In: Proceedings of ICASSP-90. Albequeque, NM, 1990.

S14.9

IDENTIFICATION OF CONTEXTUAL FACTORS FOR PRONUNCIATION NETWORKS

Francine R. Chen
XEROX PALO ALTO RESEARCH CENTER
3333 Coyote Hill Road
Palo Alto, CA 94304

Abstract

In this paper we present a data-intensive, semi-automatic method for identifying subsets of contextual factors which are useful for predicting the allophonic realizations of dictionary phonemes. The method organizes contextual descriptions of phonological variation into context trees. Context trees are computed using a combination of decision tree induction for factor selection, and hierarchical clustering for forming natural groups of factor values. We describe how the resulting context trees can be used to provide allophones in creating pronunciation networks. We use a phoneme level representation with a flexible context description which allows modeling of effects extending across syllables and word-boundaries.

1. Introduction

The context in which a phoneme occurs leads to consistent differences in how it is pronounced. Phonologists employ a variety of contextual descriptors to explain phonological variation. These descriptors are theoretically motivated by studies of different languages and are comprised of many factors, such as stress and syllable part. However, in current speech recognition systems, only a few contextual descriptors are employed when developing pronunciation networks. In these systems, generally the effects of only the preceding and following phones, as in triphone models, or implicit within-word contextual effects, as in whole word models, are captured.

The limited use of context in recognition systems is partially due to the amount of data needed to train the network units in which many contextual factors are represented. The units used in current systems include whole word models, where phones are represented in the context of the word in which they occur (*e.g.*, [10]), generalized triphones [9], and a hierarchy from words to subsets of triphones (*e.g.*, [3]). Whole word models provide the most complete context of the internal phones, but usually do not model word boundary effects well. A subword unit, such as the triphone, can be concatenated into word models to simplify additions to the lexicon; however, triphones account for only a subset of contextual factors.

The use of a wide variety of contextual factors allows better predictions of pronunciation variants, which in turn, can result in better performance of speech recognition systems. A case in point is the work at SRI [4] [13] where speech recognition performance was improved through the use of hand-derived phonological rules sets which were refined using software tools and measures of coverage and overcoverage.

In this paper, we present a data-intensive procedure for systematically selecting contextual factors to describe phonological variants, producing a "mixed" context representation. We then describe how the mixed factor representation can then be used in creating pronunciation networks.

contextual factor	values
preceding phoneme	(all phonemes) + SB
following phoneme	(all phonemes) + SB
preceding phone	(all phones) + deletion + SB
following phone	(all phones) + deletion + SB
syllable part	onset, nucleus, coda
stress	primary, secondary, unstressed
syllable boundary type	initial, final, internal, initial-and-final
foot boundary type	initial, final, internal, initial-and-final
word boundary type	initial, final, internal, initial-and-final
cluster type	onset, coda, nil
open syllable?	true, false
true vowel?	true, false
function word?	true, false

Table 1: Contextual factors used in pronunciation experiments (SB represents sentence boundary)

2. Contextual Factors

The context of a phoneme can be described using many types of theoretically-motivated, linguistically-based *factors*, such as **stress** and **word boundary**. Each contextual factor describing the context of a phoneme has a *value*. For example, the factor **stress** may take on any one value of **primary**, **secondary**, or **unstressed**. The set of factors and corresponding factor values used in this work are listed in Table 1. Some of the factors are units normally associated with several phonemes. For example, the factor **syllable part** may take on the value **onset**, and may be composed of up to three phonemes. In such cases, we assign the value of a factor to each phoneme within the unit. Thus, the exemplars /s/, /t/, and /r/ in an /str/ sequence would each be assigned a **syllable part** value of **onset**. Each factor represents a separate dimension, and a factor ignores units which are irrelevant to it. Hence, a contextual factor such as **preceding phoneme** extends across word boundaries.

These lexical contextual factors are used to describe the context of the mapping between a dictionary phoneme and its realization in a hand-transcribed segment. The mapping exemplars were derived by automatically aligning the almost 30,000 hand-transcribed segments in approximately 900 of the "sx" sentences from the TIMIT database [7] to dictionary baseforms. We use the same dictionary, the *X-Dictionary** developed at Xerox PARC, both for creating the training exemplars and later for network creation.

3. Context Trees

If the context of a phoneme is described by simultaneously using all the contextual factors listed in Table 1, a prohibitive amount

*The *X-Dictionary* has been checked for consistency and has been augmented from entries in standard dictionaries to include foot boundary indicators.

