22. C. Wooters, and A. Stolcke, “Multiple-Pronunciation Lexical Modeling in a Speaker Independent Speech Understanding System,” 1994 ICSLP.
23. G. Tajchman, D. Jurafsky, and E. Fosler, “Learning Phonological Rule Probabilities from Speech Corpora with Exploratory Computational Phonology,” 1995 ACL
24. G. Tajchman, E. Fosler, and D. Jurafsky, “Building Multiple Pronunciation Models for Novel Words Using Exploratory Computational Phonology,” 1995 Eurospeech.
25. F.R. Chen, “Identification of Contextual Factors for Pronunciation Networks,” 1990 IEEE ICASSP. pp. 753-756.
26. J.L. Gauvain, L.F. Lamel, G. Adda, and M. Adda-Decker, “The LIMSI Continuous Speech Dictation System: Evaluation of the ARPA Wall Street Journal Task,” 1994 IEEE ICASSP, pp. I-557 to I-560.
27. S. Greenberg, J. Hollenback, and D. Ellis, “Insights into spoken language gleaned from phonetic transcription of the Switchboard Corpus,” ICSLP ‘96, SaP2S1.1.

6.0 REFERENCES

1. J. J. Godfrey, E.C. Holliman, and J. McDaniel, “SWITCHBOARD: Telephone speech corpus for research and development,”, 1992 IEEE ICASSP, pp. 517-520.
2. R.M. Warren and C.J. Obusek, “Speech perception and phonemic restoration,” Perception and Psychophysics, 9(3B), pp. 358-362.
3. G. Zavaliagkos, J. McDonough, “Acoustic Modeling,” presented at the April 29, 1996 LVCSR Hub 5 Workshop, Linthicum Heights, Maryland.
4. J.M. Lucassen, and R.L. Mercer, “An Information Theoretic Approach to the Automatic Determination of Phonemic Baseforms,” 1984 IEEE ICASSP, pp. 42.5.1 - 42.5.4.
5. L. R. Bahl, P.F. Brown, P.V. de Souza, R.L. Mercer, and M.A. Picheny, “Acoustic Markov Models Used in the Tangora Speech Recognition System,” 1988 IEEE ICASSP, pp. 497-500.
6. L.R. Bahl, J.R. Bellegarda, P.V. de Souza, P.S. Gopalakrishnan, D. Nahamoo, M.A. Picheny, “A New Class of Fenonic Markov Word Models for Large Vocabulary Continuous Speech Recognition,” 1991 IEEE ICASSP, pp. 177-180.
7. L.R. Bahl, P.V. de Souza, P.S. Gopalakrishnan, D. Nahamoo, M.A. Picheny, “Decision Trees for Phonological Rules in Continuous Speech,” 1991 IEEE ICASSP, pp. 185-188.
8. L.R. Bahl, S. Das, P.V. de Souza, M. Epstein, R.L. Mercer, B. Merialdo, D. Nahamoo, M.A. Picheny, J. Powell, “Automatic Phonetic Baseform Determination,” 1991 IEEE ICASSP, pp. 173-176.
9. M.Y. Hwang, and X. Huang, “Subphonetic Modeling for Speech Recognition,” 1992 Speech and Natural Language Workshop, pp. 174-179.
10. T. Sloboda, “Dictionary Learning: Performance Through Consistency,” 1995 IEEE ICASSP, pp. 453-456.
11. T. Sloboda, A. Waibel, “Dictionary Learning for Spontaneous Speech Recognition,” 1996 ICSLP
12. C.H. Lee, F.K. Soong, and B. H. Juang, “A Segment Model Based Approach to Speech Recognition,” 1988 IEEE ICASSP, pp. 501-504.
13. E.P. Giachin, A.E. Rosenberg, and C.H. Lee, “Word Juncture Modeling using Phonological Rules for HMM-Based Continuous Speech Recognition,” 1990 IEEE ICASSP, pp. 737-740.
14. M.D. Riley, “A Statistical Model for Generating Pronunciation Networks,” 1991 IEEE ICASSP, pp. 737-740.
15. J. Bernstein, G. Baldwin, M. Cohen, H. Murveit, and M. Weintraub, “Phonological Studies for Speech Recognition,” Proceedings of DARPA Speech Recognition Workshop, February 19-20, 1986, pp. 41-48.
16. M. Cohen, G. Baldwin, J. Bernstein, H. Murveit, and M. Weintraub, “Studies for an Adaptive Recognition Lexicon,” Proceedings of DARPA Speech Recognition Workshop, March 24-26, 1987, pp. 49-55.
17. M. Cohen, *Phonological Structures for Speech Recognition*, U.C. Berkeley Ph.D. Thesis, 1989.
18. M. Weintraub, H. Murveit, M. Cohen, P. Price, J. Bernstein, G. Baldwin, D. Bell, “Linguistic Constraints in Hidden Markov Model Based Speech Recognition,” 1989 IEEE ICASSP, pp. 699-702.
19. M. Weintraub, K. Taussig, K.H. Smith, A. Snodgrass, “Effect of Speaking Style on LVCSR Performance,” 1996 ICSLP
20. C. Wooters and N. Morgan, “Acoustic Subword Models in the Berkeley Restaurant Project,” 1992 ICSLP, pp. 1551-1554.
21. C.C. Wooters, *Lexical Modeling in a Speaker Independent Speech Understanding System*, U.C. Berkeley Ph.D. Thesis, 1993.

5.0 SUMMARY

A summary of the major findings of this group are:

- 1% Improvement in WER with dictionary replacement (Significant) (Section 4.4).
- 0.2% Improvement in WER with decision tree models (Not Significant) (Section 4.8).
- Constrained alignment of phones on training set using pronunciation graphs did not allow enough pronunciation variation (Section 4.3).
- Linguistic information: stress and syllabic position are best predictors of pronunciation variation (Section 4.2).
- Likelihood of correct words are greatly increased relative to 1-Best hyp; improvement also helps other sentence hypotheses and this limits performance improvements (Section 4.7).

There were several experiments that we started at the workshop but did complete. The two major experiments are listed below:

- Retrain acoustic models: The original acoustic models were broader than they needed to be since they were implicitly modeling pronunciation variation. Retraining the acoustic models when allowing pronunciation variation should make the acoustic models much sharper, and potentially increase LVCSR performance. However time and CPU constraints prevented us from exploring this option.
- Consistency of original pronouncing dictionary: The original pronunciation dictionary was not very consistent in it's representations of the different words. This lack of consistency limits how effective the decision trees can be, since they are trying to predict the model from the baseform pronunciation to the observed phone sequences. We started a project to estimate a new baseform pronunciation. This new baseform pronunciation for each word, when used with the decision trees, would be the best predictor for observed pronunciation data. This procedure would make the baseline pronunciations consistent with the decision tree model, and potentially allow the decision tree approach to equal the direct dictionary replacement algorithm's performance.

There are additional factors such as dialect modeling (consistency of pronunciation for each speaker) or the dependence of the pronunciation on other factors (speech rate or vocal effort) that we never explored but are important to understand in future research.

on the 200 sentence development test subset that was reported on earlier in Table 6. The experiments rescored the top 75 to 100 N-best sentence hypotheses.

Experiment	**Word Error Rate**
Baseline HTK	46.4
Modified HTK: No “sp” Phones	45.7
DT1 Observed Pronunciations: Threshold = 7 Count	45.2
DT2 Generated Pronunciation Graphs Using Decision Trees	45.5
DT3 Generated Pronunciation Graphs Using Decision Trees	45.5

Table 8: Word error rate for different probabilistic decision trees

By applying the decision trees to the baseline pronunciations, we obtained a very small improvement in performance (45.7 -> 45.5). The application of the decision trees was worse than just replacing the pronunciations of the common words with the aligned phone strings (45.5 worse than 45.2). Explicit dictionary replacement with commonly observed pronunciations outperforms pronunciation word graphs

4.8.2 Maximum Entropy Model

The LVCSR rescoring results of the maximum entropy phone language model on the leaves of the decision tree are shown below. This is also on the 200 sentence development test set but these results just rescore the top 20 N-best hyps.

Experiment	**Word Error Rate**
Baseline HTK	46.4
Modified HTK: No “sp” Phones	46.0
Dictionary from DT2 Phone Alignment: Threshold = 11 Count	45.3
DT3 Generated Pronunciation Graphs with Maximum Entropy Phone Constraints	47.6

Table 9: Word error rate for different probabilistic decision trees.

The word error rate for the maximum entropy language model had a higher word error rate than the baseline system (47.6 for the maximum entropy phone LM compared to 46.0 for the relevant baseline system). This is surprising since the maximum entropy constraints seemed to work well for the diagnostics experiments described in Table 7. In the diagnostic experiments, the use of maximum entropy constraints resulted in the second best system. This led us to believe that there might have been a bug in our implementation of this experiment, but we could not find it.

Experiment	Average Per Frame Acoustic + Pronunciation Score				WER
	Correct	1-Best	Difference	#Correct > 1-Best	w/LM
Original	-68.49	-68.18	0.31	43	46.4
DTPron [max=1]	-68.32	-68.14	0.18	62	44.1
DTPron + 2g on phn pair [max = 1]	-68.35	-68.14	0.21	59	45.0
DTPron + 2g on phn pair [sum = 1]	-69.07	-68.92	0.15	62	44.5
DTLeaf + 3g on Obs [sum = 1]	-69.39	-69.28	0.11	78	41.9
Maximum Entropy for the above [sum=1]	-69.23	-69.16	0.07	77	42.2

Table 7: Experimental results on 194 sentences of development test set.

These results suggest that the maximum potential improvement that we could hope to achieve with this particular pronunciation modeling is 4.5% (46.4 -> 41.9), without retraining the acoustic models. This system corresponds to using a trigram language model on the observed phones at the leaves of the decision tree. Note that the leaf trigram grammar was much more powerful than the leaf bigram grammar. In addition, when using a statistical grammar to add additional constraints, the traditional algorithm of summing the probabilities to 1.0 was the best. Perhaps this is because the decision trees were applied uniformly to every word in the vocabulary. The previous technique (from Section 4.4) where the most common observed phone recognition sequence used applied only to the most frequent words, and so the “max = 1” technique was needed to equalize between those words that were replaced with probabilistic dictionaries and those words that retained the original dictionary entries.

Note that these experiments are overly optimistic in terms of LVCSR performance. This is because even though the correct sentence hypothesis would now be ranked better than the original 1-best hypothesis, there may be other hypotheses in the N-best list that would also be better than the original 1-best hypothesis. A real recognition experiment was therefore run and the experiments are described in the next section.

