.5,.25,1;1,.25,.5) )`

24. `(Op134@14,1.L.rnd,3.L,4.L.N+,(.5,.25,.5;.5,.25,1;1,.25,.5) )`

25. `(Op135@15,1.M.rnd,3.L,5.L.V+,(.5,.25,.25;1,.25,.25))`

26. `(Op135@15,1.L.rnd,3.L,5.L.V+,(.5,.25,.25;1,.25,.25))`

PLACE Figure 5 AROUND HERE

To illustrate the use of overlapping phonological rules and how high-level linguistic information is incorporated, we demonstrate with the example utterance “a tree at right” (the corresponding phoneme sequence is /ax t r iy ae t r ay t/). After prosodic processing, where part-of-speech tagging and shallow syntactic parsing is used for deriving the boundary and accent information, and following syllable parsing, the utterance is represented by a sequence of demi-syllables:

1. [V nuc ub ust] (ax)
2. [STP2 ons wb] [FRI1 ons] [V nuc we str] (t-r-iy)
3. [V nuc wb ust] [STP2 cod we] (ae-t)
4. [FRI1 ons wb] [V nuc str] (r-ay)
5. [V nuc str] [STP2 cod ue] (ay-t)

Each demi-syllable is matched by a phonological rule. The overlapping operators in each DSC are tried for firing. If the conditions are met, an operator is fired to derive feature bundles. During the derivation process, segment and word boundaries are recorded to "cut up" the derived network into word networks or phone networks, which are used to build word or phone-based hidden Markov models.

In this example, we illustrate the use of syllable information to realize the "affrication rule" discussed earlier in subsection 2.1. The utterance's wave form, spectragram and relevant features concerning the use of the affrication rule are shown in Figure 6. To realize the affrication rule, the phonological rule matching the second demi-syllable: [STP2 ons wb] [FRI1 ons] [V nuc we str] will have its first DSC assigned an operator: (Op2,2.L,(.25)) which allows feature overlapping on the tongue blade tier. The overlapping phonological rule matching the third demi-syllable, on the other hand, will not assign this operator to the second DSC: [STP2 cod we], blocking affrication.

PLACE Figure 6 AROUND HERE

As another example of applying high-level linguistic information, consider the use of a substitution action in an operator at utterance beginning. For the above utterance, a rule matching the first demi-syllable: [V nuc ub ust] can have an operator with a glottal substitution action: (Op5, 5.S.?, ()). This simulates an utterance with a glottal stop at the outset. Similarly, an un-released stop consonant at the end of a word or an utterance can be simulated by the phonological rule mechanism as well.

We have illustrated how “possibilities” and “constraints” can be implemented by the overlapping patterns and operators. With each DSC within a rule there may be a number of operators available for firing. When more than one operator can be fired, it is the more specific ones that are fired first. Depending on how complex we expect the generated network to be, the system is able to control how many operators to be fired.

4 Speech Recognition Experiments

In this section we describe the speech recognition experiments using the phonological rules and the generator of overlapping feature bundles described earlier in this paper. Our experiments are carried out using the TIMIT speech database and the tasks are both (continuous) word and phone recognition. Our preliminary experimental results show that this feature-based approach is a promising one with a number of new directions for future research.

4.1 Automatic Creation of HMM Topology with Feature-Bundle States

The feature-based speech recognizer we have constructed uses a special HMM topology to represent pronunciation variability in continuous speech. The variability is modeled by parallel feature-bundle state sequences as a result of applying the phonological rules to the canonical phoneme representations. The HMM topology is created automatically by rules, one for each word. Details of this process have been provided in Section 3 and we summarize this process as the following six steps for the TIMIT corpus:

1. Parse each phoneme string in a sentence into a syllable sequence, and

   further into a demi-syllable sequence with prosodic structure;

2. Match the demi-syllable sequence to a sequence of corresponding feature-overlapping patterns;

3. Select the relevant feature-overlapping operators (defined in the feature-overlapping pattern) for each phoneme according to its featural context in the sentence;

4. Apply the operators in the order from most specific to most general, with complexity control;

5. Generate a full set of overlapped or mixed feature bundles (and use them as the HMM states), as the result of the applications of feature-overlapping rules;

6. Generate state-transition graphs for all the words (and sentences) in the TIMIT database based on the parallel feature-bundle transition paths.

The last step creates the feature-based pronunciation models in the form of word-HMM's for all 6110 TIMIT words. To show the parsimony of the feature-based approach, only 901 distinct HMM states (i.e. 901 distinct feature bundles) were derived and used to represent these 6110 words, in contrast to tens of thousands generated by the conventional triphone approach. Furthermore, long-span context dependence has been incorporated due to the application of long-span feature-overlapping rules.

Given the HMM topology automatically created for each word in TIMIT, we used the HTK tools to compute the speech features (MFCC) and to train the continuous-density HMM output-distribution parameters (means, variances, and mixture weights) for all 901 unique feature bundles (HMM states) using the training data in TIMIT. The HMM's trained were then used to automatically recognize the TIMIT test-set sentences, using HTK Viterbi decoder (HVite tool).

The training and recognition with network HMM's (see Figure 4), which contain multi-path graphs, is allowed by the HTK tool as it is designed for experimenting with different model structures including multi-path topologies. We use the global mean and variance from the entire data set to initialize the models, and then use Baum-Welsh re-estimation to compute the parameters specific to each state. The re-estimation procedure (HERest tool) applied to the models avoids the alignment problem as may occur with multi-path structures because of the following two reasons. First, all the states which are derived from the same feature bundle are tied from the beginning. Second, when a branch occurs in some model, the alignment between data and alternative states is resolved when there is similar data elsewhere in the corpus aligned with a non-branching state which is tied with one of the alternative states.

4.2 Statistics in Training and Testing Data

The TIMIT database used in our experiments consists of 630 speakers in 8 dialect regions, of which 462 are in the training set and 168 are in the testing set. The sentences in the training and the testing sets are disjoint, except for two sentences which were spoken once by every speaker. The training set contains 4620 sentences and the testing set 1680. The training set contains 4890 distinct words and the testing set 2375. Among the total 6110 words in TIMIT, 1155 words occur in both the training and testing sets and 1220 words are unique to the testing set (i.e., distinct from all words in training).

The entire set of TIMIT words (training and testing sets) gives rise to a total of 901 HMM states after the application of the overlapping rules described in Section 3. Among all the 901 states, the testing set contains 754 states, of which 717 states also occur in the training set. This shows the advantage of the feature-based approach: in contrast to around 48%

sharing of words (1155 out of 2375), the sub-phonemic, feature-bundle sharing is over 95% (717 out of 754) for the testing set. This means that with about 52% of the words unseen in the training model, when it comes to feature-bundle based states, the unseen portion in the training set drops to only about 5%. For the 37 states occurring uniquely in the testing set, we synthesized them with the parameters of the states obtained from the training set which have similar features as the "unseen" states, using a feature vector similarity metric.

In short, in contrast to words, the training and testing sets differ less in terms of feature bundles. The 4890 words in the training set account for 95% of feature bundles in the words of the testing set, although they only account for 48% of the words in the testing set.

4.3 Speech Recognition Results

Using the embedded estimation tool HERest in the HTK, we trained the word-HMM's by direct tying. This means that the 901 states were used for all the words from the very beginning. Unlike the triphone training procedure which undergoes a separate state-tying process, the direct tying training was efficient in terms of both training time and memory space requirements. We estimated single-Gaussian state models twice. Then the mixture number was increased gradually to five, with one re-estimation for each increase.

This amounts to using the feature-overlapping model to construct the word-level HMM's. Since the testing set has half of the words distinct from that of the training-set, these "unseen" word HMM models are synthesized with state macros (symbolic names pointing to the trained states). We have carried out speech recognition (decoding) experiments using the HMM's, obtained by the above training procedure. Details of the recognition performance are shown in Table 1, where the word error rate (WER) and sentence error rate (Sent.ER), as well as the word substitution (Sub), deletion (Del), and insertion (Ins) rates, are shown as a function of the dialect regions (Dial.Reg.) in the TIMIT database. The size of the testing set in terms of the total number of test words and sentences in each of the dialect regions is also listed. These results are obtained on all the 1680 sentences in the TIMIT testing set covering all eight dialect regions of American English accents. A bigram language model was used, which was derived from the whole set of TIMIT prompt sentences, with one-gram probabilities lowered to -99. A five-Gaussian mixture was used as the output distribution for each of the 901 feature bundle-based HMM states.