CH2847-2/90/0000-0753 $1.00 © 1990 IEEE

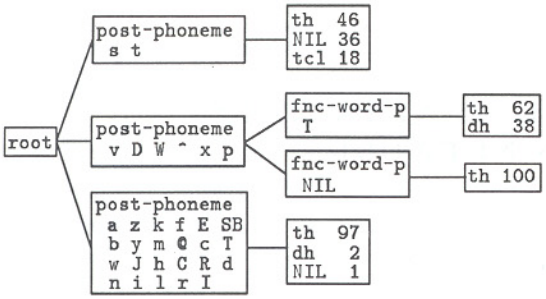

Figure 1: Pruned context tree for /th/

of data would be required to form an adequate description of each phoneme in each context. One way to handle this difficulty is to build a *context tree*, in which a subset of contextual factors is selected using a combination of decision tree induction for selecting factors and clustering for grouping factor values. In this method, the number of leaves and branching of the tree are data-dependent. An alternate method for grouping contextual factors, based on the creation of a binary tree with a preset number of leaves, is given by Sagayama [12].

A context tree describing the realizations of /th/ is shown in Figure 1. The nodes of a context tree represent the values of a particular contextual factor. Nodes with the same parent represent the same contextual factor, but different values of that factor. In Figure 1, the first child of the root node corresponds to the contextual factor post-phoneme with the values /s/ or /t/. The node representing the second child also corresponds to the factor post-phoneme but with the mutually exclusive values /v/ /D/ /W/ /^/ /x/ /p/. Each leaf of a context tree encodes the distribution of allophones in the context defined by the factor values encountered in traversing the tree from the root node to reach the leaf. In general, more than one allophone occurs in a context because phoneme realizations are not deterministic. For example, the top leaf in Figure 1 indicates that /th/ is pronounced as [th] when followed by an /s/ or /t/ 46%, is deleted 36%, and is realized as a closure 18% of the time.

To induce a context tree, the data in a node is recursively split into new nodes. At each node, a contextual factor is selected for the next split. Prior to each factor selection, we first cluster the values of a factor based on their similarity in influencing a phoneme's realization. In the next two sections we describe the algorithm for context tree creation.

3.1 Selection of Contextual Factors

Contextual factors are selected using a greedy algorithm which minimizes the loss of information at each selection, adapted from the decision tree induction methods of [1] and [11]. At each node, the contextual factor is selected which has values that best separate the realizations of the data, (*i.e.*, make each node "purer"†). The exemplars in the current node are subdivided according to each exemplar's value of the selected factor, creating a new set of nodes.

Since a phoneme may be realized as multiple allophones, we used a reduction of entropy criterion for measuring how well a selected factor separates the allophones in the data. As entropy is reduced in each level of a tree, the nodes of the tree become purer and the different realizations are better separated. We briefly review the criterion calculations here (adapted from [1] [2] [11]).

Before splitting, the entropy at a node based on the classes X is $H(X)$. The average entropy at the new nodes created by splitting on the values V of a given contextual factor is: $E(H(X|v)) = \sum_v P(v)H(X|v) = H(X|V)$ where $\{v\}$ represents all possible values of a contextual factor. The gain for factor a, $G(a)$, is the difference between the entropy before splitting, $H(X)$, and the conditional entropy after splitting $H(X|V)$: $G(a) = H(X) - H(X|V) = I(X;V)$, where $I(X;V)$ is the mutual information between X and V. To normalize for the variable number of factor values, we compute the *gain ratio*, $R(a)$, which is the gain for factor a normalized by the entropy of the number of values associated with a: $R(a) = G(a)/H(V)$. The factor which maximizes $R(a)$ is selected for splitting.

An advantage of this methodology over individual rule creation is that the usefulness of each attribute for *globally* separating the different realizations is considered as the trees are constructed. Additionally, a context tree partitions the space of contexts into mutually exclusive subspaces, permitting direct estimation of allophone probabilities.

3.2 Clustering of Contextual Factor Values

Traditionally, in tree induction, nodes are split either along all values of a factor (*e.g.*, [11]) or else binary splits are used (*e.g.*, [1]). In speech, some factors have many values (*e.g.*, in our model, preceding-phoneme has 46 values) and some of the factor values are similar in their influence on phoneme realizations. Hence, rather than splitting on all values or performing binary splits, it would be desirable to group factor values with similar effects. One could precluster the values according to theoretical ideas of what is similar, but the groupings may change depending on context. For example, the values of the factor following-phoneme could be grouped based on manner of articulation. Such a grouping is useful when predicting whether or not a plosive will be aspirated. However, such a grouping is not useful in predicting, say, when /s/ will be palatalized. Alternatively, the values could be clustered into a predefined number of groups at each node [2]. However, the appropriate number of groups is not the same for all sounds and again may depend on the current context.

