To appear in ICLSP-94

MULTIPLE-PRONUNCIATION LEXICAL MODELING IN A SPEAKER INDEPENDENT SPEECH UNDERSTANDING SYSTEM

Chuck Wooters* Andreas Stolcke

International Computer Science Institute
1947 Center St. Suite 600, Berkeley, CA 94704, USA
{wooters,stolcke}@icsi.berkeley.edu

Abstract

One of the sources of difficulty in speech recognition and understanding is the variability due to alternate pronunciations of words. To address the issue we have investigated the use of multiple-pronunciation models (MPMs) in the decoding stage of a speaker-independent speech understanding system. In this paper we address three important issues regarding MPMs: (a) Model construction: How can MPMs be built from available data without human intervention? (b) Model embedding: How should MPM construction interact with the training of the sub-word unit models on which they are based? (c) Utility: Do they help in speech recognition?

Automatic, data-driven MPM construction is accomplished using a structural HMM induction algorithm. The resulting MPMs are trained jointly with a multi-layer perceptron functioning as a phonetic likelihood estimator. The experiments reported here demonstrate that MPMs can significantly improve speech recognition results over standard single pronunciation models.

1 INTRODUCTION

The lexicon for a speech recognition system is composed of a set of models that represent the pronunciations of words. Most current speech recognition systems use lexicons comprised of a single pronunciation for each word that is to be recognized. When one considers the variety of realizations that a word may have depending on such factors as its phonological context, the dialect of the speaker, etc., it seems obvious that word models that allow for multiple alternate pronunciations of a word should perform much better in a speech recognition system than single-pronunciation word models. For example, a single-pronunciation model for the word “the” can only represent either the pronunciation [dh iy] or the pronunciation [dh ax], where a multiple-pronunciation model (MPM) could represent both. Another major factor accounting for variation is the speaker’s dialect/idiolect. For example, “often” has two standard variants, one with a [t] and one without, apart from more subtle differences such as whether or not stops are realized with or without closures, are flapped, etc.

Despite the seemingly obvious advantage of MPMs, there has been conflicting evidence as to whether they can improve the performance of speech recognition systems. Some researchers [1] have not shown any improvements in recognition performance through the use of MPMs. Others [2, 3] have demonstrated significant improvements in performance.

There are several factors that likely contribute to these conflicting reports. One such factor is the difficulty in constructing the MPMs. For example, how does one derive alternate pronunciations for a word? How can we represent the fact that certain pronunciations are more likely than others? Our approach to these problems begins with the construction of an initial set of Hidden Markov Models (HMMs) representing alternate pronunciations for words, and then automatically adapts these HMMs in conjunction with the acoustic training of the sub-word unit models.

We have tested our approach within the context of the Berkeley Restaurant Project (BeRP) [4]. BeRP is a medium-sized vocabulary, speaker-independent speech understanding system whose domain is knowledge about restaurants in the city of Berkeley. The BeRP system is similar to other spontaneous speech understanding systems that have been developed recently [5, 6].

One of the distinguishing characteristics of BeRP is that it uses a speech recognizer that combines neural networks and Hidden Markov Models (HMMs). The neural network is a Multi-layer Perceptron (MLP) which is used to estimate the acoustic likelihoods for the HMMs [7].

In the next sections we present the details of this approach. Following that we give experimental results showing that MPMs do indeed improve recognition performance significantly in the BeRP system.

2 MODEL CONSTRUCTION

2.1 Constructing General Models

The first step in the model construction process is to create a general pronunciation model for each word. This general pronunciation model consists of a series of unique state sequences or paths: one for each of the alternate pronunciations of the word. The probabilities of each of the paths through the model are assigned uniformly, reflecting the fact that we have no information regarding the likelihood of these pronunciations. Figure 1 shows an example of a word model for the word “and” with three alternate pronunciations.

In order to construct the general pronunciation models, one needs to have a “source” for the alternate pronunciations which comprise the model. There are many pronunciation sources available including: pronunciations created “by-hand,” pronunciations from text-to-phoneme systems, or pronunciations from dictionaries.

2.2 Pronunciation Adaptation

Given a set of general multiple-pronunciation word models, the goal is to adapt these models to a particular application, while at the same time replacing the *a priori* estimates of the likelihood of each of the alternate pronunciations of a word, with estimates that provide a better match between the models and the training data. The two processes of adaptation and reestimation are carried out sequentially and may be iterated in order to further tailor the models to the speech recognizer and the training data.

