Decision Tree State Clustering with Word and Syllable Features

Hank Liao, Chris Alberti, Michiel Bacchiani, and Olivier Siohan

Google Research, New York, NY, USA

{hankliao, chrisalberti, michiel, siohan}@google.com

Abstract

In large vocabulary continuous speech recognition, decision trees are widely used to cluster triphone states. In addition to commonly used phonetically based questions, others have proposed additional questions such as phone position within word or syllable. This paper examines using the word or syllable context itself as a feature in the decision tree, providing an elegant way of introducing word- or syllable-specific models into the system. Positive results are reported on two state-of-the-art systems: voicemail transcription and a search by voice tasks across a variety of acoustic model and training set sizes.

Index Terms: decision tree state clustering, large vocabulary continuous speech recognition, tagged clustering.

1. Introduction

State-of-the-art large vocabulary continuous speech recognition systems use decision trees to cluster context dependent (CD) HMM states [1]. Context dependent models arise, for example, when models are conditioned on the left and right phones, yielding so-called triphone models. The total number of all possible triphones is quite large, and not all of them may be observed in the training data, leading to data sparsity issues. Using decision trees to cluster the state distributions of these models allows their robust estimation and synthesis of unseen contexts. When decision trees were first applied, training data numbered in the tens of hours: nowadays, training systems on a few thousand hours of speech is not unheard of.

To exploit this two order of magnitude increase in data, it may be reasonable to specify more contexts and grow deeper trees. This paper revisits decision tree state clustering with "tagged clustering" [2], but with novel context features, such as the word or syllable context the phoneme appears in, and investigates how the amount of data affects model size and recognition performance. First an overview of decision tree state clustering is presented, followed by details about the new context features, implications for finite state transducer (FST) based ASR, experimental results and finally conclusions.

2. Decision Tree State Clustering

Decision trees are often used in large vocabulary continuous ASR to cluster a large number of CD units into a smaller set whose distributions can then be robustly estimated. Hence, contexts with little data are combined until sufficient data are available. Furthermore, contexts not found in the training data, but present in testing, can be assigned a model using the decision tree. Clustering can occur at the phone level [3] or, as done in this work, the state level [1].

A standard sub-word acoustic unit is the triphone: a phoneme, aka phone, and its left and right context, e.g. the word `trees` is composed of the following four triphone models

`trees` → `t+r t-r+iy r-iy+z iy-z`

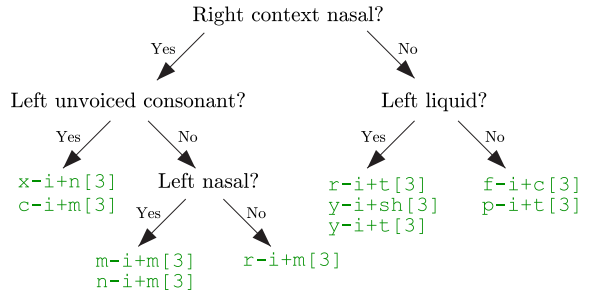

Figure 1: Decision tree clustering state 3 of phone `i`. 10 contexts are clustered into 5 leaves/states.

Each model is typically a 3-state HMM and the states themselves clustered using a decision tree with phonetic class questions. An example decision tree is shown in Figure 1.

Such binary decision trees are grown in a top-down fashion. First, a state alignment is obtained from a context independent or well-trained acoustic model. All the sufficient statistics for different contexts of the context independent state are pooled at the root node. Each node is modeled with a single Gaussian; the Gaussian distribution for a node can be easily estimated from state occupancies and 1st- and 2nd-order sufficient statistics

$$\boldsymbol{\mu}^p = \frac{1}{N^p}\sum_{c\in p}\boldsymbol{m}^c \qquad \boldsymbol{\Sigma}^p = \frac{1}{N^p}\sum_{c\in p}\boldsymbol{S}^c - \boldsymbol{\mu}^p\boldsymbol{\mu}^{p\mathsf{T}} \tag{1}$$

where p indicates a parent node and c a context with N^c the total state occupancy, $\boldsymbol{m}^c$ and $\boldsymbol{S}^c$ the 1st- and 2nd-order the sufficient statistics for context c; $N^p = \sum_{c\in p} N^c$. Hence given these statistics, it is easy to compute the centroid of node p, with mean $\boldsymbol{\mu}^p$ and diagonal variance $\boldsymbol{\Sigma}^p$ for any set of contexts.