We use hierarchical clustering to create clusters for each set of factor values in which the number and type are determined from the data, rather than predefined. Mutual information is used as the distance metric and is computed as in [5]. That is, let the average mutual information between context value v_i and the allophones X be: $I(v_i;X) = \sum_x P(v_i,x) log_2 \frac{P(v_i|x)}{P(v_i)}$, where $\{x\}$ represents all possible allophones. The increase in average mutual information resulting from pairing two factor values v_m and v_n is the difference between the average mutual information resulting from pairing v_m and v_n, $I(v_m \cup v_n;X)$, and the contribution to the average mutual information before pairing v_m and v_n, $I(v_m;X) + I(v_n;X)$; thus $\triangle I(V;X) = I(v_m \cup v_n;X) - I(v_m;X) - I(v_n;X)$.

At each iteration, the pairing that results in the largest increase in mutual information is selected and forms a new cluster. Pairing of clusters is continued until either only two clusters are left, the decrease in mutual information more than doubles from one iteration to the next, or the mutual information decreases more than a threshold, which we empirically set at -30. The conditions for stopping define when the loss in mutual information is too great to continue clustering.

Since we cluster the values of each factor prior to splitting, it may be useful to split on a factor multiple times, each time under a more specific context (*i.e.*, farther down the tree). Thus, after a factor is selected for splitting, it is *not* removed from the set of factors considered.

3.3 Creation of Robust Trees

A tree that has been constructed by the combined tree induction and clustering of factor values may be too specialized to the train-

†A *pure* node contains exemplars of only one realization type.

ing exemplars. To create a more robust tree, we prune the branches and remove unlikely leaf values.

Many methods of pruning have been suggested (*e.g.*, [1] [2] [11]). We employ two types of pruning to retain the parts of the tree which will be robust to new data. During tree creation, nodes are extended only when the number of exemplars is greater than a specified threshold [1]; we used 20. In addition, only nodes relevant to the classification of cross-validation exemplars are kept, as measured by a chi-square test [11] at the .01 level of significance. Trees were induced using 60% of the data and pruned on the remaining 40% of the data.

In creating pronunciation networks, it is hard to define an "optimum" number of pronunciations to represent. With only a few pronunciations, some variants may be poorly modeled in a speech recognizer. With many pronunciations, the amount of training data is sparse and unlikely pronunciations may confuse a recognizer.

With context trees, this problem can be handled at the phonemic level. Given a large data set, context trees tend to overgenerate pronunciations because each new allophonic realization of a phoneme in a context translates into another possible arc in a network. But because context trees contain count information on allophones in context, unlikely allophones within a leaf can be systematically removed, based upon counts or percentages and the arcs representing the removed allophones are not created.

4. Pronunciation Network Creation

The allophones in the leaves of a context tree are described by a "mixed" set of contextual factors. The contextual descriptions provide a way to specify contexts for creating context models intermediate in the continuum from adjacent phone to whole word models. Rather than using a fixed, consistent set of contextual factors, the mixed context representation in the context trees can be used in pronunciation networks. The leaves of the trees represent a one-to-many mapping between a phoneme in a particular context and a set of allophones; these contexts limit the allophones which can be joined.

The mapping from a dictionary baseform to a set of possible pronunciations is characterized by the substitution, deletion, and insertion of sounds. One context tree is created for each of the 45 dictionary phonemes in the *X-Dictionary*. Each tree attempts to segregate all the allophonic realizations of a phoneme based on the different contextual factors. The data in each tree defines the set of deletions and substitutions, represented as allophones observed in each context, of a dictionary phoneme. A separate tree was created for each phoneme because, as in the example illustrating the grouping of following-phoneme to be different for plosives and /s/ palatalization, the same contextual factor can influence different phonemes in different ways.

In addition to modeling substitutions and deletions, pronunciation network creation also requires modeling of insertions. Insertions do not fit the substitution/deletion model since insertions may occur between any pair of phonemes. In addition, one must also model when insertions do not occur to allow prediction of the probability of an insertion in any context. These requirements are met by representing all insertions and non-insertions in one tree. In organizing the data to build an insertion tree, all pairs of phonemes in the training data are checked for whether or not an insertion occurred between them. The insertion tree thus predicts when insertions can occur as well as what type of insertion can occur in a particular context.