PLACE Table 1 AROUND HERE

The efficiency of the feature-based system was evident in the experiments. For example, the state set from the very beginning was compact and the training time was also much less compared to the triphone system, at the ratio of about 1/20.

In a further experiment, we used the data-driven state clustering functionality provided by the HTK toolkit in the overlapping feature framework with unified model topologies. We performed the TIMIT phone recognition task by using 39 three-state, left-to-right, no-skip phone models trained as quinphones. Compared with triphones, a quinphone incorporates contexts of up to two phones to its left and right. This gives the possibility of utilizing the predictions made by the feature-overlapping model. The predictions were used to form decision tree questions for state tying.

The training set of TIMIT database resulted in 64230 context-dependent quinphones. The overlapping features germinating from five-phone contexts were used in designing decision-tree questions for state tying. The contexts that affect the central phones through feature overlapping, as predicted by the model, form questions for separating a state pool (a technique of state tying with decision trees). For example, the nasal release of stop consonants in such contexts as /k aa t ax n/ and /l ao g ih ng/ (the /t/ in the first context and /g/ in the second context, influenced by /n/ and /ng/) will induce questions for tying the third state of the three-state model with the conditions expressed as *+ax2n, *+ax2ng, etc. ('2' is used here to separate the first and the second right context phones). With the aid of such decision-tree questions, the quinphone states were tied and re-estimated. The testing result is compared with the triphone baseline results for the 39-phone recognition defined in the TIMIT database. This comparison is shown in Table 2, where both systems are used to recognize the same 1680 test utterances that consists of a total of 53484 phone tokens. The results in Table 2 show that the feature-overlapping model outperforms the conventional triphone model. The feature-overlapping model is able to make meaningful predictions, which lead to increase of the efficiency of model organization and training process. Without this predictive model, it would have been impossible to form meaningful state tying questions.

PLACE Table 2 AROUND HERE

Our third experiment used phone-level HMM's to perform word recognition. This is done via a pronunciation dictionary in which each word is represented by one or more sequences of phone HMM models. We used four basic types of predefined phone models, representing stop consonants, other consonants, single vowels and diphthongs respectively. The design of the HMM topologies is based on the assumption that high-level linguistic structures can influence the acoustic properties of the pronounced phonemes and this is reflected in the model structures. The design of each of the four types of phone models is given below:

1. Stop Consonants: Three HMM states, one skip from the second state to the exit dummy state, modeling non-release of stop consonants, one skip from the first state to the exit state, modeling a very short duration without release. The loss of release phase is expected to occur mainly in the coda position.

2. Other Consonants: Three HMM states, one skip from the first state to the third state, modeling a short duration in which the central state has no acoustic data, such as in fast spontaneous speech when the whole duration is influenced by the left and right contexts.

3. Monophthongs (single vowels): Four HMM states. The middle two states are in parallel. These two middle states model stressed and unstressed phones respectively, depending on the sentential accent of the phone. One skip from the first to the fourth state, modeling (optionally) faster speech.

4. Diphthongs: Five HMM states. The second and third states are in parallel, modeling the stressed and unstressed phones respectively, depending on the sentential accent of the phone. One skip from the first to the fifth state, modeling fast speech.

These models were first trained as monophones. Then they were expanded into quinphones and re-estimated in their individual contexts. Next, their boundary states were tied by decision-tree based state tying. The decision tree questions were formed again by feature-based model predictions. Finally, the models were re-estimated with increased mixtures. The unseen quinphones were synthesized by the HTK state tying algorithm.

The difference of this framework from a triphone baseline word recognition system lies in the topology design for utilizing high-level linguistic information and the state tying questions used by decision trees. The results of testing with the TIMIT database are shown in Table 3. These results demonstrate superior performance of the overlapping-feature based approach over the triphone based one.

In this experiment, we used a bigram language model similar to the one used in the first experiment. The only difference is that the one-gram word probabilities were not lowered, which accounted for the lower accuracy compared to the first experiment.

PLACE Table 3 AROUND HERE

5 Summary and Discussion

We have reported our recent theoretical development of an overlapping-feature based phonological model which includes long-span contextual dependencies. Our most recent implementation of the model and some speech recognition experiments using the TIMIT data have been described. We extended our earlier work [9, 5] by incorporating high-level linguistic structure constraint in the automatic construction of feature-based speech units. The linguistic information explored includes utterance and word boundaries, syllable constituents and word stress. 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, which, as reported in this paper, is to provide an HMM state topology for the conventional recognizers, serving as a pronunciation model that directly characterizes phonological variability. We have built a feature-based speech recognizer using the HTK toolkit for this purpose, and the implemented recognizer is reported in detail in this paper.

The overlapping-feature based phonological model described in this paper is a significant improvement upon a number of earlier versions of the model. The earliest version of the model automatically created an HMM topology based on simple, heuristic rules to constrain feature overlaps [9]. A total of 1143 distinct HMM states are created for the TIMIT sentences. When that model was used for the task of phonetic classification (TIMIT database), the phone classification accuracy of 72% was achieved using as few as one-tenth of the full training data. The next version of the model improved the phonological rules for constraining feature overlaps, and interfaced the feature-bundles with the HMM states which are nonstationary (polynomial) [8, 6]. The new rules created a total of 1209 distinct HMM states for the TIMIT sentences. Evaluation on TIMIT phonetic recognition (N-best) gave 74% phonetic recognition accuracy (and 79% correct rate excluding insertion errors). A further version of the model abandoned all rules to constrain feature overlaps, and allowed all features to freely overlap across the feature tiers [10]. This created an unmanageable number of distinct feature-bundles which rendered the HMM recognizer untrainable. The solution to this problem as reported in [10] was to use an automatic decision-tree clustering or tying algorithm (based on the acoustic clustering criterion) to reduce the total number of distinct HMM states needed for reliable HMM training. Evaluation on TIMIT phonetic recognition showed the same performance as the decision-tree clustered triphone units. This demonstrated the weaknesses of using acoustic information only without incorporating phonological information.

The current version of the model presented in this paper re-focused on the phonological rules, and it differs from all the previous versions of the model in the following significant aspects: 1) It incorporates high-level (above phoneme level) linguistic information which is used to control, in a systematic and predictable way, the feature overlaps across feature tiers through long-span phoneme sequences; 2) It formulates the phonological rules in terms of actions of operators which determine detailed behavior of feature overlaps; and 3) It has been completely re-implemented in Prolog (all the previous versions of the model were implemented in C).

The work reported in this paper initiates new efforts of systematic development of feature-based pronunciation modeling for automatic speech recognition. In this first stage of the work, we successfully implemented the theoretical constructs in terms of rule formalisms and programs generating state-transition graphs. The experimental results demonstrated feasibility of the model in speech recognition applications. In our future work, intensive efforts will be devoted to automatically acquiring more effective feature overlapping rules and to developing more effective ways of building speech recognition systems using feature-overlapping models. A data-driven feature-overlapping rule modification system will also be developed to test precision of the feature overlapping predictions and to automatically adjust the predicted articulatory feature bundles during the recognizer training and decoding phases.

Acknowledgements

Discussions with and some experimental contributions by Jeff Ma and Xing Jing are gratefully acknowledged. Support of this work has been provided by NSERC, Canada.

References

[1] S. Bird. “Computational Phonology – A constraint-based approach” Cambridge University Press, 1995.

[2] C. 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] J. Coleman. “Phonological Representations,” Cambridge University Press, 1998.

[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] L. Deng. “Finite-state automata derived from overlapping articulatory features: A novel phonological construct for speech recognition,” *Proceedings of the Workshop on Computational Phonology in Speech Technology,* (Association for Computational Linguistics), Santa Cruz, CA, June 28, 1996. pp. 37-45.

[7] L. Deng. “Integrated-multilingual speech recognition using universal phonological features in a functional speech production model,” *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Munich, Germany, 1997, Vol. 2, pp. 1007-1010.

[8] L. Deng and H. Sameti. "Transitional speech units and their representation by the regressive Markov states: Applications to speech recognition," *IEEE Transactions on Speech and Audio Processing*, Vol.4, No.4, July 1996, pp. 301–306.

[9] L. Deng and D. Sun. "A statistical approach to automatic speech recognition using the atomic units constructed from overlapping articulatory features," *J. Acoust. Soc. Am.*, Vol.95, 1994, pp. 2702-2719.

[10] L. Deng and J. Wu. "Hierarchical partitioning of articulatory state space for articulatory-feature based speech recognition," *Proceedings of the International Conference on Spoken Language Processing*, Philadelphia, PA, October 3-6, 1996, pp. 2266-2269.