Splits occur by greedily selecting the node that gives the best gain in likelihood by partitioning the contexts at that node. The goodness of the split is measured by the ratio of the log likelihood of the data at the node given by the following

$$\mathrm{LL}_{\text{gain}} = \log\left[\frac{L(X_{\in y}|\boldsymbol{\mu}^y,\boldsymbol{\Sigma}^y;)L(X_{\in n}|\boldsymbol{\mu}^n,\boldsymbol{\Sigma}^n)}{L(X_{\in p}|\boldsymbol{\mu}^p,\boldsymbol{\Sigma}^p)}\right] \tag{2}$$

$$= -\tfrac{1}{2}N^y\log(|\boldsymbol{\Sigma}^y|) - \tfrac{1}{2}N^n\log(|\boldsymbol{\Sigma}^n|) + \tfrac{1}{2}N^p\log(|\boldsymbol{\Sigma}^p|) \tag{3}$$

where p, y and n indicate the parent and yes and no child nodes. $X_{\in i}$ indicates the training data associated with node i, thus $X_{\in p} = X_{\in y} \cup X_{\in n}$. Examining Equation 3, it may be observed that splits attempt to minimize the variance within each node. In practice, a split is only considered if there are sufficient observations in each child node such that distributions can be robustly estimated. Tree growing stops when the gain in splitting nodes no longer exceeds a minimum threshold. The lowest gain leaves at the forest level can be pruned to reach a desired overall number of CD states.

The questions used to split nodes in triphone context modeling are devised by experts by grouping phones into broad phonetic classes, such as the ones in Figure 1; questions for each phone in isolation are also used. As described in [4], the same triphone context may be pronounced differently, e.g. triphone `iy-t+er` in "beater", "beat Earnest" and "return"; the first is flapped, the second an unreleased closure, and the last a closure plus release. This motivates the use of other contextual features such as stress [2] and position of the phone in the syllable [4] or word [5]. In particular it was found that it was unnecessary to differentiate word initial and final phones since this gave no gain over just indicating word boundary [5]. However, to the best of our knowledge, the use of word or syllable context of phones in decision tree building has not been reported.

3. Word and Syllable Context

In this paper, it is proposed that the phone models be conditioned on the actual word or syllable context. This produces word or syllable specific sub-word acoustic models that are chosen during the tree building process. Questions are made for every word or syllable in the system. Word context features are well motivated since they allow very specific context splits, creating word-specific phone models when the word is pronounced differently and frequently enough from what might be expected given the triphone context. In comparison, syllable context is broader. Both may be expected to give suitable way to grow larger trees and make better use of the increasing amounts of training data available. To use these features in tree building, models need to be tagged with this additional information.

In a FST-based ASR system, a weighted FST is used to represent the statistical language model G (weighted word acceptor), a phonetic lexicon L (context independent phone to word transducer), and context dependency transducer C (CD phone model to context independent phones). The FST algorithms referred to here are available in the *OpenFst* library [6]. Often the optimized decoding graph is built using the FST operations minimization, determinization and composition:

$$\min(C \circ \det(L \circ G)) \tag{4}$$

However if context dependent models are conditioned on lexical and syllabic features, C cannot be independent of the lexicon FST L. Thus a single combined CL transducer is created which maps sequences of context dependent models to words. The decoding graph is then constructed as follows:

$$\min(\det(CL)) \circ G \tag{5}$$

This type of construction can also provide some advantages described in [7], e.g. a smaller graph using reachable composition.

To create CL, first the phones in the lexicon need to be tagged with additional features as follows

```
0 1  t@word=trees#wb=true   trees
1 2  r@word=trees#wb=false
2 3 iy@word=trees#wb=false
3 4  z@word=trees#wb=true
```

Each line represents an arc in the transducer, where the first column indicates the *from state*, the second the *to state*, the third the *input label*, and the last column the *output label* (where present). Here input phones are tagged with word boundary and word features using special tokens `@` and `#` to separate the phone and feature key-value pairs. Note how the output word label is smeared across all input phones as a label context feature.

To introduce phonetic context, context labels from the decorated source lexicon FST are pushed onto the output FST. A mapping is maintained from source state indices to destination state indices with context labels. Starting from the initial state on the queue, source FST states are expanded by iterating over all the outgoing arcs, possibly splitting destination states to ensure a single context per destination state. For each arc, the context label and new arc on the output FST are added. For the new arc context and source state, if the destination state does not exist, it is created. This generates a CL with left context

```
0 1  t@word=trees#wb=true#left=sil trees
1 2  r@word=trees#wb=false#left=t
2 3 iy@word=trees#wb=false#left=r
3 4  z@word=trees#wb=true#left=iy
```

For right context, the FST is reversed and the same algorithm applied. To produce a training CL, this algorithm is applied to the lexicon FST, and contexts are then mapped to the tied CD models using the decision trees. The resulting FST is compacted by determinizing and minimizing the label encoded input FST, and then decoding the labels. The output FST can be used as an alignment network for counting statistics of all observed contexts to begin decision tree building. To create the test CL as in Equation 5, disambiguation symbols need to be introduced before determinization and minimization. Since the decision trees are not used when creating CL, the pre-optimized intermediate FST will contain many redundant arcs and hence is not suitable for generating pentaphonic, i.e. ± 2 phone, context.