To build an insertion tree, the contextual factors describing the mappings are redefined to be a set applicable to insertions. Each of the factors in Table 1 below following-phone is replaced with contextual factors describing the phonemes adjacent to where

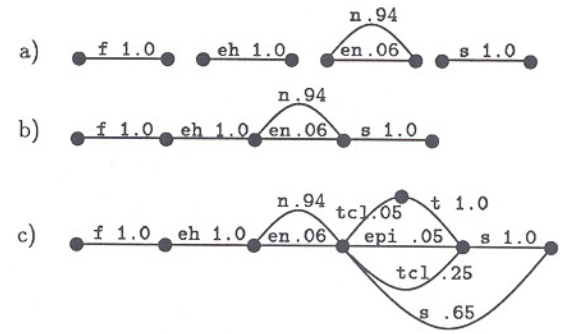

Figure 2: Pronunciation network for "fence": a) initial arcs b) arcs connected c) insertions added.

an insertion can occur. For example, the factor stress is replaced with stress of preceding phoneme and stress of following phoneme.

Networks can be created word by word and can be joined to produce a pronunciation network for a recognition system. To create a word network, a two-pass process is used. First, each dictionary "phoneme" in a word is mapped to the allophone distribution represented by the leaf in a context tree corresponding to the context in which the phoneme occurs. This produces a sequence of allophones representing the sequence of phonemes (see Figure 2a). Contextual constraints associated with the allophones from a leaf are matched to contextual constraints of adjacent allophones. If the phoneme is word-initial or word-final and the context at the word boundaries is not specified, then the allophones for each possible context must be incorporated into the network. Insertions are then added between the leaf values when the context for an insertion is compatible. Insertions are added after substitutions and deletions because the context in which an insertion occurs is dependent upon adjacent phones, which is determined by the phoneme realizations.

Using our method based on context trees, the pronunciation network produced for the word "fence" is shown in Figure 2. In creating this network, we made the simplification of not using the contextual factors describing adjacent phones for modeling substitutions and deletions. This produces the simple network in Figure 2b. Addition of insertions, in which we do include the contextual factors describing adjacent phones, produces the network shown in Figure 2c. In creating this network, we also assumed that the word was spoken in isolation and therefore preceded and followed by silence. Had we not done so, the boundaries of the word would be much more bushy with additional arcs representing the different possible allophones and probabilities in various contexts.

Because of limited training data, some of the words may contain a context value which has not been observed in the training data. However, each node of the tree contains the distribution of allophones for the partial context represented by the node. Thus, the allophones for unobserved contexts can be estimated from a partial context specification by tracing down the tree as far as consistent with the observed contextual factor values describing a phonemic baseform.

In tree induction a subset from a predetermined set of possible contexts which are good at differentiating among the realization distributions is selected. This subset is a larger number of contexts than the data would permit if the selected contexts were always considered together. Consequently, a larger overall number of contexts are used for describing the realizations. For example, in Figure 3, the three contexts of PRE-PHONEME, FNC-WORD-P, and FOOT-BDRY are used for describing the realizations of /y/, but only two contexts, either PRE-PHONEME and FNC-WORD-P or PRE-PHONEME and FOOT-BDRY, are used to describe each leaf. Thus,

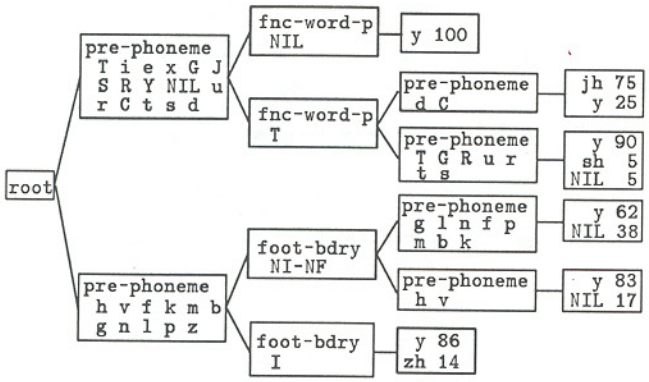

Figure 3: Pruned context tree for /y/

the effective number of contexts used to describe the conditions under which different variants of a phoneme occur is larger, given a limited amount of data.

5. Discussion

We informally examined the context categories formed by the clustering technique in which the number of groups was determined by the data. We observed that many times the values composing a category corresponded to a linguistic category. For example, one set of values of the factor PRE-PHONEME in a /b/ context tree was composed of the plosives {p t k b d g C J} and silence. And in Figure 3, the tree indicates that /y/ is often realized as [jh] when preceded by /d/ or /C/, as expected. These examples illustrate that in creating the context trees, both traditional and uncommon but expected linguistic categories are identified. However, we also noted that the categories sometimes contained unanticipated values. Examination showed that there generally were very few exemplars of these values, in agreement with [8].