[11] J. A. Goldsmith. *Autosegmental and Metrical Phonology* Blackwell, 1990.

[12] J. T. Jensen. *English Phonology* John Benjamins Publishing Company, 1993.

[13] C.-H.Lee, F. Soong, and K. Paliwal (eds.) *Automatic Speech and Speaker Recognition – Advanced Topics,* Kluwer Academic, 1996.

Appendix 1. A Parser for English Syllable Structure

The syllable structures of words are obtained by a recursive transition network-based phonological parser [3], using a pronunciation dictionary. The transition network is derived from a set of context-free grammar (CFG) rules describing the syllable structure of English words. The CFG rules are obtained by reorganizing and supplementing several lists found in [12]. These rules have been tested for all 6110 words in the TIMIT dictionary. The CFG rules used for constructing the transition network are as follows:

Word → [Init-Onset] V [CvCluster] [Final-Coda]

Init-Onset → C | p,l | p,r | p,w | p,y | b,l | b,r | b,w | b,y | t,r | t,w | t,y | d,r | d,w | d,y | k,l | k,r | k,w | k,y | g,l | g, r | g,w | g,y | f,l | f,r | f,y | v,l | v,r | v,y | th,r | th,w | th,y | s,p | s,p,y | s,t | s,t,y | s,k | s,k,y | s,f | s,m | s,n | s,l | s,w | s, y | s h,m | sh,l | sh,r | sh,w | hh,y | hh,w | m,y | n,y | l,y | s,p,l | s,p,r | s,t,r | s,k,l | s,k,r | s,k,w

CvCluster → [MidC] V [CvCluster]

MidC → MidC41 | MidC31 | MidC32 | MidC20 | MidC21 | C

MidC41 → C, s, C, C

MidC31 → s, C, C | C, s, C | Nas, Fri, Lqd | Nas, Stp, Gld, | Nas, Obs, r | Lqd, Fri, Lqd | Lqd, Obs, r | Gld, Fri, Lqd | Gld, Obs, r | Stp, Stp, Lqd | Stp, Stp, Gld | Stp, Fri, Lqd | Fri, Stp, Lqd | Fri, Stp, Gld

MidC32 → Nas, Stp, Lqd | Nas, Stp, Nas | Nas, Stp, Fri | Nas, Stp, Stp | Nas, Stp, Afr | Lqd, Fri, Stp | Lqd, Fri, Nas | Lqd, Fri, Fri | Lqd, Stp, Stp | Lqd, Stp, Lqd | Lqd, Stp, Fri | Lqd, Stp, Gld | Lqd, Stp, Afr | Fri, Fri, hh | r, C, C

MidC20 → p,l | p,r | p,w | p,y | b,l | b,r | b,w | b,y | t,r | t,w | t,y | d,r | d,w | d,y | k,l | k,r | k,w | k,y | g,l | g,r | g,w | g ,y | f,l | f,r | f,y | v,l | v,r | v, y | th,r | th,w | th, y | s,p | s,t | s,k | s,f | s,m | s,n | s,l | s,w | s, y | sh,p | sh,m | sh,l | sh,r | sh,w | hh,y | hh,w | m,y | n,y | l,y

MidC21 → C, C

Final-Coda → C | p, th | t, th | d, th | d,s,t

| k,s | k,t | k,s,th | g, d | g , z | ch, t | jh, d | f, t | f, th | s, p | s, t | s, k | z, d | m, p | m, f | n, t | n, d | n, ch | n, jh | n, th | n, s | n, z | ng, k | ng, th | ng , z | l, p | l, b | l, t | l, d | l, k | l, ch | l , jh | l, f | l, v | l, th | l, s | l, z | l , sh | l, m | l, n | l,p | l,k,s | l,f,th | r, Stp | r,ch | r,jh | r,f | r,v | r,th | r,s | r,z | r,sh | r,m | r,n | r,l

The phoneme type categories are C (consonants), V (vowels), Nas (nasals), Gld (glides), Fri (fricatives), Afr (affricates), Obs (obstruents), Stp (stops), Lqd (liquids). The MidC categories are used for assigning word-internal consonant clusters to either the previous syllable's coda or the next syllable's onset according to one of the following four possibilities:

MidC41 – 1 coda consonants, 3 onset consonants,

MidC31 – 1 coda consonants, 2 onset consonants,

MidC32 – 2 coda consonants, 1 onset consonants,

MidC20 – 0 coda consonants, 2 onset consonants,

MidC21 – 1 coda consonants, 1 onset consonants.

This grammar in its current state is not fully deterministic. The fifth rule of MidC31 and the first rule of MidC32, for example, can result in ambiguous analyses for certain input sequences. For example, the phoneme sequence in *Andrew* can be parsed either by rule MidC31 (Nas Obs r) or by rule MidC32 (Nas Stp Lqd). How to deal with this problem is a practical issue. For parsing a large number of words automatically, our solution is to use this parser to first parse a pronunciation dictionary and then resolve the ambiguities through hand checking. The parsed pronunciation dictionary is then used to provide syllable structures of the words. We carried out this procedure on the TIMIT pronunciation dictionary. The results showed that the rules are a fairly precise model of English syllable and phonotactic structures: Out of 7905 pronunciations, only 135 or 1.7% generated multiple parses. The ambiguities were hand-checked and the parsed dictionary was used for transferring phoneme sequences into syllable structures in our TIMIT-based experiments.

As an illustration, Figure 7 shows the parse tree for word *display*, which denotes that word *display* consists of 2 syllables. The category 'CvCluster' is used for dealing with multiple syllables recursively; 'MidC' and 'MidC31' are categories of intervocalic consonant clusters. The category 'Init-Onset' denotes the word-initial syllable onset. The separation of syllable-internal consonants into coda and onset is based on the consonant types according to phonotactic principles [12].

PLACE Figures 7 and 8 AROUND HERE

The parser output of an unambiguous tree is transformed into subsegmental feature vectors [1, 4, 9] with high-level linguistic information. This is illustrated in Figure 8. Here the word *display* is parsed as a single word utterance with **ub** standing for utterance beginning and **ue** for utterance end. Stress is denoted by 0 (unstressed syllable) and 1 (stressed syllable) at the syllable node. The subsegmental feature structure is viewed as an autosegmental structure [1, 11] with skeletal and articulatory feature tiers and a prosodic structure placed on top of it. There is a resyllabification by which /s/ is moved to the second syllable. Currently this is done in the lexicon on the word by word basis.

Feature overlapping is carried out by the phonological rules we have implemented computationally, incorporating high-level linguistic information. We have used a temporal feature logic [1] as the theoretical framework for imposing constraints and in the formulation of the phonological rules.

Appendix 2. A Temporal Feature Logic

A temporal feature logic for the 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.

The prosodic structure

1. $\forall xy,\ syl(x) \wedge x\, \delta\, y \rightarrow sylconst(y) \vee boundary(y)$

   Syllables can dominate syllable constituents and boundaries.

2. $\forall xy,\ sylconst(x) \wedge x\, \delta\, y \rightarrow seg(y) \vee stress(y)$

   Syllable constituents can dominate segments and stresses.

3. $\forall x, boundary(x) \rightarrow x \in \{ub, ue, wb, we, mb, me\}$

   where the boundary symbols stand for utterance beginning, utterance end, word beginning, word end, morpheme beginning, morpheme end, respectively.

4. $\forall x,\ sylconst(x) \rightarrow x \in \{onset, nucleus, coda\}$

5. $\forall x,\ stress(x) \rightarrow x \in \{0, 1\}$

The tier structure

1. $\forall x,\ seg(x) \rightarrow \exists y,\ x\,\delta\,y \wedge articulator(y)$

   Every segment dominates one or more articulators.

2. $\forall x,\ articulator(x) \rightarrow \exists y,\ x\,\delta\,y \wedge feature(y)$

   Every articulator dominates one or more features.

3. $\forall x,\ articulator(x) \ \rightarrow\ x \ \in\ \{lip, tbld, tdsm, vel, lyx\}$ where the articulator symbols stands for lip, tongue-blade, tongue-dorsum, velum and larynx, respectively.