3.1. Chou Partitioning

Chou's partitioning algorithm (CPA) [8] can be used to find a locally optimal split of the data for a node in the decision tree. It can be considered an application of K-means clustering. Two clusters representing a potential partitioning are initialized by splitting the parent node Gaussian. The mean is perturbed by some fraction of the variance, ± 0.2 was used in this work, and then several iterations of K-means performed until convergence. Each split is considered along a particular context dimension such as left or right neighbouring phone. CPA was applied to phonetic decision tree state clustering with a backoff to handcrafted phonetic classes for handling unseen triphones in [9]. This paper proposes to use CPA without the backoff and using the smaller set of labels of a child in a split to form the members to form the 'yes' node of a question. Such an approach could be useful where appropriate classes are not obvious: should a phone in particular word contexts be partitioned on whether the word is a function or content word, of foreign origin, or highly confusable? In some cases this may not be ideal, but applying CPA in this manner allows a completely unsupervised approach to creating arbitrary questions without the need for expert knowledge.

4. Experiments

Experiments were conducted on two tasks: the first, voicemail transcription; the second, a search by voice task a.k.a. Voice Search. These are internal Google data sets. Exploratory experiments are conducted on the voicemail task as the speech is more conversational and spontaneous. The query by voice task however has more available supervised training data which allows analysis on how the additional context in tree building scales with the amount of data. All the systems in this paper use a relatively large search beam such that search errors are not a factor in the experiments. The number of Gaussians, M, for a

state distribution follows a power law on the number of observations, N, aligned to the state: $M = \beta N^{0.4}$. This has also been referred to as VarMix as opposed to a fixed number of components per GMM. The language model and insertion penalty were not tuned for each system, but set to optimize baselines.

4.1. Voicemail Transcription

The voicemail transcription system is trained on 425 hours of data with around 50k voicemails. Two test sets were available each totally about 35k words or over 3 hours of speech. The language model is a Kneser-Ney smoothed, entropy pruned, trigram language model, interpolated from a variety of sources including the transcripts themselves and broadcast news; the total number of Ngrams was 3.5M and had a perplexity of 52 on the test set. The vocabulary contains 50k words with an out-of-vocabulary rate of 0.4%. The ML-trained acoustic models apply STC, LDA with 9-frame stacked static PLP features projecting 117 dimenstions down to 39, VTLN and CMLLR-SAT in a multi-pass decoding strategy to give speaker adapted systems; gains found at the ML level were found to hold after applying MMI discriminative training on the best configurations, so for expediency only ML results are reported. The lexicon was syllabified using NIST `tsylb` software [10].

In section 3.1, a form of CPA was presented to allow automatic generation of questions. This was first compared against standard questions based on phonetic classes as shown in Table 1. As expected, the average alignment cost per frame de-

Question type	Number of States		
	7000	9000	12000
	Avg. Cost per Frame		
Phonetic classes	3.06	3.03	3.00
Chou partitioning	3.07	3.03	2.99
	% WER		
Phonetic classes	27.5	27.4	27.4
Chou partitioning	27.9	27.3	27.3

Table 1: Comparing triphone systems using hand-crafted phonetic classes with automatic CPA questions. Average cost per frame (smaller indicates better fit) and word error rate are reported for increased number of leaves/states.

creases with the increased size of the decision trees and hence context dependent states. With fewer states the phonetic class based questions gave a slightly better alignment cost, which is somewhat unexpected, but with increased number of states CPA becomes slightly better. These costs correlate well with WER when comparing phonetic class versus CPA for a given number of states; however the phonetic class based decision trees do not significantly improve with a larger number of states despite the corresponding increase in likelihood. As found in [9] CPA does not perform better than phonetic classes. Note, with 7000 states the total number of Gaussians in the model is about 110k, at 12000 states 150k; even though there are 36% more Gaussians in the larger system the smaller system gives relatively the same result. Reducing the number of states to 6000 slightly degraded the baseline system accuracy. Also some experiments were conducted using CPA to determine questions for word context; this actually gave a large increase in error rates on the order of 10% relative; upon inspection of the splits, applying CPA in this manner produced many splits involving a large amount of frames rather than minimizing the variance. Perhaps instead of using total likelihood gain, incorporating an average gain per frame may regulate the partitioning. Given these mixed results, subsequent experiments will focus solely on using hand-crafted phonetic class questions.