Adaptation. The adaptation procedure begins with a Viterbi [8] alignment of the training data to the general word models. During Viterbi alignment, a single path representing one of the alternate pronunciations of a word is chosen for each instance of the word in the training corpus. This path represents the pronunciation that best matched the outputs of the phonetic likelihood estimator (the MLP) for that particular occurrence of the word. Some of the pronunciations that are represented in the general word model may never be chosen if they have a poor match to the outputs of the phonetic likelihood estimator compared to other pronunciations for the word.

Thus, the adaptation step produces a set of paths representing the pronunciations that had the best match between the outputs of the phonetic

*Currently with U.S. Dept. of Defense

likelihood estimator and the alternate pronunciations of a word. These sets of paths can then be used to reestimate the likelihood of each of the alternate pronunciations of a word as described in the next section.

Reestimation. The technique that is used to reestimate the probabilities of each of the paths through an HMM is based on an algorithm for automatically inducing HMM structure from a set of samples [9]. The algorithm begins with the construction of an initial HMM that just replicates the data (i.e. the paths representing the alternate pronunciations). Each path contains one state for each of the phonemes in the pronunciation. For example, Figure 1 shows the initial HMM for the following paths:

```
ae n
ae n
q ae n d
ae n d
ae n d
ae n d
```

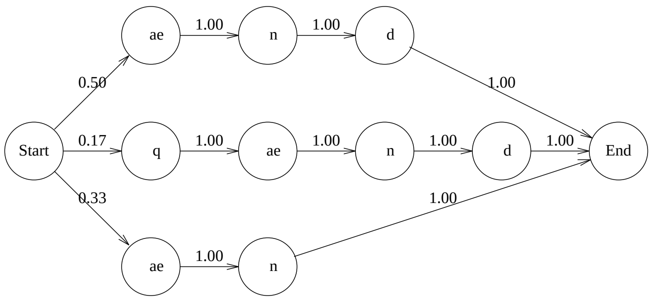

Figure 1: An HMM showing three possible pronunciations for the word "and." (The symbol "q" represents a glottal stop.)

In this HMM there are as many transitions out of the initial start state as there are unique paths, each one corresponding to an observed pronunciation. The probability of each of the paths is equal to their observed relative frequencies. Thus, since three of the six paths contained the pronunciation [ae n d], a probability of 0.5 is assigned to the transition from the initial start state to this path. It is easily seen that this initial configuration maximizes the model likelihood given the observed data.

Once the initial HMM has been constructed, it is made more compact, and possibly more general, by successively merging states.[1] (See Figure 2). The goal of the algorithm is to find an HMM structure that maximizes the model's *posterior probability*, which is given by Bayes' rule as

$$P(M|x) = \frac{P(x|M)P(M)}{P(x)} \quad (1)$$

(The denominator is constant given the data and can be ignored in the maximization.) The *model likelihood* $P(x|M)$ expresses the quality of the fit between model and data, and is maximized by the initial model, as we have seen. The *prior model probability* $P(M)$ represent a bias towards smaller models. While it would be conceivable to encode very specific preferences for certain model topologies etc. in $P(M)$, we only use a rather non-informative prior that generally favors models with fewer states and parameters.[2] The highest posterior probability is achieved by the HMM structure that represents the 'best' compromise between model fit and simplicity, according to the chosen prior.

In practice, we perform the posterior probability maximization in a greedy search that always chooses pairs of states for merging so as to give the currently best posterior. The parameters of the prior are tunable to determine the degree of generalization; in particular, the algorithm may be used to simply minimize the HMM structure without introducing new pronunciations. Once the HMM structure with a (locally) maximal posterior is found, the HMM parameters are set using standard maximum-likelihood estimation [10].

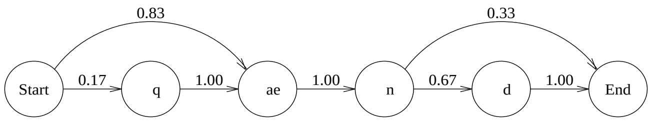

Figure 2: A merged HMM for the word "and."

One of the features of the HMM merging algorithm is that it can induce an HMM that is capable of generalizing to previously unseen pronunciations. For example, consider the following possible set of observed pronunciations for the word "have":

```
[hv ae v]
[hh ae v]
[hv ae f]
```

In this set of data, there are two phones that can occur at the beginning of the word – {hv,hh}, corresponding to a voiced /h/ and an unvoiced /h/ respectively. There are also two phones that can occur at the end of the word – {v,f}. There is one possible pronunciation – [hh ae f] that may not have actually been observed in the data, e.g., due to undersampling. Depending on the precise observed frequencies involved and the prior parameters used, the HMM merging algorithm can infer a generalized model structure that allows for the unobserved pronunciation and assigns it a probability based on its similarity with the observed pronunciations (see Figure 3).