4.8 Rescoring Results Using Decision Trees

4.8.1 Basic Decision Trees with No N-gram Constraints

Decision trees were used both for realigning training data as well as for applying to words as shown in Figure 9. Applying the decision trees to the original dictionary pronunciation, allows probabilistic pronunciation models to be constructed for each word. The LVCSR results below are

Word	Original Dictionary	Probabilistic Dictionary	
		Count	Pronunciation
another	ax n ah dh er	76	ax n ah dh er
		14	n ah dh er
		11	ax n ah dh ax
especially	eh s p eh sh ax l iy	12	eh s p eh sh ax l iy
		11	ih s p eh sh ax l iy
		10	s p eh sh ax l iy
something	s ah m th ih ng	233	s ah m th ih ng
		58	s ah m th ih n
		14	s ah m ih ng
		13	s ah m th ng
because	b ax k ah z	431	b ax k ah z
	b ax k ao z	71	k ah z
		47	b ax k ao z
		15	ax k ah z
		12	b ih k ah z
		10	b ax k ah z dh
kind	k ay n d	355	k ay n d
		102	k ay n
		21	k ah n
		15	k ay n dh

4.7 Diagnostic Studies

We wanted to gain some insight as to important factors in pronunciation modeling, and we needed an experimental paradigm that allowed for much faster experimentation. Therefore, for each of ~200 sentences we selected only two sentence hypotheses to compare: (1) the correct transcription and (2) the original system's best hypothesis with the original dictionary. Each of these two sentence hypotheses were aligned with the speech data and had the pronunciations replaced with different probabilistic dictionaries. Several features were compared and the results are shown below in Table 7.

The first column in Table 7 describes which dictionary conditions we used. Columns 2, 3, and 4 display the average log likelihood (per frame) of the different hypotheses. The 5th column shows how many of the correct hyps scored better than the original system's 1-best hyp when using only acoustic scores. The last column is the word error rate when adding in the language model (when only selecting between these two sentence hypotheses).

AND ae n d sil

AND ae n d sp

The “sp” phone is for “short pause.” The topology for the sp phone allows skips, and the output distribution is the middle state of silence. The sp phone can be aligned to 0 or more frames. Since it has a self loop, it can align to an arbitrary length duration of pause. Although the “sp” phone is supposed to only match short pauses, it’s self loop allows it to match very long pauses (e.g. 400 msec was the longest observed sp model in test set). For this reason, we decided to delete the “sp” phone from the dictionary and explicitly model whether pauses were present in acoustics. If a pause was present between two words, we would include an extra “pause” word in the sentence hypothesis. An additional benefit of this decision is that we are explicitly determining whether cross-word acoustics can be applied.

4.6 Pronunciation Dictionary Generated from Decision Trees

The following experiments used a dictionary that was generated from DT1 alignments. The results here are on a random 200 sentence subset of the development test set. While the results of experiment listed in Table 5 were on the whole development test set, the performance on this 200 sentence subset exactly mirrored the resulting performance on the whole dev set.

Experiment	**Word Error Rate**
Baseline HTK	46.4
Modified HTK: No “sp” Phones	45.7
DT1 Observed Pronunciations: Threshold = 7 Counts	45.2
DT1 Observed Pronunciations: Threshold = 3 Counts	45.3

Table 6: LVCSR performance for probabilistic pronunciation dictionary using DT1 alignments.

The results from Table 6 show that

- Improvement of 0.7% over Baseline by removing “sp” phone (46.4 -> 45.7)
- Further improvement of 0.5% with dt1 observed pronunciations (45.7 -> 45.2)
- Lowering threshold below count of 7 did not improve performance
- Total improvement of 1.2% (46.4 -> 45.2)

4.6.1 Sample Pronunciations from DT3

A set of sample pronunciations generated from the latest DT3 alignments is listed below to illustrate the types of pronunciation variation found by these techniques.

pronunciation. To compensate for these differences, we consider the case where we modify the pronunciation probabilities so that the maximum pronunciation probability is scaled to be 1.0.

To combine this dictionary with the standard lexical pronunciation dictionary, we removed the words (and their corresponding pronunciations) from the original lexical dictionary that were in the new pronunciation dictionary. Then, each pronunciation in the original lexical dictionary was assigned a pronunciation probability of 1.0. The two dictionaries were then combined and this new dictionary was used for recognition experiments.

4.4.1 Observed Pronunciation Dictionary Experiments

The first set of experiments used a dictionary generated from the original phone recognizer with a phone bigram grammar. The pronunciation probabilities were scaled so that the maximum pronunciation probability for each word was 1.0. The pronunciation probabilities were treated like a language model, and were scaled by an independent weight. The best performance for the trigram language model was using a scale of 12.0. For the pronunciation probabilities, the minimum error rate occurred when the pronunciation probabilities were scaled by factors of either 10 or 12. This indicated that the pronunciation probabilities can be treated as language model transitions, and need to be weighted relative to the acoustic models in a similar way to current language models.

The word error rate using the new dictionary was computed on the whole development test set and is show below in Table 5:

Experiment	**Word Error Rate**
Baseline HTK dictionary	46.4
New dictionary: Phone Recognizer with Bigram Grammar Threshold count = 10 Max pronunciation probability = 1.0	45.5

Table 5: Word error rate comparing the original dictionary and the new dictionary generated using a phone recognizer

This improvement in LVCSR performance due to the new dictionary was 0.9%. This improvement in performance was significant. One interesting point about the resulting dictionary was that for many words the pronunciations looked reasonable. However for short words such as “A” or “I”, many of the pronunciations were clearly not plausible in a linguistic sense, although there was an acoustic match for these pronunciations.

4.5 Handling of PAUSE in HTK

LVCSR systems need to allow for pauses between words. For many systems, pauses are treated as words and handled explicitly in the grammar. In HTK, pauses between words are handled in the pronunciation dictionary. A sample pronunciation from the HTK dictionary is listed below:

nunciations was a better match to the acoustics. By this time the constraints were too powerful, and the statistics in this case (line 4 in Table 4) did not match the hand labeled phones. Due to time constraints (each line took several CPU months to compute) we were not able to repeat these experiments with a less constrained grammar.

4.4 Pronunciation Dictionary Generated from Bigram Phone Recognizer

The aligned data from the different grammars listed below were used to construct new probabilistic pronouncing dictionaries:

- Phone recognition alignment
- Constrained alignment from decision trees (dt1)
- Constrained alignment from dt2

These alignments as described in the column on the right hand side of Figure 1 illustrated how a dictionary can be constructed (e.g. a sample set of Unix shell commands is listed below):

cat aligned_data | fgrep "WORD:" | sort | uniq -c | gawk '{if ($1 > 10) print}' > new-dict1

Here, the aligned data was generated by aligning the current dictionary with one of the above 3 sets of phone sequences. The pronunciation for each word is accumulated, and a threshold is set. There were two different thresholds used for generating the dictionary. The first threshold was that the phone sequence had to occur at least N times (N was typically set between 3 and 20). Pronunciations that were seen a small number of times won't necessarily generalize to a testing set. There may have been certain anomalous reasons why a particular phone sequence was recognized. Only if a particular sequence happened many times do we have any confidence that there is some real underlying process that is affecting the acoustics.

In addition to this first threshold (number of counts), we also added a second threshold that was based on the probability of this pronunciation (relative to all pronunciations for this word). If the number of counts for this pronunciation divided by the total number of counts for this word (i.e. the relative frequency of this pronunciation was less than a threshold (typically 1.0e-4) the pronunciation would also be pruned.

The recognized phone sequences that were above these two thresholds, along with their counts/probabilities defined a new probabilistic pronunciation dictionary for the common words.

There were two different normalization algorithms that were used to estimate the pronunciation probabilities for the common words in the new probabilistic pronunciation dictionary.

- The pronunciation with maximum count has it's probability set == 1.0 and all other pronunciations are scaled accordingly.
- Sum pronunciation counts (for all pronunciations that are greater than the threshold and make it to the dictionary) and normalize the probabilities to sum to 1.0

In Section 3.1 we described the difficulties of summing the probabilities when a word has multiple pronunciations. Therefore, a word with multiple pronunciations would be penalized for splitting its pronunciation mass among different hypotheses compared to a word that only had a single

4.3 Phone Alignment Statistics

The statistics of different phone recognizers are shown below in Table 4. For the correct answers, we selected the current pronouncing dictionary as the truth, since we did not have hand transcriptions for the data in these experiments.

Experiment	Phones Correct	Lexical Phone Deletion Rate
1. Hand Labeled Phones	67.3	12.5
2. Phone Recognition: Phone Bigram	66.7	12.7
3. Use Orthographic Transcription to Obtain Lexical Pronunciation Baseforms. Apply Decision Trees DT1 to Lexical Pronunciation Baseforms Use Resulting Pronunciation Graph (from Decision Tree Leaves) as Constraint for HTK system. Compute Viterbi Backtrace using Pronunciation Graph. Pronunciation Probabilities ** 10	93.1	3.7
4. Same as system 3 except: Use Second Generation Decision Trees: DT2 No Pronunciation Probabilities	89.7	4.0

Table 4: Performance of different iterations: phone recognizers using bigram grammar and using previous iterations decision trees.

The results of experiments 3 and 4 in Table 4 correspond to using the 8 steps described in Section 3.3. Instead of using an N-best hypothesis, we use the transcribed words as the sentence hypothesis. The application of the decision trees to the sentence orthographics results in a pronunciation graph (as defined by the leaves of the decision tree). This pronunciation graph was then used as the constrained search, and the Viterbi path through this graph is what was output as the phone sequence.

By comparing the first two lines in Table 4, we see that the overall statistics of hand labeled phones and phone recognition are quite similar. However, when we compare the statistics of the alignment procedure generated from the first set of decision trees (line 3 in Table 4), we see that the correct rate is much higher (better agreement with the original dictionary) and the deletion rate is also much lower. We believe that the reason for this is that we weighted the pronunciation probabilities by the language model weight of 10.0. The reason that we did this is that this turned out to be the optimum language model weight when used in recognition experiments described later in this section. However, for aligning the training data we realized that the statistics did not match the ideal of the hand labeled phones.

