AUTOMATIC LEARNING OF WORD PRONUNCIATION FROM DATA

Eric Fosler[*], *Mitch Weintraub*[†], *Steven Wegmann*[‡], *Yu-Hung Kao*[§], *Sanjeev Khudanpur*[¶], *Charles Galles*[‖], *Murat Saraclar*[¶]

JHU/CLSP Workshop '96 Pronunciation Group

ABSTRACT

The "Automatic Learning of Word Pronunciations" group at the 1996 Summer Workshop on Innovative Techniques for LVCSR focused on learning pronunciation variations as seen in the Switchboard corpus. In particular, the group explored methods of automatically learning word pronunciations that utilized constrained phone recognition.

1. INTRODUCTION

One of the issues in continuous speech recognition is how to model the bridge between acoustics and word sequences, that is, pronunciation variations. The choice of modeling technique depends on many factors, including vocabulary size, type of sub-word unit used to model acoustics, and mode of speech. Many recognizers use dictionaries that map words to one or more hand-written *baseform* pronunciations. Other work has tried to capture pronunciation alternatives either in a top-down fashion (e.g. using phonological rules applied to baseforms), or learning bottom-up through mechanisms such as decision trees.

At the start of the workshop, we investigated how our current baseform pronunciation models matched hand-labeled Switchboard data. We performed a dynamic programming alignment of the baseform pronunciations to the hand-transcribed phone sequence. The resulting alignment had a correct phone rate of 78% with an accuracy of 64.8%. It is interesting to note that there was a 12.5% phone deletion rate, meaning that relative to our "standard" baseforms, speakers were dropping 1 out of every 8 phones.

In order to model phenomena such as the phone deletion described above, the work described here uses data-driven learning techniques. Phonological baseforms were replaced with models that better matched the acoustic data of the training set, with the hope that this would reduce word error rates on the test set.

[*] International Computer Science Institute/UC Berkeley
[†] *Group Leader*, SRI International
[‡] Dragon Systems
[§] Texas Instruments
[¶] Center for Language and Speech Processing, Johns Hopkins
[‖] Department of Defense

2. APPROACH

In recognition, we compute the following likelihood for the acoustic score[1] (distinct from the language model score):

$$P(\begin{smallmatrix}\text{Acoustic}\\ \text{Features}\end{smallmatrix}|\text{Words}) = \sum_{\text{Pron}} P(\text{Features}|\text{Pron})P(\text{Pron}|\text{Words})$$

Computing $P(\text{Features}|\text{Pron})$ is straightforward; all that is required is the substitution of the appropriate context-dependent phone models; the probabilities are computed from the HMM state output distributions. In building a model to compute $P(\text{Pron}|\text{Words})$, the following desiderata were considered:

- The choice of pronunciation models should be data-driven—influenced by the acoustic models.
- The model for a word should be influenced by the surrounding context of pronunciations and words.
- The model should be modulated by lexical stress and syllabic constraints.
- The modeling should be as automatic as possible, requiring little to no human intervention.

We used a triphone-based phone recognizer[2] in order to determine to a first order which acoustic models matched a 10-hour subset of the acoustic training set. Using a dynamic programming algorithm that computed the distance between two phone strings, the phone recognition output was aligned to the reference baseforms for each transcript in the 10-hour set.

One product of this alignment was that every word in the training set had associated with it a new phonetic representation, that (usually) was a close variant of the baseform.

[1] We used a Viterbi approximation for this equation, replacing the sum with a maximum.

[2] This recognizer enforced triphone-clustering constraints during recognition.

In one experiment, we used these pronunciations as replacements for the baseforms in the dictionary of the recognizer, a process we termed *static dictionary replacement*. In order to handle noise, a threshold t was applied, so that any pronunciation variant that was not seen at least t times was discarded; if no variants exceeded the threshold, then the baseform was used. The new dictionary was then used to rescore n-best lists of hypotheses generated by the original recognizer by calculating for each hypothesis a new acoustic score and combining it with the old language model score.