The utility of contexts other than preceding and following phoneme was tested by examining the contexts in the top two levels of the 45 context trees which were created. In 22 trees (N a ^ x E I | U u y l r n s J C g d b k t p) preceding-phoneme and following-phoneme were included as factors in the top two levels. In nine trees (c W i h w m D Z S) only preceding-phoneme appeared in the top two levels, and in 12 trees only following-phoneme (L R @ Y e o O G v T f z) appeared in the top two levels. However, other factors also appeared in the top two levels, as well as in lower levels. The additional contextual factors which appeared and the number of times each appeared in the first two levels of the tree are: stress 15, function-word? 9, syllable-part 6, syllable-boundary-type 4, foot-boundary-type 2, cluster-type 2, and word-boundary-type 1. This data indicates that the use of additional/alternate contextual factors can permit better modeling of phonological variation if a limited number of factors is used.

6. Concluding Remarks

In this paper, we presented a systematic, data-intensive approach for describing and modeling phonological variation. By basing the models on a large data set, counts of the occurrence of different variants are available for cross-validation to produce more robust models. We advocated a phone representation with an enriched set of contextual descriptors and the use of a combination of decision tree induction and hierarchical clustering to organize the pronunciation data into a context tree. Although tree induction methods do not find the "best" model, a "good" model identifying a subset of contextual factors is generally found. The context trees possess many properties which can be exploited in the creation of pronunciation networks. These properties permit ease of context combination, estimation of distributions from a partial context description in HMM's, representation of allophone probabilities, systematic reduction of network size, and identification of natural groups for tying in HMM's [6] from the leaves. We described a method for using the mixed context descriptions, as specified by the leaves of the context trees, for building pronunciation networks, permitting a wide variety of factors to be used to model contextual effects. Finally, our data indicate that use of contextual factors in addition to preceding and following phoneme can permit better modeling of phonological variation.

Acknowledgments. The author wishes to thank Jeff Shrager, Meg Withgott, and Julian Kupiec for many valuable discussions, and SRI for the use of their RULE system. This work was sponsored in part by the Defense Advanced Research Projects Agency (DOD), under the Information Science and Technology Office, contract #N00140-86-C-8996.

References

[1] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone, *Classification and Regression Trees*, Wadsworth International Group, Belmont, CA, 1984.

[2] P. Chou, *Applications of Information Theory to Pattern Recognition and the Design of Decision Trees and Trellises*, Doctoral Dissertation, Stanford University, Stanford, CA, June 1988.

[3] Y. Chow, R. Schwartz, S. Roucos, O. Kimball, P. Price, R. Kubala, M. Dunham, M. Krasner, and J. Makhoul, "The role of word-dependent coarticulatory effects in a phoneme-based speech recognition system," *Proc. ICASSP*, pp. 1593-1596, 1986.

[4] M. Cohen, *Phonological Structures for Speech Recognition*, Doctoral Dissertation, University of California, Berkeley, CA, April, 1989.

[5] F. Jelinek, "Self-organized language modeling for speech recognition," unpublished, IBM T.J. Watson Research Center, Yorktown Heights, N.Y., 1985.

[6] F. Jelinek and R. Mercer, "Interpolated estimation of Markov source parameters from sparse data," *Proc. Pattern Recognition in Practice Workshop*, E. Gelsema and L. Kanal, eds., North-Holland, 1980.

[7] L. Lamel, R. Kassel, S. Seneff, "Speech database development: design and analysis of the acoustic-phonetic corpus," *Proc. DARPA Speech Recognition Workshop*, L. Baumann, ed., pp. 100-109, 1986.

[8] K-F. Lee, *Large-Vocabulary Speaker-Independent Continuous Speech Recognition: The SPHINX System*, Doctoral Dissertation, Carnegie Mellon University, Pittsburgh, PA, April 1988.

[9] K-F. Lee, H-W. Hon, M-Y. Hwang, S. Mahajan, R. Reddy, "The SPHINX speech recognition system," *Proc. ICASSP*, pp. 445-448, 1989.

[10] D. Paul, "The Lincoln robust continuous speech recognizer," *Proc. ICASSP*, pp. 449-452, 1989.

[11] J.R. Quinlan, "Induction of decision trees," *Machine Learning*, Kluwer Academic Publishers, Boston, vol. 1, pp. 1-86, 1986.

[12] S. Sagayama, "Phoneme environment clustering for speech recognition," *Proc. ICASSP*, pp.397-400, 1989.

[13] M. Weintraub, H. Murveit, M. Cohen, P. Price, J. Bernstein, G. Baldwin, and D. Bell, "Linguistic constraints in hidden Markov model based speech recognition," *Proc. ICASSP*, pp. 699-702, 1989.