To try to make up for this, when using the second generation of decision trees we did not use any pronunciation probabilities, but just let the acoustic models decide which of the possible pro-

4.2.2 Top 20 Questions for 3rd Generation DT

Question	# of Times Question Was Asked as A Function of Depth in Decision Tree							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
L1_CODA	8	0	4	8	6	10	2	0
R1_NUCLEUS	6	4	2	2	6	8	4	2
R1_Silence	4	6	20	8	2	6	2	2
L1_ONSET	4	6	0	8	14	2	4	4
R1_STRESS_1	4	4	2	4	8	12	2	0
C_CODA	4	4	0	4	0	2	2	0
L1_NUCLEUS	4	2	4	4	10	0	0	0
R1_Syllabic	4	0	2	2	2	2	4	2
L1_Silence	2	8	6	10	6	2	2	0
C_STRESS_1	2	6	6	4	12	14	2	4
L1_Lenis	2	6	4	6	2	4	0	2
L1_STRESS_1	2	6	2	4	14	6	6	18
L1_Sonorant	2	2	8	0	0	0	0	0
R1_Anterior	2	2	4	4	2	4	0	0
L1_Nasal	2	2	2	8	6	12	4	6
L1_Stop	2	2	0	4	6	6	4	6
L1_High	2	2	0	0	10	0	2	0
R1_ONSET	2	0	10	2	10	12	2	2
R1_STRESS_0	2	0	4	8	6	4	0	4
R1_STRESS_2	2	0	4	2	12	4	2	2

Table 3: The most common splitting criteria in the decision trees. The term "L1_" on the first line means that the most common question was when 1 phone to the left of the current phone was a coda. Similarly, the term "R1_" means one phone to the right, and "C_" means the current phone.

Decision tree questions indicate which information is useful in predicting the realization of a particular phone. It is important to note that 11 of the top 20 questions are linguistic questions concerned with stress and syllabic position information.

4.2 Building Decision Trees

4.2.1 Decision Tree Pruning

The decision trees were built until the training data was exhausted. A separate cross-validation training data set was used to prune the decision trees The decision trees were pruned using perplexity on the cross-validation training set. The perplexity was measured on the held out cross-validation training data for each phone's decision tree. This was done as a separate and automatic post-processing step from the building of the decision trees. The entropy on the cross-validation test set is shown below as a function of the number of leaf nodes.

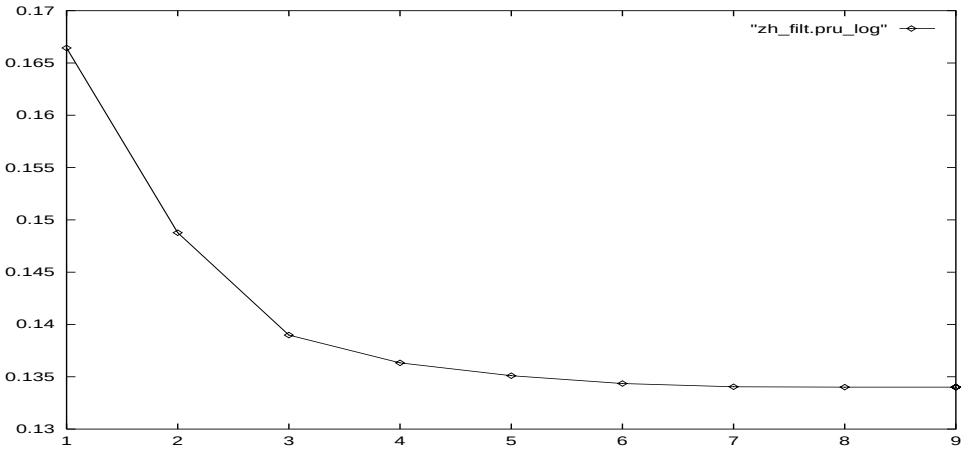

Figure 11: Entropy of the phone ZH as a function of the number of leaf nodes.

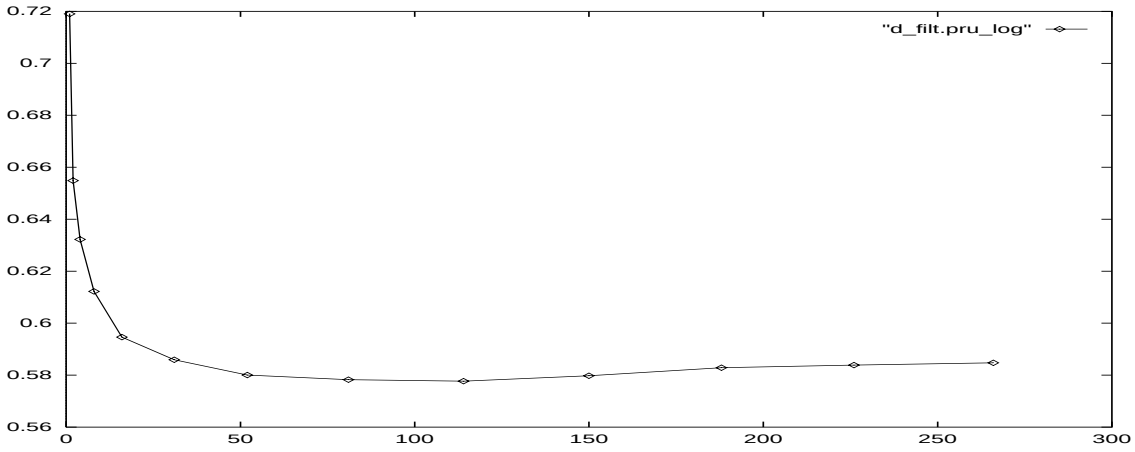

Figure 12: Entropy of the phone D as a function of the number of leaf nodes.

It was determined to automatically prune these decision trees when the slope of these curves was -45 degrees (computed in the lower left hand corner). This stopping criterion was selected by watching how a Seven selected the operating point when balancing the complexity of the trees against the improvement in entropy. Rather than pick the number of nodes that corresponds to the lowest entropy on the cross-validation set (that would have corresponded to a slope of 0.0), we wanted to also minimize the complexity of the trees. Therefore, if there wasn't much improvement in entropy, we decided to select the smaller model.

guage model). Therefore, an optimization criterion is needed to decide what is an appropriate phone recognition string. After much discussion, we decided that the goal/desired output of a phone recognizer is:

GOAL: To match the hand transcription phone labels

This is in contrast to our initial experiments, where we tried to optimize the phone recognizer to match the original dictionary pronunciations for the transcribed words. However, we decided that the ideal case would be to have an unlimited supply of hand labeled phone labels, and the closer we could come to this goal, the better.

The components of the phone recognition system were:

- Used context-dependent acoustic models trained on selected 49 hours of Switchboard data.
- Decision-tree based clustered acoustic models
- Phone transitions enforced clustering constraints
- Language model used: phone bigram
- Parameter Optimization: Reference Answers are Hand Labels
  - Phone Bigram Language Model Weight (10) & Insertion Penalty (0)

There were two possible reference answers that we considered using to evaluate performance:

- ICSI hand labels converted into pronlex phone set
- Words converted into dictionary pronunciation

4.1.2 Phone Recognition Results

The results of different phone recognition experiments are shown below in Table 2.

Experiment & Data Set	Reference Answers	Phones Correct	Phone Error Rate
Phone Rec Training	Dictionary	67.9	38.4
Phone Rec Testing	Dictionary	55.7	50.4
Hand Labels on Test Set	Dictionary	78.0	35.2
Phone Rec Testing	Hand Labels	53.5	55.8

Table 2: Phone error rate for different experimental conditions

Note that performance when recognizing on the training set is higher than when recognizing on the test set since the acoustic models were trained on that data. We should also note the similarity between results when dictionary is used as reference answer (50.4% WER) and when hand labels are used (55.8% WER).

4.0 EXPERIMENTAL RESULTS

During the workshop we made 4 iterations where we trained decision trees to model the pronunciations. A summary of the experiments is shown below in Figure 10.

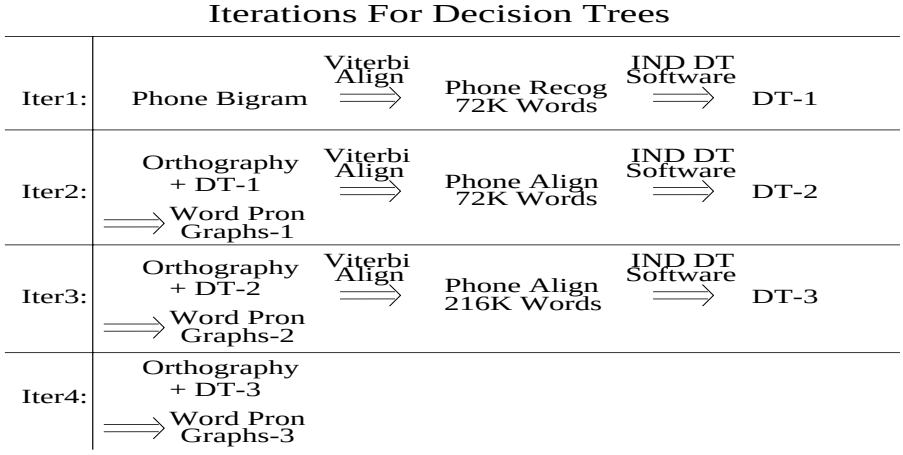

Figure 10: Experiments run at summer workshop

Many of the experiments in this Section will refer to Figure 10. DT1 will be used to denote the first set of decision trees that were trained as shown on the first line in Figure 10. The details of how these decision trees were trained was described earlier in Section 3.4. Experiments will be described that use each of the decision trees (DT1, DT2, and DT3). Since DT1 was used to create DT2, we refer to this as iterative training of the decision trees.

This section will start out by describing the results of the phone recognition experiments. Then we will describe some properties of the decision trees and how the resulting decision trees compare to the original dictionary.

Instead of using decision trees, we can alternatively use the pronunciations generated by aligning the dictionary entries with a phone recognizer directly. The procedure for generating this probabilistic dictionary will be described as well as some sample pronunciations generated from using such an algorithm. We will also describe LVCSR performance with this dictionary.

Finally, we will describe the use of decision trees. We start out with some diagnostic studies to understand the effects of different types of algorithms on the current 1-best hyp as well as the correct hypothesis. This is also accompanied by LVCSR performance with different experimental systems.