The other product of the dynamic programming alignment between the baseforms and phone recognition was a mapping from each baseform phone to zero, one, or more observed phones from the phone recognition. Given this mapping, one can build a statistical model that predicts each observed phone given its baseform context. We trained decision trees to give probabilistic output distributions over observed phones, given the current baseform phone, and its neighbor on either side. The set of features presented to the decision tree included phonetic features, lexical stress, and syllabic position of the baseform phones.

In an n-best rescoring paradigm, we force aligned each hypothesis in the dev-test set to determine the best-matching baseform pronunciations for the hypothesis. The decision tree models were then applied to these baseforms to generate a *dynamic pronunciation graph* (where the pronunciation of each word is dependent on the surrounding words and their pronunciations), from which the m-best pronunciations for each word in context were derived. In addition, we sometimes incorporated n-phone-gram constraints on the observed phone sequences, either by interpolation or a maximum entropy model, in order to smooth the tree pronunciations.

3. EXPERIMENTAL RESULTS

In our first experiment, we created a new static dictionary, replacing frequent words in our static dictionary with pronunciation variants that occurred 7 or more times, deriving pronunciation probabilities based on the frequency counts of each word. Using this *initial static dictionary*, we rescored the 20 best hypotheses, reducing the word error rate from the baseline 46.4% to 45.5% on the 2116 sentence dev-test set, statistically significant at the $p < 0.05$ level.

At this point, we noticed that the short pause (sp) model in the baseline HTK-based system was actually modeling longer term acoustic events (up to 400 ms. long), so we removed the sp model from the lexicon, explicitly modeling pauses as longer-term silence models. Removing the sp model improved the baseline slightly to 45.7% on a random 200 sentence subset[3] of the dev-test (chosen to speed up evaluations). We considered the non-sp system to be the new baseline system.

[3] Rescoring the first experiment on just the 200-sentence subset kept the error rate at 45.5%.

Word Error Rate, Random 200 Sentences			
	n=20	n=75	n=100
Baseline WS96 System	46.4%	46.4%	46.4%
Initial Static Dictionary	45.5%		
No "sp" phones	45.7%	45.7%	45.7%
DT1 Static Dictionary	45.2%		
DT2 Dynamic Graphs	46.4%	45.5%	45.5%
DT2 Static Dictionary	45.3%		
DT3 Dynamic Graphs	46.2%	45.5%	
with Maximum Entropy	47.6%		

Table 1: Summary of Experimental Results

Using the initial alignment of the baseforms to the phone recognition, a first set of decision trees (DT1) were built. We realigned the training data using dynamic graphs generated by these trees, producing a new phone sequence that was then aligned against the baseform models. Experimenting with static dictionary replacement again (with threshold t=7) resulted in a further reduction of the error using 20-best lists to 45.2% error (DT1 Static Dictionary). Reducing the threshold t to 3 did not change performance significantly.

Iterating, we built a second and third set of decision tree models (DT2 and DT3) using the forced alignment from the decision tree pronunciations of the previous iteration. The static dictionaries from the DT2 alignment performed about the same as the DT1 alignments (45.3%). The dynamic pronunciation graphs constructed from each iteration were also used to rescore the 20-best lists; they performed slightly worse than the static replacement dictionaries and the modified baseline. Looking at larger n-best lists (n=75 or 100) improved performance, although not to the level of the static dictionary replacements. We also introduced n-phone-gram constraints with a maximum entropy model, but this raised the error rates above the baseline (47.6%), indicating that we need to look at this model more closely.

Due to space constraints, we are only able to present some of the work accomplished at the workshop. Our group looked at retraining acoustic models using the initial static dictionary, and also analyzed some of the models discussed above. In this analysis, we noted that mismatches between the phone recognition system and the baseline pronunciation dictionary are strong indicators of places where hand labeling also differs from the dictionary. Also, stress and syllabic position were often the best predictors of pronunciation variation.

4. SUMMARY

In our experiments, we found that replacing the frequent words in the static dictionary improved recognition by around 1% absolute from the initial baseline. Using decision trees to produce pronunciation graphs directly did not significantly change the error rate over the no-sp-phone baseline. None of these models were used to retrain the acoustic models; our current thinking is that retraining would also lead to improved performance.