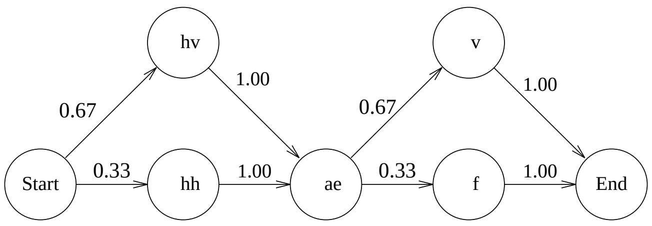

Figure 3: A possible HMM for the word "have" with a pronunciation – [hh ae f] that was not observed in the data.

3 MODEL EMBEDDING

The word models that result from the HMM merging can now be used as input to a second Viterbi alignment. The output of this Viterbi alignment can be used as the input to HMM merging, and yet another set of word models can be constructed. This iterative process is shown schematically in Figure 4, and coincides with the iterative process used for the generation of training labels for the Multi-layer Perceptron (MLP). Thus the generation of multiple pronunciation word models is easily integrated into the MLP's training procedure.

The embedded training procedure needs initial MLP weights, as well as an initial source of pronunciations. The initial pronunciation inventory varies with the experiment, as described below. For the MLP, we generally use the weights from a network independently trained on the TIMIT speech database for bootstrapping. The same TIMIT weights are also used to initialize the MLP in each round of training, which we found to give better results than using the final MLP parameters from the previous iteration (this procedure should help avoid the danger of

[1]For MPM construction purposes, we only consider merging states that have the same labels and which preserve the left-to-right character of the HMM. In the full algorithm, there are no such constraints, and arbitrary HMM structure can be found in principle.

[2]This approach expresses our ignorance about 'correct' pronunciation models, but also leads to very efficient computational solutions. The prior used is a combination of a term involving the description length (number of bits needed to encode) the model's structure, and a Dirichlet prior over the transitions and emissions [9].

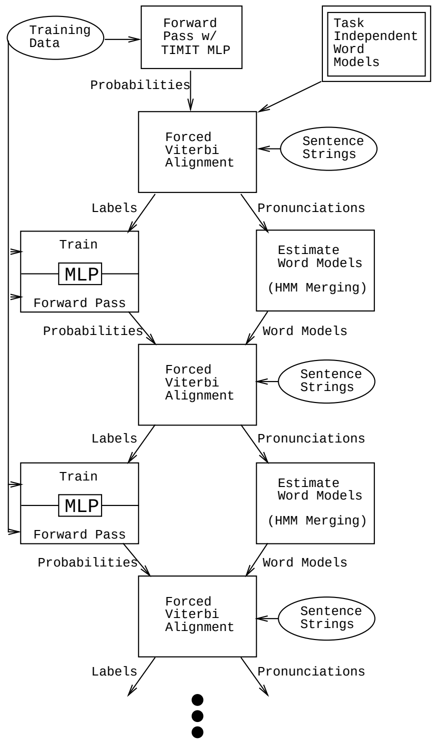

Figure 4: A schematic outline of the embedded training for MLPs and multiple pronunciation lexica.

reaching a local minima during MLP training). The MLP training uses cross-validation to avoid overfitting; 10% of the total training corpus are typically set aside for this purpose.

We can view this iterative embedded training procedure as an approximate version of the Expectation Maximization (EM) algorithm commonly used for maximum-likelihood estimation with incomplete data [11]. The Viterbi alignment step approximates the E-step, estimating the unobserved labels at each time frame, whereas the MLP training and word model induction realize the M-step, i.e., maximizing model parameters relative to the estimates of the E-step.

The implementation of both steps is inexact, however, for various reasons having to do with efficient implementation. Indeed, we have observed that the best results are obtained by combining the lexicon obtained in the first iteration with the MLP trained in the second iteration.

4 MODEL EVALUATION

4.1 Multiple vs. single pronunciations

Previous experiments using the approach described here [3] addressed whether multiple-pronunciation modeling in the BeRP system could give improvements over the traditional single-pronunciation HMM word models. This comparison was based on a 2,319 training utterance corpus and a 364 utterance test set. The test set was gathered from 8 speakers, 4 males and 4 females, each providing approximately 45 utterances. The speech is spontaneous and was not screened to remove any disfluencies or out-of-vocabulary words.

