LEXICAL ACCESS WITH A STATISTICALLY-DERIVED PHONETIC NETWORK

Michael D. Riley and Andrej Ljolje

AT&T Bell Laboratories
Murray Hill, NJ 07974

ABSTRACT

A probabilistic approach to lexical access from a recognized phone sequence is presented. Lexical access is seen as finding the word sequence that maximizes the lexical likelihood of a sequence of phones and durations as *recognized by a phone recognizer*. This is theoretically correct for minimum error rate recognition within the model presented and is intuitively pleasing since it means that the "confusion matrix" of the phone recognizer will be learned and its regularities exploited. The lexical likelihoods are estimated from training data provided by the phone recognizer using statistical decision trees. Classification trees are used to estimate the phone realiziation distributions and regression trees are used to estimate the phone duration distributions. We find they can capture effectively allophonic variation, alternative pronunciation, word co-articulation and segmental durations. We describe a simpified, but efficient implementation of these models to lexical access in the DARPA resource management recognitiion task.

1. INTRODUCTION

We describe a new approach to lexical access in a phone-based speech recognition system. By "lexical access" we mean taking a sequence (or, more generally, a lattice) of phones and durations that is output by a phone recognizer and mapping it onto a word sequence (or, more generally, a lattice).

In conventional word-based speech recognizers, segmental durations, word co-articulation and alternative pronunciations are usually poorly modelled if at all since the architecture is not convenient or efficient for exploiting these constraints.

Phone-based recognition offers an attractive alternative from this point of view. Our approach will be to create a probabilistic model that provides the likelihood that a particular word sequence gives rise to a particular phone sequence. This model will take into account allophonic variation, alternative pronunciation, word co-articulation and segmental durations.

We then combine these *lexical likelihoods* with the acoustic likelihoods generated by the phone recognizer and priors from our language model to get an overall recognition model whose error rate we seek to minimize.

We have taken this stochastic approach for two reasons. First, it provides a principled way to combine seemingly disparate information: (a) acoustic likelihoods, (b) segmental durations, (c) alternative pronunciations, and (d) the language model. Second, the availability of large speech corpora now allow the statistical estimation of these probabilities.

2. PROBABILISTIC MODEL

We form the probabilistic model as follows. Let $\mathbf{w}$ be a sequence of words, let $\mathbf{y}$ be a sequence of phones, let $\mathbf{d}$ be a sequence of durations, and let $\mathbf{s}$ be a (fixed) speech signal. Then

$$P(\mathbf{w}|\mathbf{s}) \propto \sum_{\mathbf{y},\mathbf{d}} P(\mathbf{s}|\mathbf{y},\mathbf{d})\, P(\mathbf{y},\mathbf{d}|\mathbf{w})\, P(\mathbf{w}). \qquad (2.1)$$

The lefthand side of this relation is the probability that a given speech signal corresponds to a particular word sequence. The word sequence that maximizes this term gives the minimum sentence error rate. The first factor on the righthand side gives the acoustic likelihoods provided by the phone recognizer. The second factor gives the lexical likelihoods to be provided by the lexical access stage describe here. The third factor represents whatever language model we use.

In this paper, we have used the output of the current Bell Labs phone recognizer as input to the lexical access component [1]. At present, this recognizer outputs a single sequence of phones and durations per utterance, which represents its best estimate of the true sequence. As such, $\mathbf{y}$ and $\mathbf{d}$ are fixed in Eq. 2.1 for a given speech signal. A more general approach, which would consider alternative sequences – phone lattices – is currently under investigation, but not reported here.

Also in this paper, in which we present results on the DARPA resource management task, we consider only the the simple word-pair language model. Thus, for a given utterance, the best scoring word sequence, $\mathbf{w}$, will be the one that maximizes the lexical likelihood, $P(\mathbf{y},\mathbf{d}|\mathbf{w})$ for a given phone recognizer output $\mathbf{y}$ and $\mathbf{d}$, and which is a legal sequence in the word-pair grammar. In this model, finding the word sequence that maximizes this likelihood is the goal of lexical access and estimating this likelihood is the goal of this paper.

A crucial factor for this estimation is that $\mathbf{y}$ and $\mathbf{d}$ are not the true sequence of phones and durations, but the output of a phone recognizer. As such, we must train our estimator on the *output* of the phone recognizer. This is theoretically correct for minimum error rate recognition in tnis model and is intuitively pleasing since it means that the model will learn the "confusion

matrix" of the phone recognizer and thus exploit its regularities. This combined with our probabilistic model differeniates us from other approaches to lexical access [2,3].

We can further decompose this problem by breaking the lexical likelihoods into two factors:

$$P(\mathbf{y}, \mathbf{d}|\mathbf{w}) = P(\mathbf{y}|\mathbf{w})\, P(\mathbf{d}|\mathbf{w}, \mathbf{y}) \qquad (2.2).$$

The first factor is the pronunciation model, which gives the probability of a phone sequence given a word sequence and the second factor is the segmental duration model which gives the probability of a duration sequence given the phone and word sequences.