4.1 PHONE RECOGNITION

4.1.1 Goal of Phone Recognizer

There are a number of parameters in designing a phone recognizer (e.g. weight of bigram lan-

Equation 4:

$$\text{Prob}\,(p_n \mid l_n\,,\, p_{n-1}\,,\, p_{n-2}) \;=\; \frac{1}{Z\,(p_{n-2}\,,\, p_{n-1}\,,\, l_n)} \prod_{i=1}^{K} \alpha_i^{f_i\,(p_{n-2}\,,\, p_{n-1}\,,\, l_n\,,\, p_n)}$$

where the f_i 's are features indicator functions, one corresponding to each of the constraints, and Z is a normalization factor. Building a maximum entropy model is therefore equivalent to computing the model parameters α_i such that the constraints are satisfied.

Several algorithms are known for computing these parameters. One such algorithm was implemented, with a fairly flexible and problem-independent interface, at the workshop by Eric Ristad. The implementation is available as a general purpose Maximum Entropy Modeling Toolkit. We use this tool for computing the model parameters.

3.7.3 Implementation and Model Size Details

The decision tree leaves had 1328 leave labels, and 186 distinct observed phones.

The phone+leaf 4-gram model was built using:

- 1,491 phone unigrams
- 17,192 leaf->phone bigrams
- 58,785 phone+leaf->phone trigrams
- 51,528 phone+phone+leaf->phone 4-grams.

The maximum entropy model was built using:

- 60 phone unigram constraints
- 1,383 phone bigram constraints
- 16,859 phone trigram constraints
- 3,929 leaf->phone (DTree) constraints

The two models resulted in improvement of the acoustic pronunciation score per frame of the correct utterance relative to the baseline recognizer's best hypothesis, thereby suggesting that they may help with the overall recognition performance. The models were also used for rescoring the development-test set in an N-best framework. The recognition results are given in Section 4.

a 3-gram, using only the previous realized phone, and then to the bigram, which is the same as the decision tree probability.

The drawback of this model, however, is that it may cause data fragmentation in interesting cases. *e.g.* the fact that the previous phoneme was deleted may have the same effect on the /IH/->NULL transformation in many contexts of /IH/. This knowledge, however, is scattered over different 4-grams such as

sil NULL (1200) NULL and sil NULL (1198) NULL

which correspond to different leaf labels of /IH/. This could lead to unnecessarily inaccurate estimates of model probabilities.

An alternate method based on the maximum entropy principle, that avoids such fragmentation, is described next.

3.7.2 A Maximum Entropy Model

Consider a model for assigning probabilities to the realization p_n of the n-th phoneme in a context given by it's leaf label l_n, when the preceding phones are p_{n-1} and p_{n-2}

Equation 3: $$Prob\,(p_n \mid l_n\,,\, p_{n-1}\,,\, p_{n-2})$$

Based on the observed 4-grams in the training corpus, we first extract a set of "constraints" that we would like our model to satisfy. Amongst all the models of the above form that satisfy these constraints, we then select the model with the maximum entropy. The constraints include:

- DTree Constraints: For every leaf in the decision trees, if a phone was observed at least once in that leaf, the (marginal) conditional probability assigned to the phone, given only the leaf, is constrained to equal the probability assigned to the phone by that leaf of the decision tree.
- Phone Ngram Constraints: The marginal probability of p_n being any particular phone is constrained to be the relative frequency of that phone. The (marginal) joint probability of $(p_{n-1}\,,\, p_n)$ being equal to a particular phone `bigram' is constrained to be the relative frequency of the bigram for all frequent bigrams. Similarly the joint (marginal) probability of $(p_{n-2}\,,\, p_{n-1}\,,\, p_n)$ is constrained for all high-count phone `trigrams'.

Note that this way of constraining the model overcomes the fragmentation problem. It enforces the collocation of all frequent realizations. It additionally enforces the probabilistic structure that would result from using the decision trees alone.

The model which satisfies such marginal constraints and maximizes entropy has the well known parametric form

In the example of Figure 9, for instance, it is observed that amongst all the occurrences of /B/ in the context (identified by the decision tree leaf label) 385, followed by /IH/ in the context 1200,

Independent of Previous Context

Prob(/IH/ -> ih) = 0.61

Dependent on Previous Context

Prob(/IH/ -> ih | /B/ -> b) = 0.76

Prob(/IH/ -> ih | /B/ -> NULL) = 0.39

Independent of Previous Context

Prob(/IH/ -> NULL) = 0.13

Dependent on Previous Context

Prob(/IH/ -> NULL | /B/ -> b) = 0.02

Prob(/IH/ -> NULL | /B/ -> NULL) = 0.31

The above probabilities show that the realization of one phone is dependent on the realization of previous phones. If we ignore this context, we are losing information about the constraints and our probability model will not accurately reflect the true pronunciation probabilities. The way that we model this dependence is described in the next section.

3.7.1 A 4-gram Model of Phone Dependence

We construct “4-grams” that consist, in order, of the realization of two preceding phones, a reference phoneme in its context (identified by its decision tree leaf label) and the current realization of the phoneme, from our training data. We then infer a standard backoff 4-gram language model for assigning a probability to a realization given it’s reference phoneme (in context), and the realizations of the two previous phonemes. We call this the phone+leaf 4-gram model. e.g. In Figure 9, the realization of phoneme /IH/ in the context 1200 as the phone ih, when the previous phoneme /B/ is realized as b gives rise to the 4-gram

sil b (1200) ih

and when /B/ is realized as NULL, to

sil NULL (1200) ih

In the previous section, we showed how the model for the realization of a particular phone is conditioned on how previous phones are realized. However in the current model, we are only interested in the actual realizations of the previous phone, and do not condition on the previous node.

This model, clearly, generalizes the dependence assumptions of using only the decision trees -- the case where only the decision tree is used for assigning a probability is the bigram version of this model.

This model also has a systematic backoff structure: if a phoneme->phone transformation was seen with the given preceding realizations often enough, the observed 4-gram probabilities from the training data are used. If the given preceding phones were not seen, the model backs off first to

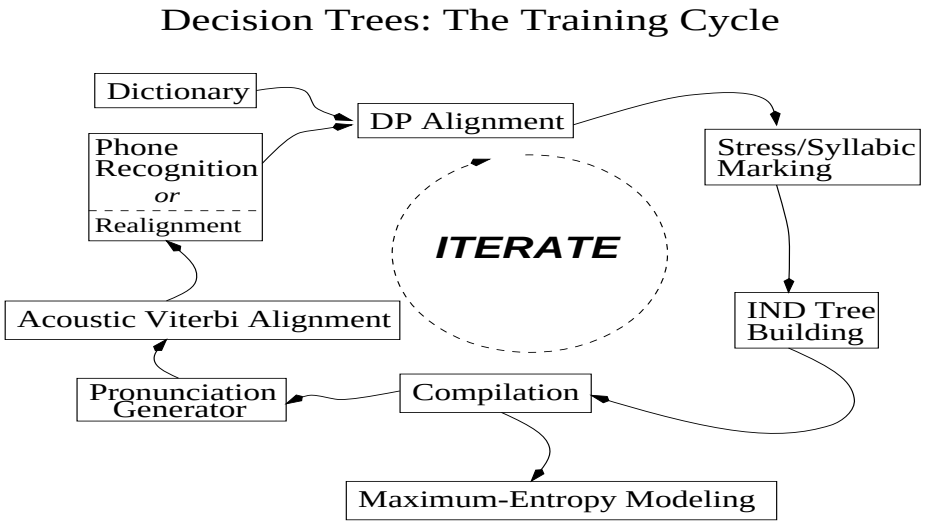

Figure 8: The training cycle for decision trees when using multiple iterations.

3.7 Maximum Entropy Modeling

The decision trees are used to compute estimates of the observed phones given the baseform phones. However, the realization of one phone is correlated with the realization of previous phones. If we assume that these processes are independent, then we are losing information that is useful for constraining pronunciations. An example of this process is shown below in Figure 9.

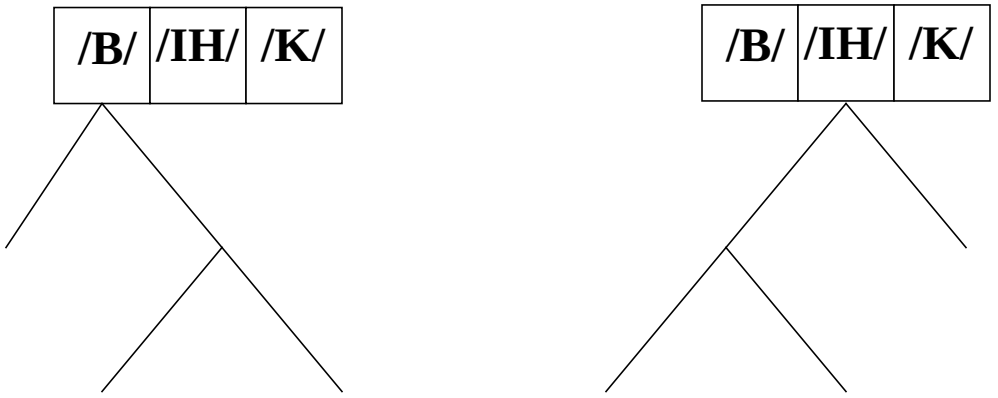

Figure 9: Decision trees for each phone applied to a phone sequence “/B/ /IH/ /K/”. Depicted are: (a) the left decision tree is for the /B/ phone, and (b) the right decision tree is for the /IH/ phone. The leaves depicted are those that are appropriate for this phone context. The node #’s are the unique indexes of the leaves of the different trees. The values at the leaves are the probabilities and output phones.

To get the pronunciation probability of a pronunciation hypothesis, the probabilities at each of the leaf nodes are multiplied together. The HTK stack decoder is used to generate the top N pronunciations for each word in the context of a particular sentence hypothesis.

Each pronunciation is labeled with (a) the sentence hypothesis index, (b) the word index in the sentence hypothesis, and (c) the rank of this pronunciation within the word. This information is written out to a file (with pronunciation probabilities) and then another program constructs the pronunciation graph in a format that HTK can use for the decoder.

3.6 Training with Multiple Iterations

The first iteration as described in the previous sections trained decision trees to predict the recognized phone sequence given the dictionary phone sequence. However, there is another way that the decision trees can be used. The original phone recognizer used context-dependent phone models connected with a bigram grammar. We can use a different grammar, as shown below in Figure 7.