During Viterbi decoding (recognition) a simple bigram language model was used. The bigram model contained only observed word co-occurrence frequencies, without any smoothing.[3]

The single pronunciation (SP) version of the system was training using the embedded procedure, but keeping the lexicon fixed. The lexicon consisted of single pronunciations generated by a text-to-speech system. The multiple pronunciation (MP) version, on the other hand was bootstrapped from a collection of pronunciation sources, including the same text-to-speech system, the TIMIT database, a pronunciation lexicon from LIMSI-CNRS, the Resource Management pronunciations, and hand-crafted pronunciations (for some words). The MPMs obtained from the embedded training were slightly pruned by removing paths accounting for less than 10% of the total probability mass, which was found to improve performance somewhat.

System	Error Rate	Ins	Dels	Subs
SP	38.6	4.6	10.6	23.4
MP	32.1	7.3	5.9	18.9

Table 1: Performance of fixed single vs. multiple pronunciations

Table 1 shows the results from this comparison, in terms of word error percentage, and broken down by word substitutions, deletions and insertions. The improvement found with MPMs over the single pronunciation version is significant at the .01 level.

4.2 Multiple vs. most likely pronunciations

The SP lexicon used in the previous comparison was derived from an independent source and held fixed while adapting the other parts of the system. Other research [2] suggest that a substantial improvement can be gained by simply selecting the single *most likely* pronunciation for each word (as opposed to, say, one obtained from a dictionary).

Choosing the most likely pronunciation from a training corpus is a simple form of adaptation within the space of SP models. We therefore carried out a second experiment to see whether the the advantage of MPMs would hold up against the most-likely SP approach.[4]

The training corpus consisted of 912 sentences. There were 554 sentences in the test set. The word models induced from the training corpus were complemented by fixed, rule-based MP models for unobserved words, taken from a comprehensive default lexicon so as to avoid out-of-vocabulary words in the test set. These fixed default pronunciations were kept identical across all test conditions, and helped minimize recognition errors not directly related to the comparison of the learned word models. The lexicon used in bootstrapping was also based on dictionary pronunciations and rules for standard phonetic variations. The language model used for all recognition tests was a simple bigram as before, derived from an independent corpus.

The SP models were built from the training by choosing the most frequently observed label sequence for each word, as given by the forced Viterbi alignment. The MP models were obtained from merging the Viterbi pronunciations as described above, without additional pruning of unlikely paths. Two MP versions were tested: merging without generalization, i.e., the lexicon contained only observed pronunciations, and merging with generalization. The prior parameters, and hence the amount of generalization were not specially tuned for this task; we reused a configuration from a previous experiment involving TIMIT data [9]. The embedded training procedure was identical except for the use of different word models in each alignment step.

System	Error Rate	Ins	Dels	Subs
Most likely SP	27.1	3.8.	8.0	15.3
MP	25.2	4.4	7.1	13.8
Generalized MP	25.3	4.2	7.1	14.0

Table 2: Performance of most likely and multiple pronunciations

Table 2 gives the error percentages obtained using the three system configurations. As can be seen, the MP system still enjoys an advantage over the (most likely) SP version (significance $p < 0.05$). Generalization seems to have virtually no effect on this particular task, except for slightly reducing insertions at the expense of substitutions.

[3]For this reason it is not meaningful to calculate perplexity for this particular corpus, but in [3], the perplexity of a 227 sentence subset screened for out-of-vocabulary words was estimated as 10.6. Roughly one-third of the sentences in the test set had either out-of-vocabulary words or out-of-grammar word pairs. Thus, this task is much more difficult than this perplexity suggests.

[4]This experiment used a different corpus and bootstrap conditions than the previous one, and incorporated numerous unrelated improvements to the system; therefore the results are not directly comparable.

It is interesting to compare some statistics of the generated MP models. The total number of words in the training corpus, and hence word models, was 644. The simple MPMs had a total of 7205 states and 7679 transitions, with an average number of pronunciations per word of 1.56. As expected, the generalized models were smaller (5133 states and 5668) and produced more pronunciations (1.86 on average).[5]. The average number of pronunciations is generally low due to the large number of words (369) that have only a single pronunciation, many of which (269) appeared only once in the corpus.