Given a word sequence we can use a dictionary to look up the corresponding phoneme sequence [4]. We can then replace the word sequence **w** in Eq. 2.2 with the phoneme sequence augmented with word boundaries and lexical stress with little loss of information.

It is important not to confuse phonemes and phones at this point. A *phoneme* is a coarse description of the pronunciation of a word as usually found in a dictionary. A *phone* gives a finer description indicating how the speaker uttered a word in context. For example, the `/t/` in `'butter'` may be pronounced as a flap, `[dx]`, or as a released t, `[tcl t]` . In this paper, we use the TIMITBET symbols, a superset of the ARPABET symbols, for specifying phones [5]. Which phone will be the realization of the phoneme `/t/` in this word depends, in part, on the speaker's dialect and speaking rate. Nor is the phonetic realization of a phoneme always determinstic; only about 75% of the `/t/'s` in a similar context to `'butter'` are flapped, estimated from the TIMIT database. It is precisely the phoneme-to-phone mapping that comprises the pronunciation model that we are trying to generate.

Let us make this idea precise. Let $\mathbf{y} = x_1 x_2 ... x_m$ be the string of phonemes of some sentence. So that we can mark both word boundaries and stress we augment the phoneme set to include `/#/` as a word boundary marker and split each syllabic phoneme into an unstressed, a primary stressed, and a secondary stressed version. Further, let $\mathbf{y} = y_1 y_2 ... y_n$ be the string of corresponding phones. We include the phone symbol `[-]` to indicate that a phoneme may delete.

The most general form of our predictor is $\hat{P}(\mathbf{y}|\mathbf{x})$, where $\hat{P}$ estimates the probability that the phone sequence **y** is the realization of the phoneme sequence **x**.

This specifies the probalitity of an entire phone sequence **y**. For convenience, we want to decompose this into one phone prediction at a time. Since

$$P(\mathbf{y}|\mathbf{x}) = p_n(y_n|\mathbf{x} y_1 ... y_{n-1}) p_{n-1}(y_{n-1}|\mathbf{x} y_1 ... y_{n-2}) ... p_1(y_1|\mathbf{x}), \qquad (2.3)$$

we can restate the problem as finding a suitable predictor, $\hat{p}_k(y_k|\mathbf{x}\, y_1 ... y_{k-1})$, that estimates the probability that y_k is the kth phone in the realization, given the phoneme sequence **x** and the previous k-1 phones $y_1 ... y_{k-1}$.

Eq. 2.3 is more general than necessary since realistically the kth phone will depend only on a few neighboring phonemes and phones. Suppose that we can place the phoneme and phone strings into alignment. In fact, forming a good alignment betwen phonemes and phones is easy if deletions and insertions are permitted, using a phonetic feature distance measure and standard string alignment techniques [6]. Since we have augmented the phone set to include a deletion symbol, the only stumbling block to such an alignment would be if phones insert. For the moment, assume that they don't; we will come back to insertions later. Thus, under this assumption we can talk about the kth phoneme and its corresponding phone. We assume

$$p_k(y_k|\mathbf{x}\, y_1 ... y_{k-1}) = p(y_k|x_{k-r} ... x_{k-1} x_k x_{k+1} ... x_{k+r} y_1 ... y_{k-1}). \qquad (2.4)$$

In other words, p_k is stationary and depends only on the $\pm r$ neighboring phonemes.

If we assume the kth phone does not depend any of the previous phones, we have

$$p(y_k|x_{k-r} ... x_{k-1} x_k x_{k+1} ... x_{k+r} y_1 ... y_{k-1}) = p(y_k|x_{k-r} ... x_{k-1} x_k x_{k+1} ... x_{k+r}) \qquad (2.5)$$

This is the assumption that phones are conditionally independent given the phonemic context.

To handle phone insertions, we add a second model that predicts the phone insertions. Consider a phone sequence $z_0 y_1 z_1 y_2 z_2 ... y_n z_n$ that is the realization of phoneme sequence $x_1 x_2 ... x_n$. We view phone y_i as the realization of phoneme x_i and view phone z_i as an insertion between phoneme y_i and y_{i+1}.

3. DECISION TREES

We now discuss the question of how, in general, we can estimate the likelihoods in Eq. 2.2. We stated in the introduction that we intend to estimate them directly from training data by statistical means. In the DARPA resources management task, we use the output of the phone recognizer run on the training set. Since the phone recognizer is also trained on this same data set, the phone recognition rate would be much better than on independent test sets if we did this directly. Instead, we train the phone recognizer on 9/10 of the training set and then run it on the remaining 1/10. By doing this ten times on the different portions of the training set, we are able to obtain a more realistic phone training set for lexical access.

Given this data, how can we obtain estimates the the pronunciation and duration likelihoods in Eq. 2.2?

The simplest procedure would be to collect n-gram statistics on the training data. A bi-phonemic or possibly tri-phonemic context would be the largest possible with available training data if we want statistically reliable estimates.

We believe that a straight-forward n-gram statistics on the phonemes are probably not ideal for this problem since the contextual effects that we are trying to model often depend on a whole class of phonemes in a given position, e.g., whether the preceding phoneme is a vowel or not. A procedure that had all vowels in that position clustered into one class for that case would produce a more compact description, would be more easily estimated, and would allow a wider effective context to be

examined.