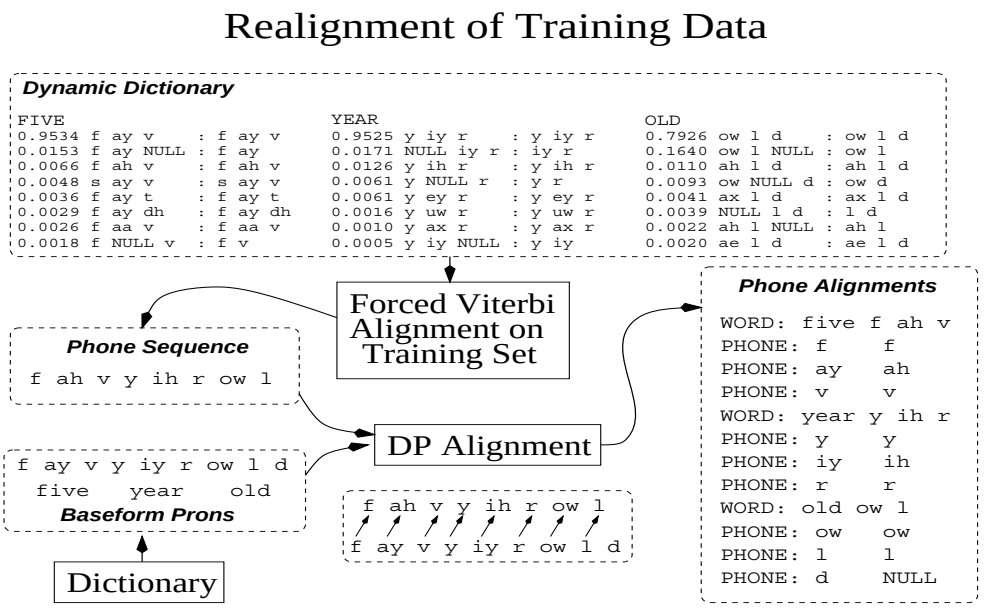

Figure 7: Realignment of the training data using phone sequences from new dictionary

The previous iteration is used to define a new dynamic dictionary. Using this dynamic dictionary, we find the best alignment of this grammar with the speech waveform using the Viterbi backtrace. The resulting phone sequence will be the best guess as to the actual pronunciation of these words using the previous iteration's dictionary. The iteration cycles are shown in Figure 8.

3.5 Using Decision Trees

The decision trees are applied to the lexical dictionary phone sequence as determined by the Viterbi alignment of the sentence hypothesis. Each phone of this sequence has the stress and syllabic information added to the sequence, as well as the answers to the 140 linguistic questions. For each phone in this sequence, we search the decision tree (based on which branches are appropriate from the answers to the 140 questions) and we arrive at a leaf node of the decision tree. At the leaves of the decision tree are a probability distribution over the pronunciation sequences for this phone. This is illustrated in Figures 5 and 6.

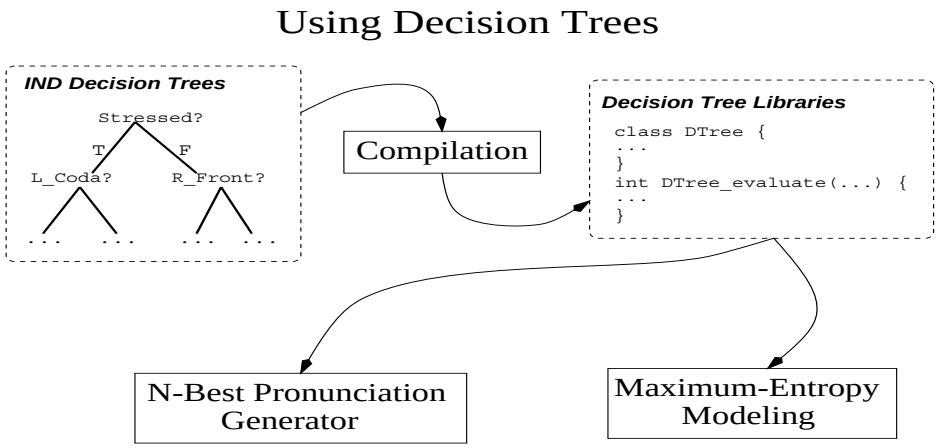

Figure 5: The IND decision trees are compiled into a program to generate pronunciation graphs.

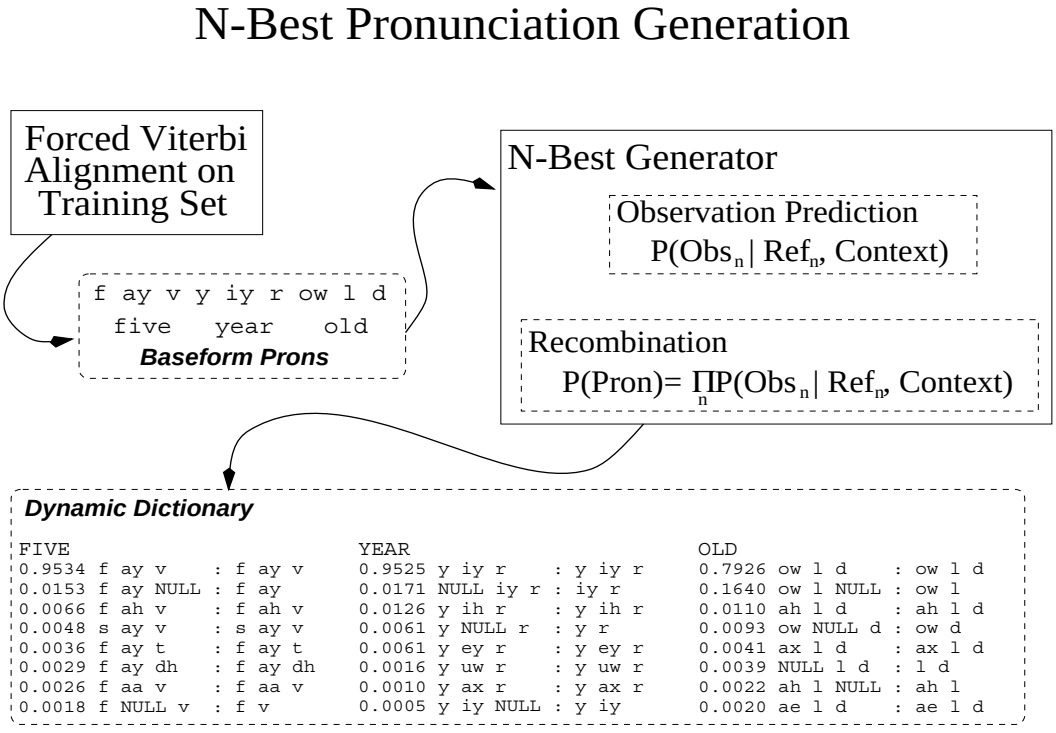

Figure 6: The generation of the top N pronunciations for each word

To train decision trees, we used a procedure as shown below in Figure 4.

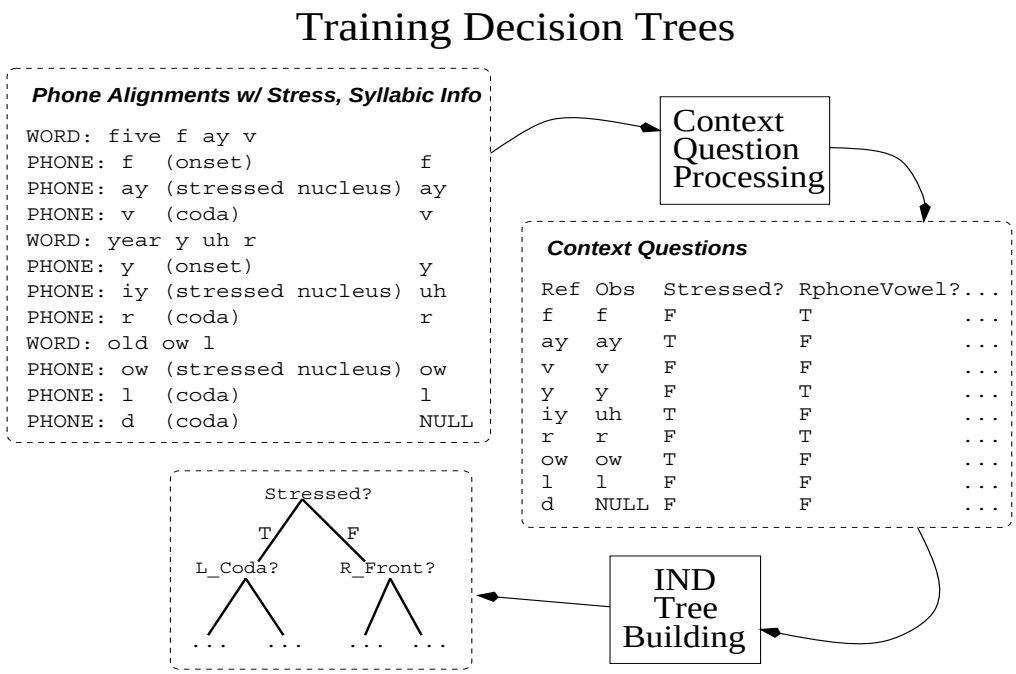

Figure 4: Building decision trees using the IND software package

We built one tree for each reference phone using IND. The trees were trained from alignments of a dictionary's prediction of the pronunciation with the observations derived from phone recognition. At each leaf of a given phone's tree is a probability distribution on the observed phones, which is used as input for the dynamic dictionary. We built what IND called "bayes" style trees, which means that the leaf probabilities are smoothed with the root node probabilities.

We asked 140 questions about the lexical reference phones: 126 questions were from the HTK state clustering, 7 questions were derived from ICSI's and SRI's phonological rules, 7 questions about stress and syllabic position. All of the questions were asked about the right and left neighbors of the reference phone, but only the stress and syllabic position questions were asked about the central, reference phone.

The trees were pruned to optimize perplexity as measured on a test set held out from each phone's data. This was done as a separate automatic, post-processing step.

with the speech data). The pronunciations in the dictionary are also time-aligned with the speech data. These two time-aligned phone sequences are then aligned. This alignment procedure uses a dynamic programming algorithm to find the best alignment between the two phone strings. The constraints placed on this alignment procedure are:

- Each phone is defined/replaced by a set of linguistic features (e.g. rounded, vocalic, high, low, nasal, ...). The "distance" between two phones is the city block distance between their two feature sets.
- There is a limit imposed based on the time alignments between two phones. Two phones must be within N msec of each other before the system can consider aligning them. Experimentally, 100 msec worked well. This means that the difference between the (start or end) of one phone must be less than 100 msec from the (start or end) of another phone.