4.3 Discussion

Multiple pronunciation modeling is interesting for both speech recognition and linguistics. Regarding the former, our experiments have shown that significant improvements over SP models can be realized using MPMs, as one would expect. From a linguistic perspective, the automated building of MPMs can be seen as a form of phonological learning; as such, it should be compared to alternative learning approaches, such as inducing pronunciations based on local phoneme contexts [12, 13].

Experiments on isolated TIMIT word pronunciations [9] have shown a significant improvement (measured in phone perplexity) when using the merging algorithm to infer new pronunciations from observed ones. The fact that generalization did not produce any improvements in the recognition experiments reported here could be an artifact of the small size or relative uniformity of the data; and needs further investigation.

It is important to realize that the underlying model for pronunciation variability used so far is quite limited. For example, we assume word pronunciations vary independent of their context. In reality there are obvious context effects, e.g., due to co-articulation involving adjacent phonemes from preceding and following words.

In principle this can be addressed by refining (at least some of) the word models to be specific to a phonetic context. However, in doing so care must be taken to not run into data sparseness problems.

The assumption that the various pronunciations of a word occur with fixed probabilities from one occurrence to the next is also flawed. At a minimum, one would expect systematic co-variation of all words of a given type, depending on such factors as speaker identity, accent, rate of speech, etc. We are currently studying adaptive approaches that change the prior probabilities of the paths in a word model based on estimates of the speaker's accent type, especially for native vs. non-native speakers [4]. More fine-grained synchrony among pronunciation variants is also likely, e.g., whether a speaker generally tends to flap certain stops. Modeling such subword regularities would require revising the straightforward word-based approach.

In summary, we believe that data-driven multiple pronunciation modeling is a rich subject for research and potential improvements of speech systems, and anticipate seeing more probabilistic modeling and learning techniques applied to this task.

Acknowledgments

Thanks are due all members of the ICSI speech group for helpful discussions and suggestions. Special thanks go to Gary Tajchman for providing the rule-based MP lexicon and general assistance in setting up the experiments.

This work was partially funded by ICSI and an SRI subcontract from ARPA contract MDA904-90-C-5253. Partial funding for the BeRP system development came from ESPRIT project 6487 (The WERNICKE project).

References

[1] K. F. Lee. *Automatic Speech Recognition: The Development of the SPHINX System*. Kluwer Academic Publishers, 1989.

[2] M. Cohen. *Phonological Structures for Speech Recognition*. PhD thesis, University of California, Berkeley, 1989.

[3] C. Wooters. *Lexical Modeling in a Speaker Independent Speech Understanding System*. PhD thesis, University of California, Berkeley, 1993.

[4] Daniel Jurafsky, Chuck Wooters, Gary Tajchman, Jonathan Segal, Andreas Stolcke, Eric Fosler, and Nelson Morgan. The Berkeley Restaurant Project. These proceedings.

[5] P. Price. Evaluation of spoken language systems: The ATIS domain. In *Proc. Third DARPA Speech and Language Workshop*, pages 91–95, Hidden Valey, PA, June 1990.

[6] V. Zue, J. Glass, D. Goodine, H. Leung, M. Phillips, and S. Seneff. The VOYAGER speech understanding system: Preliminary development and evaluation. In *Proceedings Int'l Conference on Acoustics Speech and Signal Processing*, pages 73–76, Albuquerque, New Mexico, 1990.

[7] H. Bourlard and N. Morgan. *Connectionist Speech Recognition A Hybrid Approach*. Kluwer Academic Publishers, 1993.

[8] H. Bourlard and C. J. Wellekens. Links between Markov models and multilayer perceptrons. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(2):1167–1178, 1990.

[9] A. Stolcke and S. Omohundro. Best-first model merging for Hidden Markov Model induction. Technical report, International Computer Science Institute, 1947 Center St., Suite 600, Berkeley, CA 94704, 1994. TR-94-003.

[10] L. E. Baum, T. Pitrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions in Markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970.

[11] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the *EM* algorithm. *Journal of the Royal Statistical Society, Series B*, 34:1–38, 1977.

[12] Francine R. Chen. Identification of contextual factors for pronunciation networks. In *Proceedings IEEE Conference on Acoustics, Speech and Signal Processing*, volume 2, pages 753–756, Albuquerque, NM, April 1990.

[13] Michael D. Riley. A statistical model for generating pronunciation networks. In *Proceedings IEEE Conference on Acoustics, Speech and Signal Processing*, volume 2, pages 737–740, Toronto, May 1991.

[5]The number of pronunciations is computed as the exponential of the entropy of the paths through an HMM, i.e., the 'pronunciation perplexity.'