This paper suggests word and syllable context may be useful for clustering CD states. These features may be compared and complemented with a word boundary context feature. The following results show how well these context features combine with phonetic class triphonic questions. In Table 2 it can

Phonetic		Word		Syllable
Left	Right	ID	Boundary	ID
50.9%	49.1%			
41.4%	40.6%			18.1%
47.6%	46.6%		5.7%	
40.8%	41.5%	17.7%		
39.4%	39.8%	16.0%	4.8%	

Table 2: Percentage of splits by context feature for different 9000 leaf trees. Each row is a mix of different context features.

be seen that the percentage of splits on left and right neighbouring phones is fairly even. When word boundary is available it is only used about 5% of the time, much less than phonetic, word or syllable splits. This is half of the total quaternary initial/internal/final/only word position splits reported in [4].

Table 3 presents results when using different context features. At 7k states, all the context features give gains over the

Context		Number of States		
Phonetic	Non-phonetic	7000	9000	12000
Triphone	—	27.5	27.4	27.4
	Syllable ID	27.4	27.4	27.0
	Word Boundary	26.9	27.3	26.9
	Word ID	26.9	27.1	27.0
	Word Boundary + ID	26.9	26.9	26.5

Table 3: Performance results when using different context features (% WER) corresponding to the rows in Table 2.

baseline triphone system. However, despite many splits on syllable context, 18.1%, it is not very helpful compared to other features. At 9k states, the binary word boundary feature is used only 5.7% of the time, yet is slightly better than the more flexible syllable feature. Surprisingly, the word context and word boundary context give similar results; these two are complementary, improving the baseline result by 0.9% absolute.

With larger numbers of Gaussians, by increasing β, the performance of the syllable and word systems was better than word boundary, but the gain between the combined and baseline system still remains as shown in Figure 2. Baseline triphonic context was best at 7k states; large trees either gave bigger models with no improvement or slightly degraded results. In contrast, when adding word and word boundary features, growing the trees larger to 9k or 12k states gave better results. This improvement was consistent across a range of acoustic model sizes. Performance peaked for all systems at around 400k Gaussians—with about 150M training frames, this provides slightly less than an average of 400 frames of data for each Gaussian to be estimated. This seems a reasonable size for the acoustic model to plateau in performance given the amount of training data. For 140k Gaussian discriminatively trained systems using boosted MMI, the baseline triphone gave a WER of 25.3% and improved to 24.2% with word and word boundary context clustering.

4.2. Search by Voice Task

The Voice Search task, see `www.google.com/mobile`, facilitates experiments on how these additional context features behave with a larger amount of training data. For these experiments, ML-trained STC speaker independent models, with an

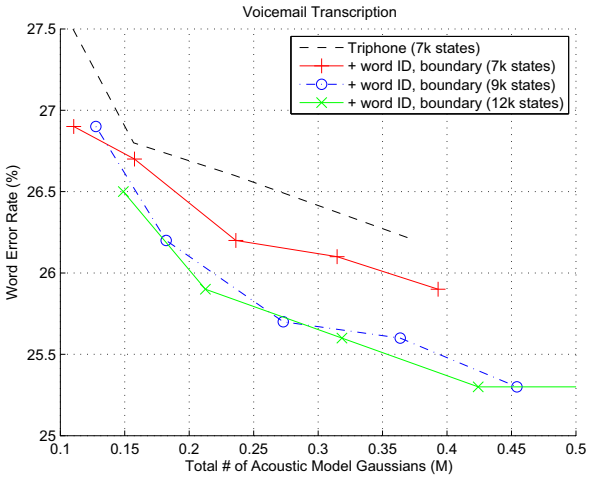

Figure 2: Comparing baseline triphone system with various combined triphone/word/word boundary systems, increasing number of system Gaussians.

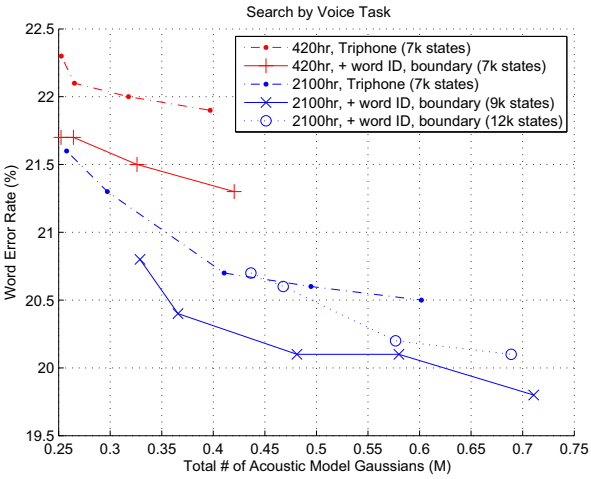

Figure 3: Comparing baseline triphone system with various combined triphone/word/word boundary systems, varying training set size and increasing number of system Gaussians.