The output of the alignment program is shown in the right hand side of Figure 2. The pronunciation (from the phone recognizer) that was aligned to each phone as well as to each word is output. To extract a pronouncing dictionary directly, it was straightforward to write a gawk/pearl script to extract out the multiple pronunciations for each word and set thresholds and compute apriori probabilities of each of the multiple pronunciations.

In addition to the alignment of these baseline phones (from the pronouncing dictionary) with the observed phone sequence (from the phone recognizer), stress and syllabic position information was added to the alignment data as shown in Figure 3.

Preparing Decision Tree Training Patterns II

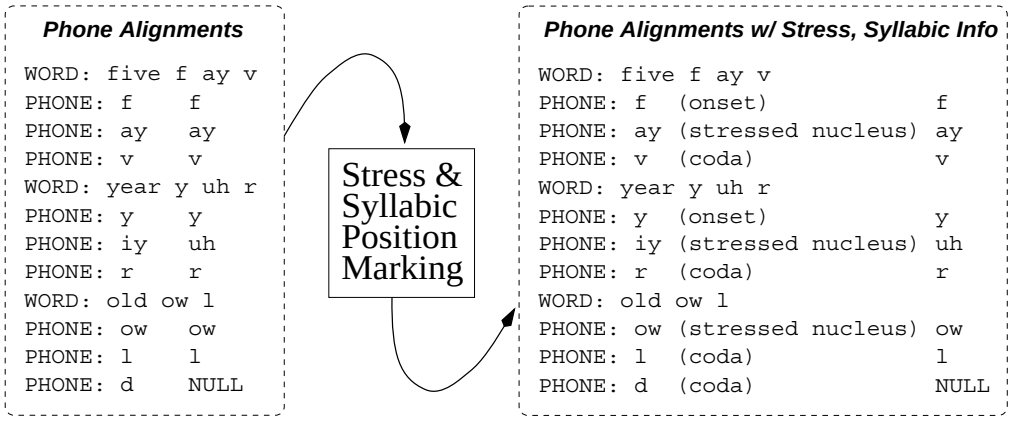

Figure 3: Adding stress and syllabic position information to decision tree input data.

A dictionary was developed for the workshop that contained stress and syllabic position information. This information was then used by aligning the relevant phones with the dictionary entries, and inserting this information into the alignment data file. The sample output on the right hand side of Figure 3 shows this information inserted into the alignment file.

4. replace each word with lexical dictionary phone sequence as determined by the Viterbi alignment in step 3.
5. Apply decision trees to the phone sequence of stage 3. The result of this stage is that each phone is replaced by a sequence of leaf nodes from the decision tree. Leaf nodes at the decision tree can contain possible phone strings (that were generated by the phone recognizer) that are NULL or that contain multiple phones (which allows for phone insertion). (See Section 3.4 for the training of these decision trees).
6. Extract the top-N pronunciations for each word. This uses the probabilities at each of the leaf nodes from the decision tree.
7. Result is a list of pronunciations (with associated probabilities) for this word in the context of a particular position in a particular sentence hypothesis.
8. Apply N-gram constraints on phone sequences (both with word as well as across words) to further constrain pronunciation probabilities.

The above list of steps is a summary of the stages that were developed at the workshop to model pronunciations. The following sections will describe some of these stages in more detail. Before we can get to applying the different modeling algorithms, we will describe how we prepared the data and trained our models.

3.4 Training Decision Trees

An overview of the data preparation for training decision trees are shown below in figures 2 and 3.

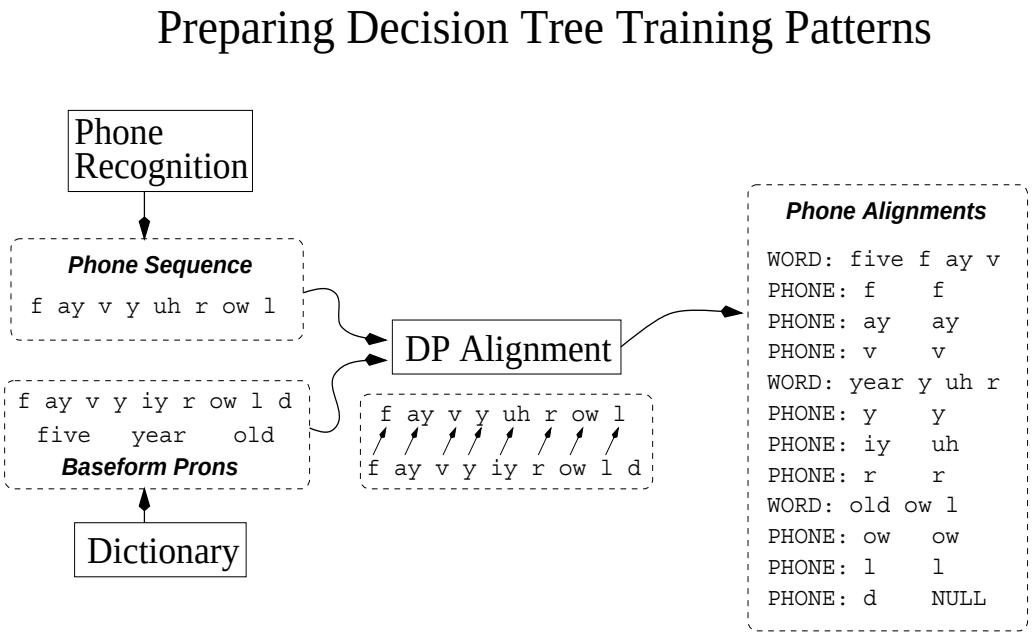

Figure 2: Alignment procedure to prepare data for decision tree training.

The output of the phone recognition is a sequence of phone labels with time marks (alignments

3.2 Dynamic Dictionary

The approach that we decided to use for the workshop is to use an N-best list rescoring paradigm. The HTK baseline system was used to construct a list of the best N unique sentence hypotheses. A new pronunciation model was constructed for each sentence, and a new acoustic score generated for each hypothesis. This new acoustic score was combined with the trigram language model score to form a new score for the sentence hypothesis. After reranking the hypotheses, the best scoring hypothesis is selected and scored against the correct words for that sentence.

When computing a model for a particular sentence hypothesis, we used a procedure that we called a *dynamic dictionary*. We used the name *dynamic dictionary* because the pronunciation for a word was dependent on the neighboring word context for the current word.

Each of the potential pronunciations for a word are listed as separate entries. There are language model transitions from each of the pronunciations of the previous words to the current word. These probabilities are computed as:

Equation 2: $$P(Pron_i \; for \; Word_j \mid Pron_k \; for \; Word_{j-1})$$

If there is no dependence of the pronunciation of the current word on the previous word, then the pronunciation probabilities would be treated like a standard probabilistic dictionary.

3.3 Computational Model

This section describes how we compute the pronunciation model for each sentence hypothesis. There were two basic experiments performed at the workshop:

- Use a phone recognizer to determine the possible pronunciations and their probabilities for the most common words.
- Use a decision tree applied to the pronunciation baseform to generate a dynamic dictionary and their associated probabilities.

The steps for applying the decision tree to the pronunciation baseform and generating the pronunciation for a sentence hypothesis are:

1. Look up each word in the N-best hypothesis in the original pronouncing dictionary
2. Construct a pronunciation graph of the original pronunciations. Each of the pronunciations in the original dictionary has an optional silence or sp phone added to the end of it.
3. Compute a Viterbi alignment between the original pronouncing dictionary and the acoustic data for each sentence hypothesis. This Viterbi backtrace will yield two pieces of information: (a) whether there was silence between the neighboring words in the hypothesis (this is necessary for cross-word pronunciation modeling) and (b) which of the possible original lexical baseforms we should use to apply decision-tree rules to (see Section 3.5 for application of decision trees to lexical baseforms).

3.0 WORKSHOP APPROACH

The approach that we are taking in this workshop is to compute a probabilistic pronunciation model for each word. The pronunciation models will be generated by using decision trees, and will be trained automatically using data. By using decision trees to generate the pronunciations, we will be able to generalize the pronunciations to words that we have not seen or seen only a few times in the Switchboard corpus.

This technique of generating pronunciation models will be compared with an algorithm that generates pronunciation models directly (from phone recognition) for the most common words. By generating the pronunciations for each common word, we are no longer able to (a) predict the dependence of the pronunciation on other factors (e.g. neighboring words, or global factors such as dialect) or (b) extend this pronunciation model to words that have little or no training data.

3.1 Overview

To predict the probability of observing a set of observations given a sentence hypothesis, we use:

$$P(Feature|Words) = \sum_{Pron} P(Feature|Pron)\,P(Pron|Words)$$

Equation 1

The first term in Equation 1

P(Feature Vectors | Pronunciation Phone Sequence)

is straightforward to compute. For each phone, we substitute the appropriate context-dependent phone models and compute the probabilities from the state output distributions. The second term in Equation 2

P(Pronunciation Phone Sequence | Word Sequence)

requires a model to generate the pronunciation probabilities given the word sequence. If we are using a probabilistic pronouncing dictionary, we can substitute each of the pronunciations word by word. If we are using a decision tree, we can generate the pronunciation graph with associated probabilities dynamically.

A difficultly in computing Equation 1 is the sum over all pronunciations. If two pronunciation arcs in a graph will be merged, the probabilities for the different paths cannot be summed unless both paths correspond to the same sentence hypothesis. This can only be accomplished by using a stack decoder or by having each sentence hypothesis searched separately. To simplify Equation 1, we will replace the sum over all pronunciations with the max probability of the best pronunciation.

- Pronunciations that occur in specific words or contexts will not affect other models that share the same triphone context but not these pronunciation variations.
- When the models are trained and tested with pronunciation models, each of the models will be sharper, and will better correspond to the linguistic identity of that phone.

While we were aware of the importance of retraining the HMM models with the new pronunciation models, logistically we did not have the time, manpower, or CPU power to accomplish these experiments at the summer workshop. Instead, we focused our efforts on developing techniques to model pronunciations with a fixed set of acoustic models.

by allocating some Gaussians to model these differences.

If the observed pronunciations match the lexical model used by the LVCSR system, then the Gaussian mixtures for each phone will model the acoustics of that phone in the lexicon. However if there is a mismatch between the observed phone sequence and the lexical phone pronunciation, then when the phone models are trained (e.g. with the forward-backward algorithm), the data aligned to this faulty lexical phone will come not only from this phone but also from other phones that were not present in the lexical pronunciations. The reestimation process for the Gaussian mixture models will receive training data from several different phones, resulting in changes in the Gaussian mixture model parameters (changes to the means, variances, and mixture weights).