4. $\forall x,\ feature(x) \rightarrow poa(x) \vee cdg(x) \vee shape(x)$ where *poa* stands for place of articulation; *cdg* stands for constriction degree and *shape* stands for the shape of the lips. (Figure 8 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]. When some B is a component of some A, we say A dominates B, or A δ B. When two events A and B overlap in time, we denote this by A $\circ$ B; otherwise, either A precedes B or B precedes A: A $\prec$ B $\vee$ B $\prec$ A. In Bird's temporal feature logic, dominance implies overlap. This is called *the locality constraint*:

$$\forall xy,\ x\,\delta\,y \rightarrow x \circ y$$

Precedence, on the other hand, implies no overlap and vice versa. This is described as *the mutual exclusion of* $\prec$ *and* $\circ$:

$$\forall xy,\ x \prec y \rightarrow \neg\, x \circ y \text{ and}$$

$$\forall xy,\ x \circ y \rightarrow \neg\, x \prec y$$

There is an important property related to the above two fundamental properties which is called *the transitivity of* $\prec$ *through* $\circ$: $\forall wxyz,\ w \prec x \wedge x \circ y \wedge y \prec z \rightarrow w \prec z$. Logically it is hard to prove this. However, this property can be visualized in the following way. To see that $w \prec z$, we note that the left boundary of x is to the right of w and the right boundary of y is also to the right of w (since $x \circ y$) and the left boundary of z is to the right of the right boundary of y, therefore, the left boundary of z is also to the right of w. This situation can be illustrated by the following diagram:

```
    w       x
|-------|------|
         y     z
      |----|-------|
```

Referring to Figure 8, these operators can be illustrated in the following:

$$Syllable_1\ \delta\ Nucleus_1\ \delta\ \text{/ih/}\ \delta\ Larynx\ \delta\ \text{V+},$$

$$Syllable_1 \circ Nucleus_1 \circ \text{/ih/} \circ Larynx \circ \text{V+},$$

$$/\mathrm{d}/ \prec /\mathrm{ih}/ \prec /\mathrm{s}/ \prec /\mathrm{p}/ \prec /\mathrm{l}/ \prec /\mathrm{ey}/.$$

Feature overlapping in this temporal logic framework is defined as a process of dynamic realization of segmental tier structure synchrony (i.e. articulatory features are synchronous within segments). In dynamic realization, a feature on some articulator tier may spread temporally and may overlap with features of neighboring segments. If we use a predicate "*possib*" to denote possibility of realizing a planned segment sequence, this can be expressed as follows:

$$seg_a \circ articulator_p \circ feature_l \wedge seg_a \prec seg_b \rightarrow possib(seg_b \circ feature_l)$$

When $feature_l$ overlaps with seg_b, it has a chance to overlap with the features dominated by seg_b.

We abandon the "linearity constraint" which requires that events of the same sort be in precedence relation only (i.e. features on the same articulator tier can only be in precedence relation). Instead, we allow features at the same tier (therefore, of same sort) to overlap and we call this "feature mixing", denoted by $\bowtie$. This relation is expressed as

$$\forall xy, tier_i(x) \wedge tier_i(y) \wedge x \circ y \leftrightarrow x \bowtie y.$$

That is, if events on the same tier are in the same dominant group and overlap in time, they are said to mix with each other. As an overlap can be either partial or full, so is the mix relation. This mix relation is used to describe coarticulation involving the same tier in the articulatory feature space (i.e., *co-production*). Articulatory mix is a very general phenomenon. Whenever consecutive phonemes involve the same articulator, there is often a feature mix in the transition phase.

The temporal feature logic described above is motivated by empirical observations of speech data including articulatory data and speech spectrograms. In particular, the subsegmental, articulatory features are components in forming a phoneme, and at the same time these features can spread beyond the conventional boundaries of phonemes, exerting influence on the articulatory or acoustic properties of neighboring phonemes up to some distance away. If we consider such spreading as independent events, these events can take the form of temporal overlap or temporal mix. Overlap refers to simultaneous events occurring at different tiers, while mix refers to simultaneous events occurring on the same tier. A combination of overlapping and mixing accounts for a great part of transitions between phonemes. In descriptive terms, these transitional phases in speech can be modeled by a set of feature bundles constructed from interactions between phonemes via the mechanisms of overlap and mix. These overlapped and mixed feature bundles derived from the pre-defined, canonical, context independent feature bundles are then taken as the basic units of speech to form the HMM state topology for speech recognition.

In Figure 9, the spectrogram shows some acoustic properties of the utterance "step in" which can be described by feature overlapping in accounting for the transitional phases. The vowel /eh/ in word *step* contains two transitional phases. One has a carry-over *tongue-tip* feature spread from the previous phoneme /t/; this overlapped feature from /t/ to /eh/

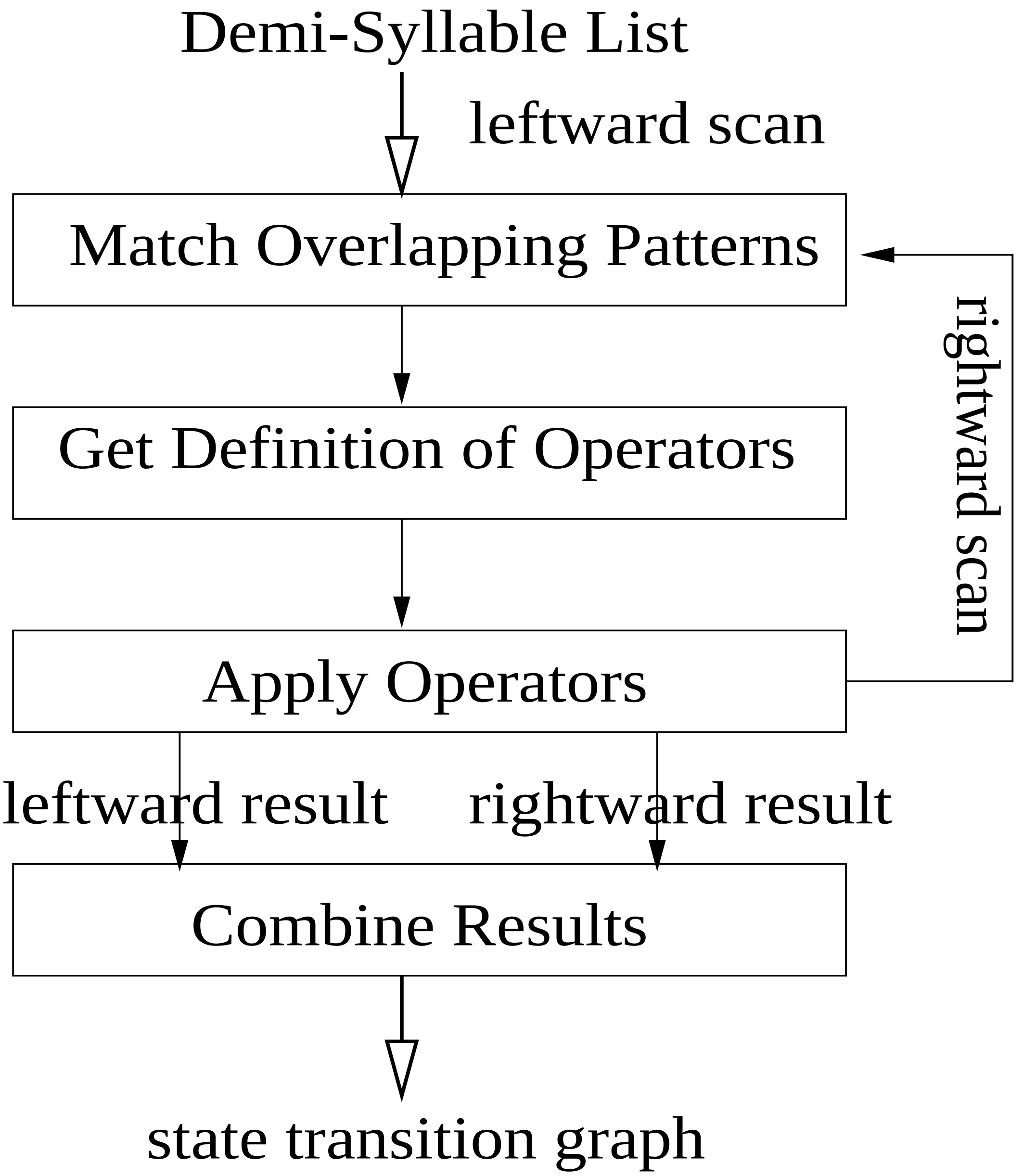
accounts for the initial formant transition in /eh/. The second transitional phase contains a look-ahead *lips* feature spread from the following phoneme /p/. The acoustic effect is a conspicuous formant transition over a major length of /eh/ with the formant transition targets towards those of /p/. The stop /p/, due to its coda position, has a very weak release phase. The spectral shape of the release burst in /p/ is affected by the look-ahead *tongue-dorsum* feature of the following vowel /ih/. The vowel /ih/ is partially nasalized due to the *velum* feature spread from the following phoneme /n/. At the same time, both the *lips* feature of /p/ and the *tongue-tip* feature of /n/ are overlapped into /ih/, creating the obvious formant transition throughout the entire vowel.