LDA projection of 9-frame stacked static PLP features down to 39 dimensions are used. The language model is a back-off trigram model containing 14M Ngrams and vocabulary of 1M words. The test set contains 14k words with each utterance about 3 seconds long. A single-pass decoding strategy is used.

Figure 3 compares standard phonetic class triphonic systems with added word and word boundary context for Voice Search. The top two lines demonstrate the gain in including the additional context features with 420 hours of training data, but the gain of 0.4%-0.6% is smaller compared to the voicemail task with a comparable amount of training data. Increasing the number of states to 9k or growing the models further did not improve the overall performance with this amount of data. Examining the number of splits for each context in Table 4 shows why this might be. Compared to the number of splits in the 9k

Train Set	Context	Number of States 7000	9000	12000
420hr	Left phone	44.9%	43.7%	—
	Right phone	39.4%	38.3%	—
	Word ID	10.0%	12.7%	—
	Word boundary	5.7%	5.3%	—
2100hr	Left phone	44.0%	43.4%	42.9%
	Right phone	38.3%	36.9%	36.6%
	Word ID	11.8%	14.3%	16.0%
	Word boundary	5.9%	5.4%	4.4%

Table 4: Percentage of splits of each context feature, varying the number of states and amount of training data.

system in Table 2, there are fewer word questions, e.g. 12.7% to 16.0%. This indicates less variation due to word context, which seems reasonable given short utterances. At the 2100 hour training set size, a slightly larger tree size of 9k states is better and the total size of the acoustic model can be larger. Here, the absolute WER improvement over the baseline is slightly, but not sizably, larger at 0.8 than with 420 hours. Only a few hundred words of a 150k word training vocabulary are split on, the most frequent splits include e.g. california, restaurant, florida, texas and restaurants, but also include function words; these are among the high frequency words in the training data. Of note, at the smaller model sizes the 420 hour deeper context system is almost as accurate as the baseline triphone system trained on five times more data.

5. Conclusions and Future Work

This paper investigated the use of new context features based on the context dependent phone model's word and syllable context in combination with other standard features such as word boundary and triphonic context. On two different real-world applications, with state-of-the-art systems, the use of these additional features, in particular a combination of triphone, word and word boundary, gave a consistent and reliable gain over a baseline triphone system. This gain holds over varying model sizes, amounts of training data and with discriminative training. These additional features may help model the variation due to the spontaneity of the speech for the task: on the voicemail transcription the gain is about 1% absolute, but for Voice Search around 0.5%. The gains would probably be less in a more controlled, read speech task. Future work could entail exploring other context features, or larger phonetic context, although the results from using pentaphones are mixed [4] and may overlap with syllable or word context features.

6. References

[1] S.J. Young and P.C. Woodland, "State clustering in HMM-based continuous speech recognition," *Computer Speech and Language*, 1994.

[2] D.B. Paul, "Extensions to phone-state decision-tree clustering: single tree and tagged clustering," in *Proc. ICASSP*, 1997.

[3] L.R. Bahl, P.V. de Souza, P.S. Gopalakrishnan, D. Nahamoo, and M.A. Picheny, "Decision trees for phonological rules in continuous speech," in *Proc. ICASSP*, 1991.

[4] I. Shafran and M. Ostendorf, "Acoustic model clustering based on syllable structure," *Computer Speech and Language*, 2003.

[5] C. Fügen and I. Rogina, "Integrating dynamic speech modalities into context decision trees," in *Proc. Eurospeech*, 2000.

[6] `www.openfst.org/`

[7] C. Allauzen, M. Riley, and J. Schalkwyk, "A generalized composition algorithm for weighted finite-state transducers," in *Proc. Eurospeech*, 2009.

[8] P. Chou, "Optimal partitioning for classification and regression trees," *IEEE TPAMI*, vol. 13, no. 4, Apr. 1991.

[9] H.J. Nock, M.J.F. Gales, and S.J. Young, "A comparative study of methods for phonetic decision-tree state clustering," in *Proc. Eurospeech*, 1997.

[10] `www.nist.gov/speech/tools/tsylb2-11tarZ.htm`