Each of these three types of phonetic changes (substitution, insertion, or deletion) result in phone models being blurred and made more confusable. The acoustic model that corresponds to a phone no longer represents the perceptual phone, but instead represents the actual acoustic realizations of this phone.

Decision trees that are constructed by asking linguistic questions about the neighboring phones can help to alleviate some of these issues. For example, if phone P1 is occasionally realized as phone P2 in certain contexts, constructing a decision tree for the acoustic realization of phone P1 can result in a cluster of the appropriate phone contexts, thereby decreasing the problematic effects of having P1 and P2 be confused in all contexts.

2.5.2 Advantages to Explicit Pronunciation Modeling

Current output distributions for an HMM state are quite broad. For example, in the "Shared Gaussian Continuous Densities" presented by Zavaliagkos and McDonough [3] for training Switchboard English acoustic models, a typical state has 2,000 nonzero mixture weights on a shared set of 20,000 Gaussians. Lack of appropriate pronunciation modeling is one of the responsible factors for the broadness of the acoustic distributions.

Experiments by Giachin et. al. [13] showed some of the advantages of using phonological rules:

> *The results show that, although the errors of the type correctable by the phonological rules are relatively rare, the phonological rule system performs significantly better. This is also due to the indirect benefit given by the rules during the training process. More than 50% of the performance improvement is actually due to the general improvement of the model set rather than to a direct effect obtained by applying the rules in recognition. ...*
>
> *Results show that the effectiveness of this approach is highest when the HMMs of the speech units attain the highest acoustic resolution.*

Some of the other advantages of explicitly modeling pronunciation variation are:

- Pronunciation and dialect variations can be explicitly modeled as a function of transitions to other phones. Speaker-specific pronunciations can be modeled.
- Within-word and across-word pronunciation effects can be modeled. If the pronunciation transitions are correlated with other pronunciations or other variables, these correlations can be explicitly modeled.

listic dictionary. This is illustrated in the example below. Let us consider the following observed phone sequence:

ae n ay w aa n t

Which of the following is the correct word hypothesis?

AND I WANT

AN I WANT

In order to determine the probability of each of the different hypotheses, we need to consider both the pronunciation probabilities and the language model probabilities. The word *an* is much more likely to be realized with the phone sequence "ae n" than the word *and*, but the word *and* is much more likely to transition to the words *I want* than the word *an*. Therefore, to arrive at the correct word hypothesis, we need a good pronunciation dictionary as well as a good language model.

2.4.3 Estimating Pronunciation Probabilities

There are very few viable approaches to estimating pronunciation probabilities. The most accurate linguistically is to obtain a large volume of speech hand-transcribed by trained linguists. Such an approach has limitations, however, because hand transcribing is:

- Expensive
- Slow
- Difficult
- Often inconsistent across different transcribers

A second approach is to use computers to transcribe large amounts of speech. The phone-based HMM models that computers use are perceptually based, but when used to phonetically transcribe speech they make "mistakes" (in the linguistic sense; in the HMM sense they are optimizing a specific criterion). The resulting phone labels are noisy and typically differ from those produced by a human transcriber.

The resulting transcription may or may not make linguistic sense. This depends on the type of grammar and the type of HMM acoustic models that are used by the computer. If the grammar is constrained (e.g. by applying phonological rules to a baseform to get a possible phone graph), then the computer is prevented from outputting any linguistically implausible hypotheses. If the grammar is relatively unconstrained (e.g. a triphone bigram grammar) where every phone can connect to every other phone, then it is possible for the computer to output phone sequences that "do not make sense" to a linguist, but that match the observed acoustics.

2.5 Implicit versus Explicit Modeling

2.5.1 Implicit Modeling of Phoneme Substitution, Insertion and Deletion

One of the most common types of output distribution used for context-dependent LVCSR system is the Gaussian-mixture. A mixture of Gaussians is a very powerful modeling tool. If there are slight variations in a phone model (e.g. variations due to speaking rate), these can be easily handled

acoustics. The resulting model does not necessarily represent true pronunciation variation (as determined by a linguist), since it may allow for acoustic realizations that are implausible but that match the observed data. In the limit, if we had extremely good context-dependent phone units, this approach would accurately model the true pronunciation variation that is observed in the language.

2.4 Distinguishing Between Words with Similar Pronunciations

2.4.1 Homophones

A homophone is a word that is pronounced in the same way as another word but that differs in orthography. For example, the words *to*, *too* and *two* are words that differ in orthography but are all pronounced the same way. How do listeners distinguish between these alternatives? People rely on context (syntactic and semantic) to distinguish the intended word from alternatives.

When we allow each word to have multiple pronunciations, we increase the confusability between the words. For example, let us look at two words: *an* and *and* in the original dictionary and in a dictionary that was developed during this workshop.

Original Dictionary

AN	**ae**	**n**			
AND	**ae**	**n**	**d**		

Probabilistic Dictionary

AN	**0.36**	**ae**	**n**	
AN	**0.23**	**ih**	**n**	
...				
AND	**0.41**	**ae**	**n**	**d**
AND	**0.15**	**ae**	**n**	
AND	**0.10**	**ih**	**n**	
...				

We can see that in the second probabilistic dictionary, the words *an* and *and* have pronunciations that make them identical. One would think that this would make the words more confusable. However, if these pronunciations and probabilities are accurate, then it is not the dictionary that is making the words more confusable, but the talker is making the words more confusable by pronouncing the words in a reduced manner. The dictionary is merely reflecting this fact.

However, if the probabilistic dictionary pronunciations or probabilities are not accurate, then it is possible that the dictionary could increase confusability and reduce LVCSR performance.

2.4.2 Interaction Between Pronunciation Probabilities and Language Modeling

In the previous section we described how words that are homophones need to rely on the language model to distinguish between them. This is also true for a multiple pronunciation probabi-

2.2.2 Letter-to-Sound Rules & Text-to-Speech

For languages with relatively regular orthographics, letter to sound rules can be used. These algorithms can be trained automatically using statistical decision trees [4] between letters and arbitrary phone units.

A second alternative is to use a text-to-speech system. One of the outputs of many commercial text-to-speech systems is a phone sequence, and this phone sequence can be substituted for the word's pronunciation.

2.2.3 Applying Rules/Decision Trees to Baseform Pronunciation

This technique typically relies on using one of the previous two techniques to generate the pronunciation baseform. Then, an algorithm is used to generate multiple pronunciations by rule [16-18, 26, 23, 24], or from a decision tree [4, 7, 14]. These techniques can either be used in training, recognition, or in both cases.

2.2.4 Algorithms that Use Acoustics

In addition to those algorithms that use non-phone units (Multone/Senone graphs), other approaches use phone recognition [10, 11] or tied-rule pronunciation estimation with the forward-backward algorithm [17] to estimate pronunciation probabilities. All of these techniques typically prune low-probability pronunciations.

2.3 Context-Dependent Phone Units

It is well known that the perception of a stationary sound will change depending on the acoustic context. For example, one can delete a phone from speech and replace the missing segment with a loud noise and perceive the phone even though it is not there (this is known as *phonemic restorations* as demonstrated by Warren and Obusek, 71). This effect is especially relevant since some phones in a pronunciation are perceived by people listening to the acoustic signal, but a speech scientist that looks for the phone in the waveform or a spectrogram may have a hard time finding any evidence of it.

Pronunciations as transcribed by linguists are perceptually-based. The pronunciations that are used for words do not correspond to fixed acoustic units. To try to model this effect, speech engineers have developed context-dependent acoustic models [e.g. triphones]. These context-dependent phone units allow the phonetic identity to remain the same even though the acoustic realization will change based on the neighboring context.

A different approach to modeling words is to keep the acoustic units fixed. In this case, if the acoustic realization of a phone changes, it is modeled by a different set of units. Such an approach was developed at IBM with the use of fenones [5] and later multones [6].

The approach taken in this workshop is a hybrid between these two extremes. Like the first approach, we use context-dependent phone models (built using HTK decision trees) to account for the changing acoustic realization of phones. However, we also build pronunciation networks for each word in the lexicon based on automatic techniques of matching these units to the observed

summer's workshop, there are three entries for the word *the*:

THE	**dh**	**ah**
THE	**dh**	**ax**
THE	**dh**	**iy**

One typical property of most dictionaries is the lack of pronunciation probabilities. Each instance of the word "the" can be represented with any of the above phone sequences, with an equivalent probability of 1.0.

The other common approach is to use a pronunciation graph. The most common approaches to generating pronunciation graphs (most of these have pronunciation probabilities) are to:

- Use a baseform representation and apply a decision tree to generate alternate pronunciations [4, 7, 14].
- Use a baseform representation and apply a sequence of pronunciation rules to generate the possible pronunciation alternatives [16, 17, 18, 26, 23, 24].
- Merge multiple linear pronunciations together to form a pronunciation graph [20-22].

One should note that listing the pronunciations for a word may be adequate for isolated word recognition but this technique cannot account for any dependence of the pronunciation realization on the neighboring words. The technique of applying rules or decision trees to a pronunciation baseform can be used to generalize the pronunciations to account for the dependence of pronunciations on neighboring words.

2.1.4 Fenone/Multone/Senone Graph

A hybrid approach between those just described is to use a sequence of *tied* units. These units are denoted "Multones" [6] or "Senones" [9]. Whereas the HMM model topologies do not share any properties between words, these approaches can share output-probability distributions and training data across many different contexts. The decision of which acoustic units to use is determined automatically by a number of different possible algorithms.

2.2 Generating a Pronouncing Dictionary

2.2.1 Hand Generated Dictionaries

The most common method to generate pronunciation dictionaries is to use a trained linguist. The linguist will typically say the word out loud in different ways, and then write down the sequence of sounds that he hears. In this approach, the units that are used are perceptually based, as the linguist uses his acoustic perception to determine the sequence of units used. Typical dictionaries are only in citation form, and do not try to specify all possible variants found in natural conversational speech.

One limitation of this approach is that the linguist may not conceive of all the ways a person might say a word. In addition, it is difficult for a linguist to estimate the pronunciation probabilities.

2.0 MODELING ISSUES

2.1 HMM Representation of A Word

Most of today's LVCSR systems use a sequence of HMM states to represent words. There are several approaches to representing words as a sequence of HMM states. These approaches are described below in the following sections.