PLACE Figure 9 AROUND HERE

Appendix 3. Abbreviations Used In Figures

ASP	aspiration
B.e	tongue-blade of /e/
B.l	tongue-blade of /l/
C/V	consonant or vowel
ClsAlv.Br	(feature mix of) closure-alveolar and tongue blade of /r/
ClsAlv	closure-alveolar
ClsLab	closure-labial
CrtAlv	critical-alveolar
CvCluster	consonant vowel cluster
Fri	fricative (consonant)
Init-Onset	word initial onset
MidC	word-middle consonant
Nas	nasal (feature)
Rnd.o	lip-rounding of /o/
Rnd.r	lip-rounding of /r/
seg	segment
Stp	stop (consonant)
syl	syllable
T.Blade	tongue blade

T.Dorsum	tongue dorsum
ub	utterance begin
ue	utterance end
V-	unvoiced
V+	voiced



Demi-Syllable List
leftward scan
Match Overlapping Patterns
rightward scan
Get Definition of Operators
Apply Operators
leftward result
rightward result
Combine Results
state transition graph

Figure 1:

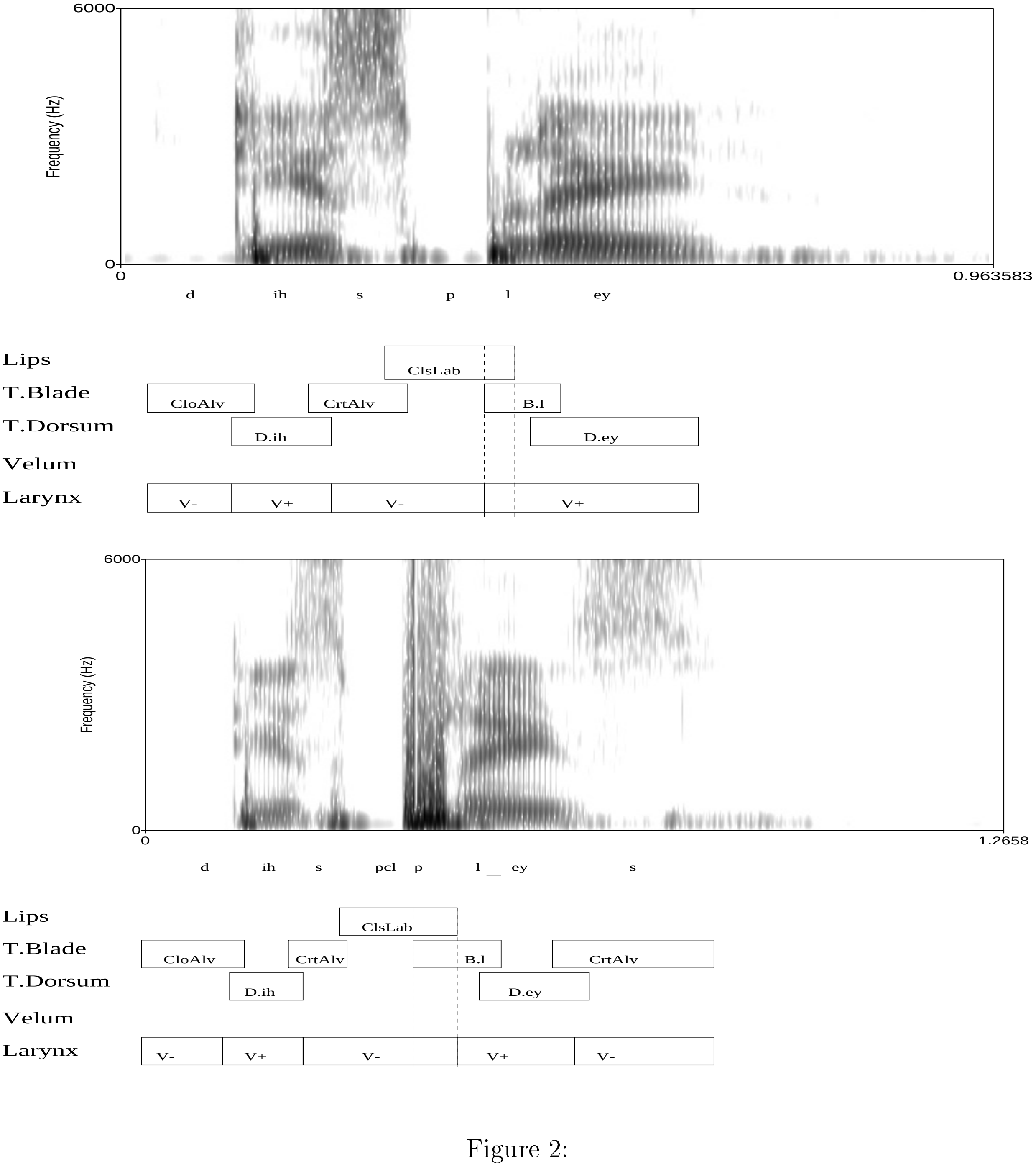

6000
Frequency (Hz)
0
0
0.963583
d
ih
s
p
l
ey
Lips
T.Blade
T.Dorsum
Velum
Larynx
ClsLab
CloAlv
CrtAlv
B.l
D.ih
D.ey
V-
V+
V-
V+
6000
Frequency (Hz)
0
0
1.2658
d
ih
s
pcl
p
l
ey
s
Lips
T.Blade
T.Dorsum
Velum
Larynx
ClsLab
CloAlv
CrtAlv
B.l
CrtAlv
D.ih
D.ey
V-
V+
V-
V+
V-

Figure 2:

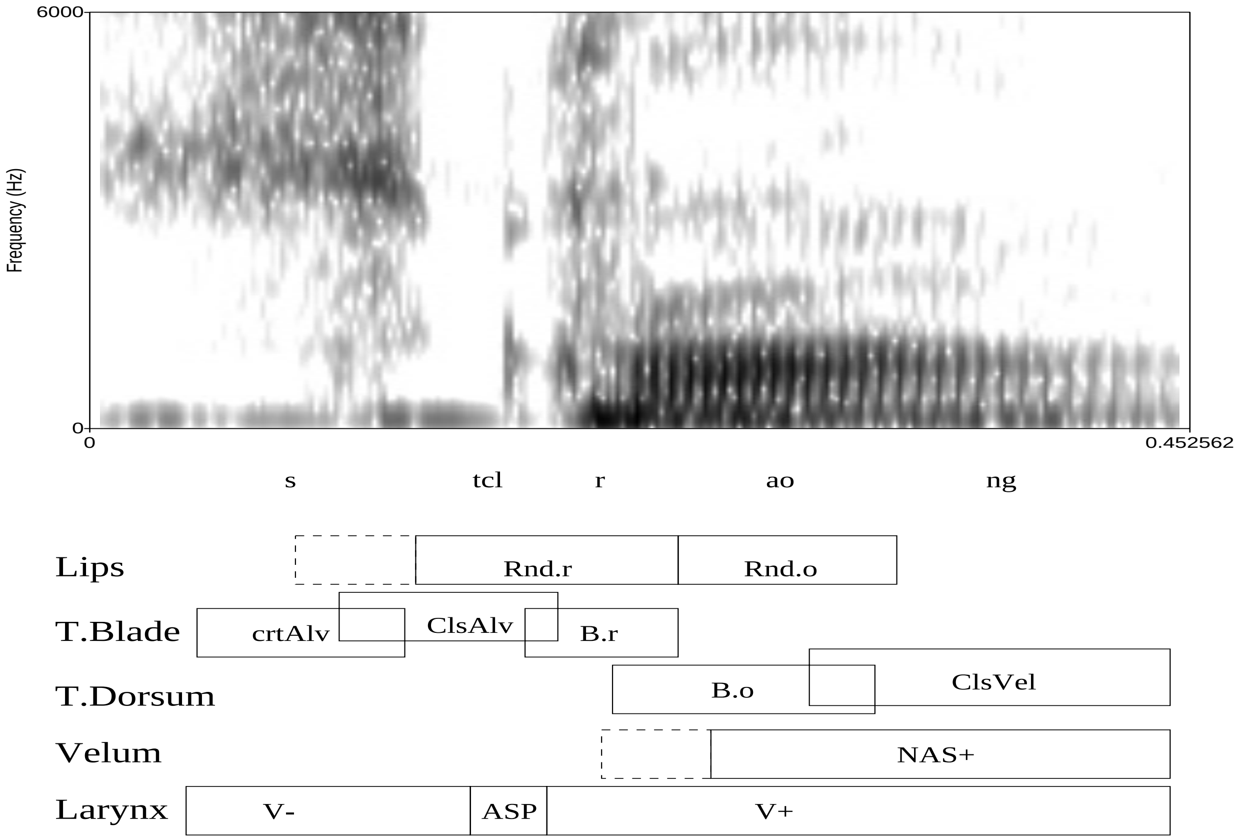

6000
Frequency (Hz)
0
0
0.452562
s
tcl
r
ao
ng
Lips
Rnd.r
Rnd.o
T.Blade
crtAlv
ClsAlv
B.r
T.Dorsum
B.o
ClsVel
Velum
NAS+
Larynx
V-
ASP
V+