2.1.1 State Sequence

Examples of word representations that use state sequences for representing word models are "digit" word models. Here, each word is represented by a number of states. This sequence of states is typically linear (where each state connects to one successor state) or where successive states can be skipped [5].

In this approach, states generally have no "meaning," but are simply used to represent the observed acoustic observations. This approach requires that enough training data be provided for each word to allow probability distributions to be estimated on the feature space for each state.

This type of model is limited to words that have a significant amount of training data. In addition, it is difficult to model cross-word acoustic or pronunciation effects with a linear state sequence. These disadvantages are offset by having very detailed acoustic models for the words that they represent.

2.1.2 Phone Sequence

All high-performance LVCSR systems currently use phone sequences to represent words. Each word is represented as a sequence of phones, and each phone is represented as a sequence of states. The advantages of representing a word with a sequence of phones are:

- Efficiency: Phone models that occur in different words can share acoustic training data, thereby allowing one to train models with fewer repetitions of each word.
- Unseen words: Systems constructed with phone models can be used to generate models for words with little or no training data. This is especially important when constructing very-large vocabulary systems.
- Cross-word co-articulation: Context-dependent phone models can be used to represent the acoustic production of words across word boundaries.
- Explicit pronunciation modeling: By representing models as a sequence of phones, the pronunciation for each word can be modeled explicitly.

2.1.3 Phone Graph

There are two common ways to represent multiple pronunciations for a word. The first is to explicitly list each of the pronunciations for a word. For example, in the dictionary used for this

- Speech rate
- Geographical region of speaker
- Age
- Race
- Gender
- Education

This research found that speakers tend to fall into phonological groups/clusters, which might allow for rapid adaptation of the pronunciation models to the individual talker.

1.2 Overview of Paper

Section 2 describes different modeling issues that are affected by pronunciation modeling. We review what approaches different researchers have taken to modeling pronunciation variation, and describe how the different modeling issues interact. The interaction between pronunciation probabilities and language model probabilities are described, and we summarize the benefits for explicit modeling of pronunciation probabilities.

Section 3 focuses on the techniques that we used in the workshop. We present the dynamic dictionary approach and describe how this interacts with decision trees and N-best pronunciation generation. We also describe how to train the different models and how to iterate the process of predicting the pronunciation probabilities.

Section 4 contains the experimental results. Results are described for relating phone recognition to hand-transcribed phone labels, and for experiments using decision trees to model either of these two processes. Recognition results are presented for a number of probabilistic dictionary techniques, as well as for some diagnostic studies that attempt to understand what factors affect LVCSR performance.

Section 5 summarizes the results that were obtained at the workshop, and provides some direction for future research.

Switchboard corpus the average word is 3 phones long, then 1 out of every 3 words will have a deleted phone. If we also assume that a word will be misrecognized when it's pronunciation does not correspond to the pronunciation listed in the dictionary, then this will limit our LVCSR performance at 33% word-error rate (WER). In addition, the stronger the language model is, the more this phone deletion will impact negatively on hypotheses that contain this word. It has been observed both at the workshop and at other sites that it is very difficult to get N-best lists with error rates much lower than 30%. Accurately modeling phone deletions is extremely important if we are to achieve WER that approach 10%.

1.1.2 Individual Differences in Phone Deletions

A series of studies were conducted at SRI early in the DARPA program to study the difference between read speech and spontaneous speech [15-18]. Three subjects were interviewed in a conversational manner. Their data was transcribed orthographically and sentences were extracted from this material. The subjects were brought back and asked to re-read portions of their original interview in three reading modes: fast, normal and slow. Each of the four repetitions (spontaneous, fast reading, normal reading, and slow reading) were hand transcribed phonetically.

The first set of experiments showed that each of the speakers had different behaviors comparing spontaneous speech and read speech, and is shown below in Table 1.

Speaking Style	**Percent of Phonemes Deleted**		
	Male 1	**Male 2**	**Female 1**
Spontaneous	18%	9%	8%
Fast Reading	15%	10%	4%
Normal Reading	9%	4%	3%
Slow Reading	4%	4%	2%

Table 1: Percentage of Phones Deleted for 3 Subjects and different speaking styles

> *People cover more linguistic material per time in spontaneous speaking than in normal reading. However, when instructed to read faster, they mostly increase rate by speeding up each segment that is spoken. This is in contrast to spontaneous speech, where fast rate is accomplished more by skipping segments*
>
> *With regard to selecting a procedure for recording read materials to train speech recognizers, normal reading may be the best for studies of phonetic durations and coarticulatory phenomena, but fast reading will most likely yield a better approximation of the phonological patterns of the speaker. No reading seems to yield both.* [15]

Other experiments were performed by Michael Cohen for his thesis [17] on the correlation between different pronunciation variations. He found that many pronunciation variations were significantly correlated with each other as well as with other factors such as:

1.0 INTRODUCTION

In order for large-vocabulary conversational speech recognition (LVCSR) systems to achieve high-performance, it is necessary for the modeling at all levels of the system to fit the data well. This includes language modeling, acoustic modeling as well as pronunciation modeling. These different system components interact in a number of ways that are described in more detail in Section two and three. Some of the most important variations of pronunciation modeling that are often not captured by today's modeling techniques are:

- High rate of phone deletions in spontaneous speech
- Individual dialectal differences in pronouncing words

1.1 Rate of Phone Deletions

During the summer workshop, we compared the ICSI hand labeled phone transcriptions [27] with the lexical phone representations used in the workshop dictionary. If there were multiple pronunciations for a word, the pronunciation that was the best match to the acoustics was selected. The results of this experiment are shown below in the following experimental diagram:

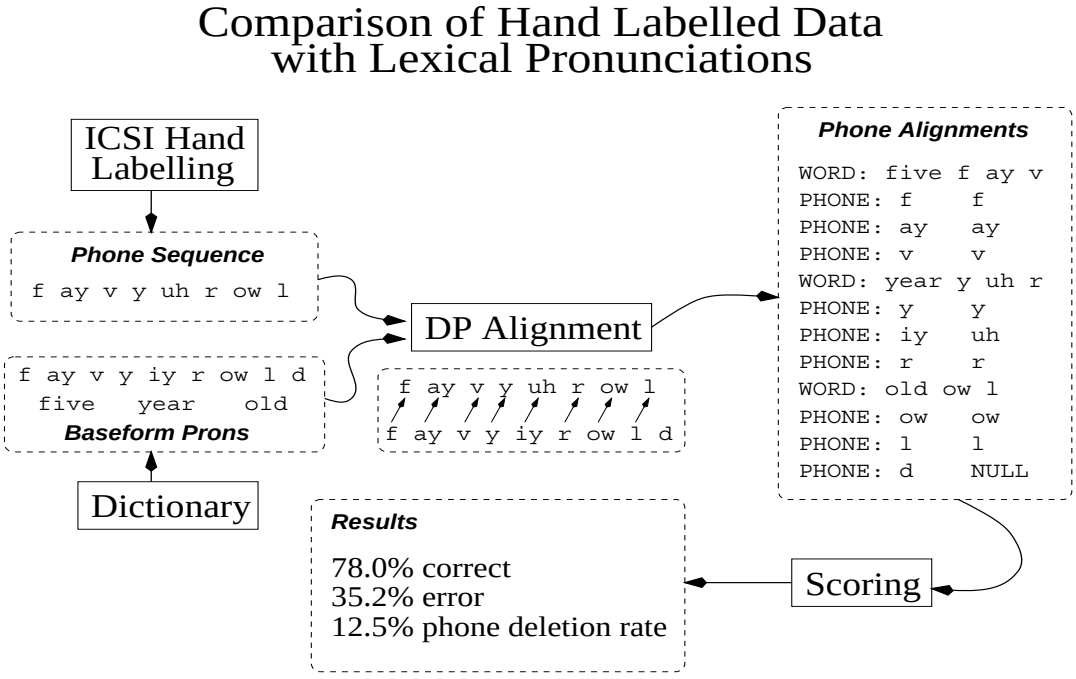

Figure 1: Comparison of Hand Labeled Phone Transcriptions with Phone Recognition Output.

1.1.1 Impact of Phone Deletions

The above experiment showed that 12.5% of all lexical phones are deleted. Since for the

Abstract

Today's recognizers are primarily based on single pronunciations for most words. This means that the burden of modeling phonetic variability falls entirely on acoustic modeling. In addition, certain types of pronunciation variation (phone deletion/reduction, dialect) are impossible to model well at the acoustic level. We suspect that one of the difficulties in recognizing conversational speech (compared to read speech) is the greater variability of pronunciation. We propose to capture this variability by modeling the pronunciations for each word.

The goal of this project is to automatically learn a model of word pronunciation from data. We focus on frequent words that appear many times in the Switchboard and Callhome corpora, since a small number of words make up a large fraction of the total errors. We can hope to learn these pronunciations automatically since these words occur many times in the training data.

All past attempts in this area have treated pronunciation variants as mutually independent, i.e., under the assumption that any speaker would choose one of the given variants with a given probability, independent of related choices in the same phonological context, conversation, by the same speaker. Such an approach is simple to implement, but increases the number of parameters and the phone-level perplexity of the model.

The advantage of learning word pronunciations automatically is that such a model will be a better fit to the observed data. However, the advantage of improved modeling must be accompanied by solutions to the following problems:

- The pronunciation probabilities (independent of other factors listed above) must be estimated accurately from a small amount of data, especially since there will be a great deal of variability in spontaneous data.
- The dependence of the current pronunciation probability on dialect and the pronunciation of neighboring words must be captured to increase the constraints that reduce word confusability.
- The increased degrees of freedom in the pronunciation model must be accompanied by a reduction in the broadness of the acoustic models that represent each phonetic segment in context.

The working goal of this group is to gain a better understanding of the modeling process (relationship between acoustic, pronunciation, and language) for speech. Some errors between actual pronunciations and the dictionary pronunciations of words can be compensated for by using large numbers of Gaussians in context dependent phone models. Similarly, some errors in the assumptions of the acoustic models can be compensated for by allowing for multiple pronunciation models. The language model is one way to impose constraints on pronunciations (constraints between words and dialectal constraints).

WS96 Project Report

Automatic Learning of Word Pronunciation from Data

Project Leader
Mitchel Weintraub (SRI)

Members
Eric Fosler (ICSI)
Charles Galles (DOD)
Yu-Hung Kao (TI)
Sanjeev Khudanpur (JHU)
Murat Saraclar (JHU)
Steven Wegmann (Dragon)

JHU Workshop 96
Pronunciation Group