Figure 3:

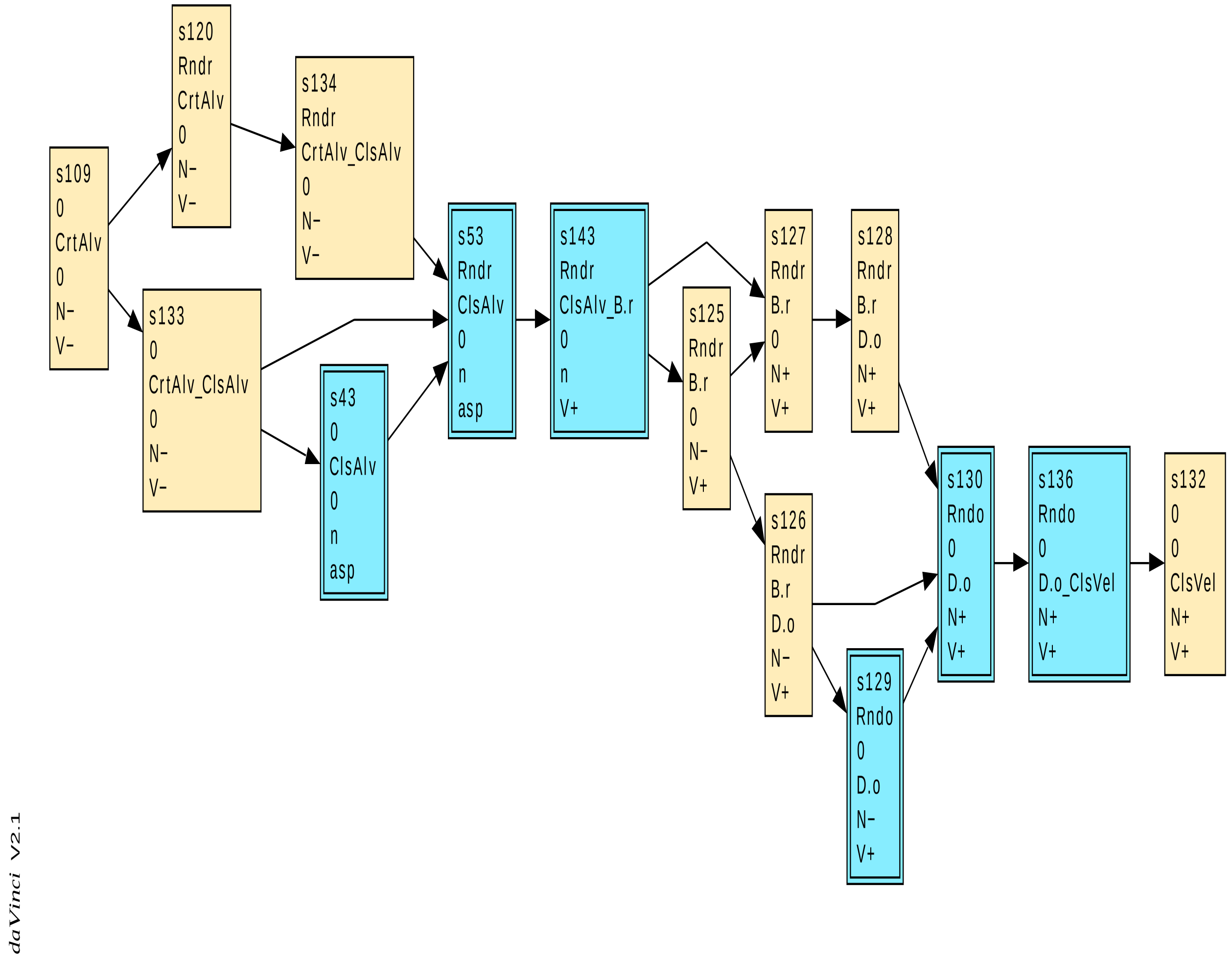

s109
0
CrtAlv
0
N-
V-
s120
Rndr
CrtAlv
0
N-
V-
s133
0
CrtAlv_ClsAlv
0
N-
V-
s134
Rndr
CrtAlv_ClsAlv
0
N-
V-
s43
0
ClsAlv
0
n
asp
s53
Rndr
ClsAlv
0
n
asp
s143
Rndr
ClsAlv_B.r
0
n
V+
s125
Rndr
B.r
0
N-
V+
s127
Rndr
B.r
0
N+
V+
s128
Rndr
B.r
D.o
N+
V+
s126
Rndr
B.r
D.o
N-
V+
s129
Rndo
0
D.o
N-
V+
s130
Rndo
0
D.o
N+
V+
s136
Rndo
0
D.o_ClsVel
N+
V+
s132
0
0
ClsVel
N+
V+
daVinci V2.1

Figure 4:

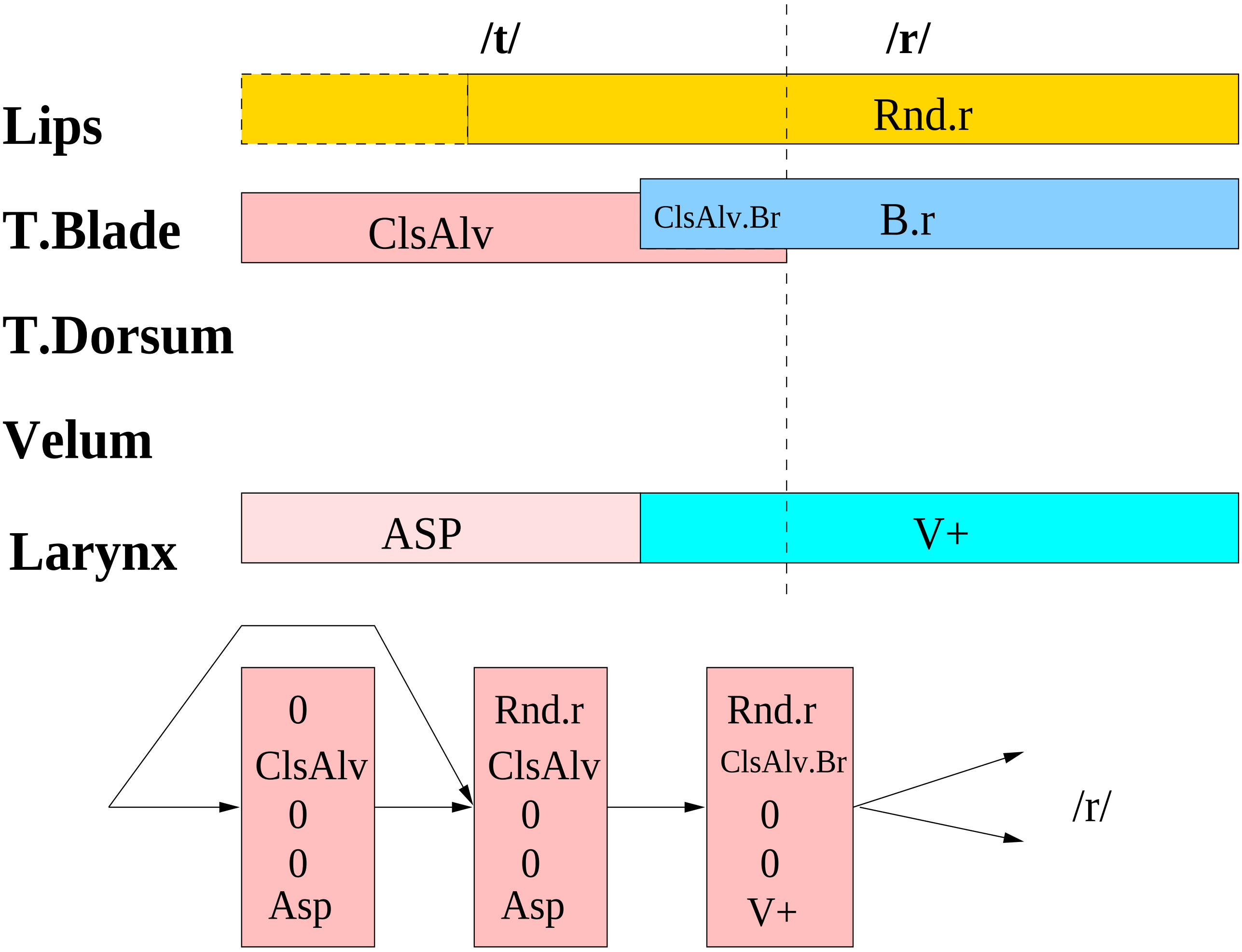

/t/
/r/
Lips
Rnd.r
T.Blade
ClsAlv
ClsAlv.Br
B.r
T.Dorsum
Velum
Larynx
ASP
V+
0
ClsAlv
0
0
Asp
Rnd.r
ClsAlv
0
0
Asp
Rnd.r
ClsAlv.Br
0
0
V+
/r/

Figure 5:

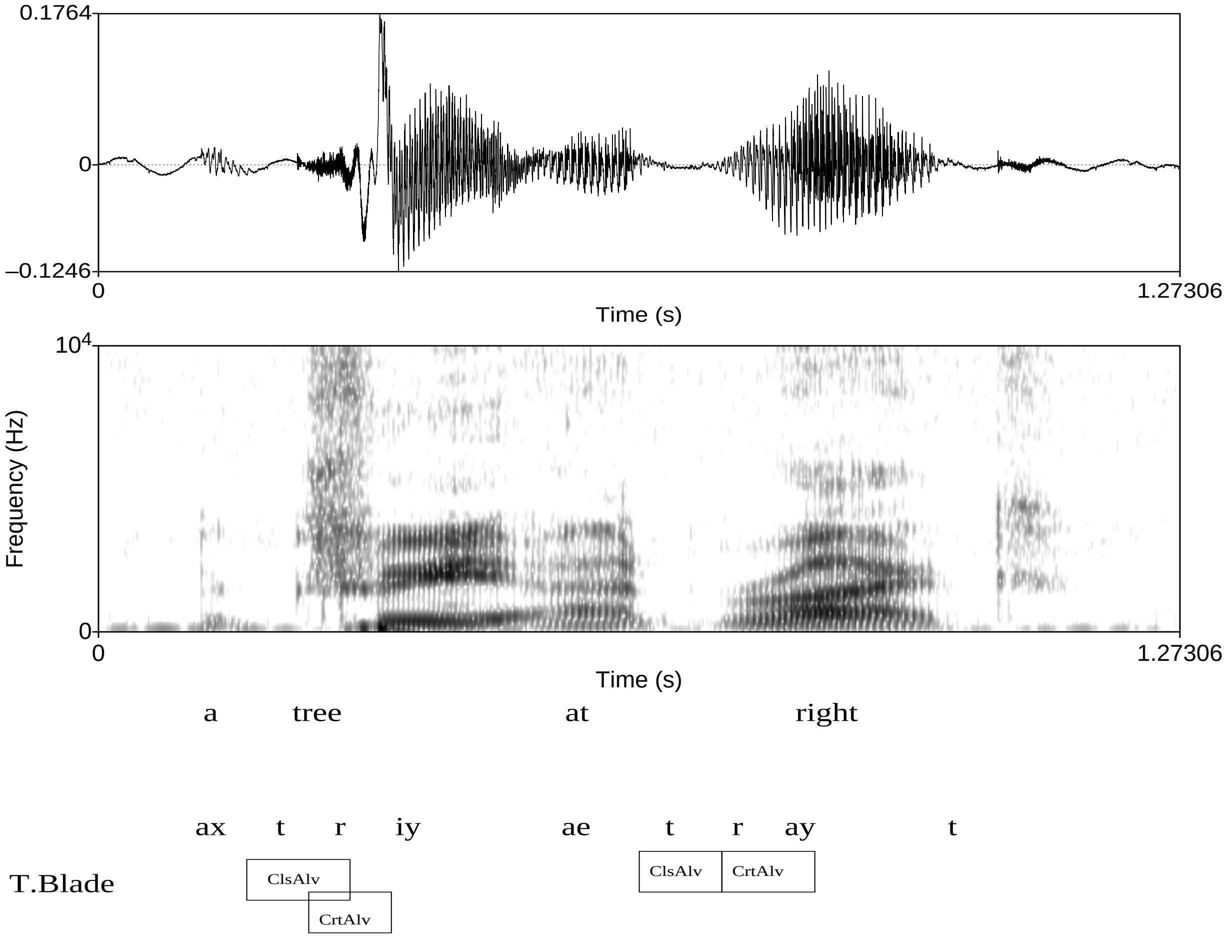

0.1764
0
–0.1246
0
1.27306
Time (s)
10^4
Frequency (Hz)
0
0
1.27306
Time (s)
a
tree
at
right
ax
t
r
iy
ae
t
r
ay
t
T.Blade
ClsAlv
CrtAlv
ClsAlv
CrtAlv

Figure 6:

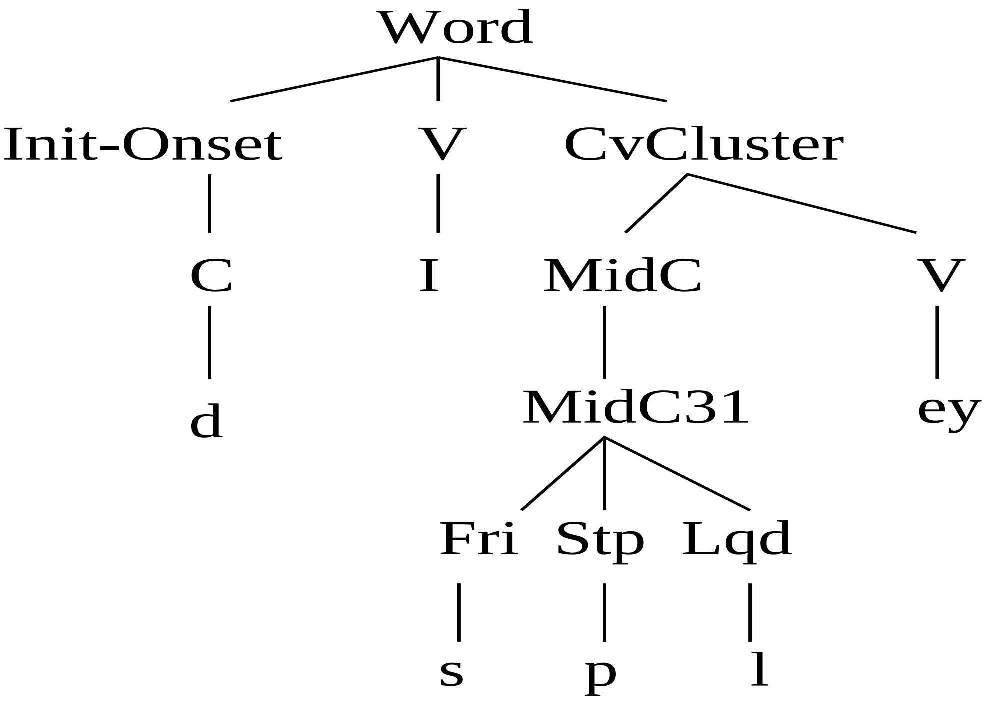

Word
Init-Onset
V
CvCluster
C
I
MidC
V
d
MidC31
ey
Fri
Stp
Lqd
s
p
l

Figure 7:

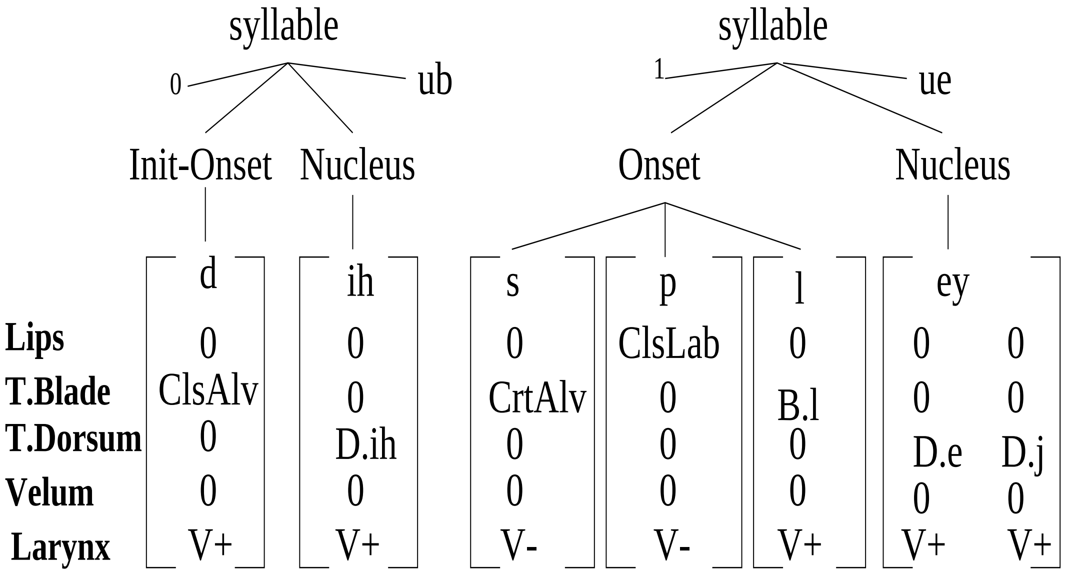

syllable
0
ub
Init-Onset
Nucleus
syllable
1
ue
Onset
Nucleus
d
ih
s
p
l
ey
Lips
T.Blade
T.Dorsum
Velum
Larynx
0
ClsAlv
0
0
V+
0
0
D.ih
0
V+
0
CrtAlv
0
0
V-
ClsLab
0
0
0
V-
0
B.l
0
0
V+
0
0
D.e
0
V+
0
0
D.j
0
V+

Figure 8:

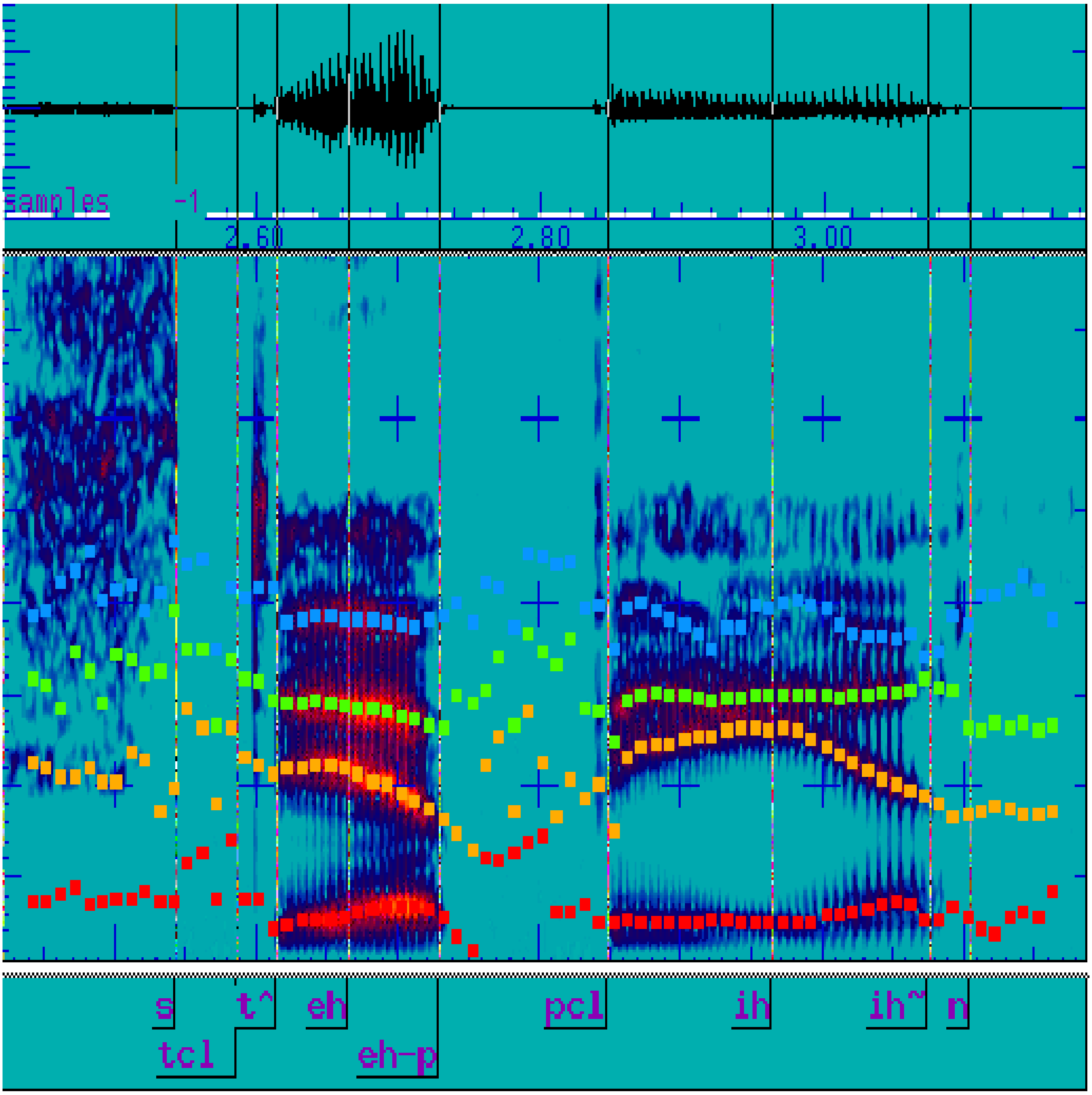


Figure 9:

Speech Recognition Decoding Results								
Dial.Reg.	No.Sents	No.Words	Corr%	Sub	Del	Ins	WER%	Sent.ER%
1	110	964	89	48	55	10	12	36
2	260	2281	92	97	71	17	9	33
3	260	2271	92	94	77	20	9	31
4	320	2714	90	141	114	21	11	33
5	280	2438	88	175	116	25	13	40
6	110	966	91	51	27	14	10	35
7	230	1967	91	107	62	9	10	33
8	110	956	91	61	23	2	10	29
Total/Ave.	1680	14557	90	752	567	120	10	34

Table 1:

Phone Recognition Decoding Results		
System	Correct %	Accuracy %
Triphone (baseline)	73.90	70.86
Overlapping Feature	74.70	72.95

Table 2:

Speech Recognition Decoding Results: Word Correction and Accuracy						
Dial.Reg.	No.Sents	No.Words	Baseline Corr	Baseline Acc	Feature Corr	Feature Acc
1	110	964	81.98	80.42	82.92	81.88
2	260	2281	86.10	85.44	86.58	85.62
3	260	2271	85.78	85.16	86.35	85.78
4	320	2714	83.05	81.80	83.35	82.09
5	280	2438	80.31	79.16	80.84	79.61
6	110	966	85.92	85.51	85.20	84.47
7	230	1967	88.66	87.90	89.22	88.66
8	110	956	86.61	85.67	87.13	86.51
Total/Ave.	1680	14557	84.61	83.69	85.04	84.13

Table 3:

Figure and Table Captions

Figure 1: The overlapping feature bundle generator.

Figure 2: Feature overlaps for words *display* (upper panel) and *displace*(lower panel).

Figure 3: Feature overlaps and mixes for word *strong*.

Figure 4: State-transitional graph for word *strong*.

Figure 5: Results of applying operator Op125@15 to /t/ before /r/ and the corresponding state transition graph of /t/.

Figure 6: Use of phonological rules and high-level linguistic information.

Figure 7: Parse tree for word *display*.

Figure 8: Subsegmental feature structure for word *display*.

Figure 9: An example spectrogram for *step in* illustrating acoustic properties associated with feature overlaps.

Table 1: Continuous speech recognizer performance on words and tested on all 1680 sentences in the TIMIT testing set. Feature-based phonological model is used to construct the word-level HMM's and bigram language model is used for word recognition. Each feature-defined HMM state was trained with a five-Gaussian mixture using HTK.

Table 2: TIMIT phone recognition results: Triphone baseline versus feature-overlapping

model. The latter uses the feature overlapping rules in the decision-tree based state tying process of phone-level HMM's.

Table 3: TIMIT word recognition results: triphone baseline versus feature-overlapping model. The latter uses the feature overlapping rules to construct context-dependent phone-level HMM's incorporating high-level linguistic constraints.