Thus intuitively we would like a procedure that pools together contexts that behave similarly, but splits apart ones that differ. An attractive choice from this point of view is a statistically-generated decision tree with each branch labelled with some subset of phonemes for a particular position. The tree is generated by spliting nodes that statistical tests, based on available data, indicate improve prediction, but terminating nodes otherwise.

An excellent description of the theory and implementation of tree-based statistical models can be found in *Classification and Regression Trees* [7]. The interesting questions for generating a decision tree from data – how to decide which splits to take and when to label a node terminal and not expand it further – are discussed in these references along with the widely-adopted solutions.

Suffice it to say here the result is a binary decision tree whose branches are labelled with binary cuts on the continuous features and with binary partitions on the categorical features and whose terminal nodes are labelled with continuous predictions (*regression tree*) or categorical predictions (*classification tree*). By a continuous feature or prediction we mean a real-valued, linearly-ordered variable (e.g., the duration of a phone, or the number of phonemes in a word); by a categorical feature or prediction we mean an element of an unordered, finite set. (e.g., the phoneme set).

When categorical predictions are made, the relative probability of each outcome at a node can be directly estimated, and when continuous predictions are made, the distribution at a node can be parameterically estimated. In this way, the trees can serve as estimators of distributions like in Eq. 2.2 and not just as classifiers and predictors.

We have chosen to use decision trees to form our estimators since they (1) relatively efficiently use the available data, (2) are able to handle both categorical and continuous inputs and outputs, (3) are trainable to new corpuses quickly (which is necessary since we train on the output of a changing phone recognizer), and (4) generalize well to new test data due to the cross-validation procedure for selecting tree size [7]. The use of decision trees for these kinds of purposes has already met with some success [8-11].

4. PRONUNCIATION MODEL

In the exposition in Section 2, we combined word boundary and stress information into the phoneme set itself. When we actually input the features into the tree classification procedure we have found it more convenient to keep them separate.

We include $\pm r$ phonemes around the phoneme that is to be realized ($r = 2$). This is irrespective of word boundaries. We pad with blank symbols at sentence start and end.

Since there are about 40 different phonemes, if we directly input each phoneme into the tree classification routine, 2^{40} possible splits would have to be considered per phoneme position at each node, since, by default, all possible binary partitions are considered. This is clearly intractable, so instead we encode each phoneme as a feature vector. A manageable choice is to encode each phoneme as a four element vector: `(consonant-manner, consonant-place, vowel-manner, vowel-place)`. Each component can take one of about a dozen values and includes `'n/a'` for 'not applicable'. For example, `/s/` is encoded as `(voiceless-fricative, palatal, n/a, n/a)` and `/iy/` is encoded as `(n/a, n/a, y-diphthong, high-front)`

If the phoneme to be realized is syllabic, then we also input whether it has primary or secondary stress or is unstressed. We use stress as predicted by the Bell Labs text-to-speech system; this is essentially lexical stress with function words de–accented. If the phoneme is not syllabic, we input both the stress of the first syllabic segment to the left and to the right if present within the same word (and use `'n/a's'` if not).

To encode word boundaries, we input the number of phonemes from the beginning and end of the current word to the phoneme that is being realized.

Our output set is simply a direct encoding of the 47 element phone set used in Ljolje[1] plus the symbol `[-]` if the phoneme deletes. Computation time grows only linearly with the number of *output* classes so this direct encoding presents no problem similar to the exponential growth found with size of the input feature classes.

We now describe the results of this model applied to the DARPA resource management database. The phonetic transcription for 3838 sentences of the training set produced by our phone recognizer as described above were aligned with their phonemic transcription as predicted by the Bell Labs text-to-speech system from their orthographic transcription. For each of the resulting 140168 phonemes, the phonemic context was encoded as described. A classification tree was grown on this data and the tree size was chosen to minimize prediction error in a 5-fold cross-validation. The resulting tree had approximately 300 terminal nodes. The resulting model predicts the phone output by the recognizer 79.5% of the time (cross-validated), contains the "correct" phone in the top 5 guesses 97% of the time, and has a conditional entropy of 1.1 bits.

The corresponding insertion tree predicts whether or not the phone recognizer inserts a phone between phonemes 94.5% of the time. This seemingly good prediction is, in fact, quite poor, since the mere constant decision "doesn't insert" is correct by almost the same percentage. The best cross-validated insertion tree has only six terminal nodes, which essentially represents a fixed insertion distribution depending little on context. This reflects the fact that our choice of phone set does not produce many regular insertions (as it would if stop closure and release were separate phones), and the fact that the phone recognizer apparently does not insert spurious phones in a predictable manner.

5. DURATION MODEL

Our duration model, corresponding to the second factor in Eq. 2.2, has a very similar form to the pronunciation model. Our prediction, of course, is a continuous quantity, segmental duration, so we use a regression tree. We include all the input features

described above for the pronuncation tree, but we now add the corresponding phone too. We encode the phone with a scheme similar to that for phonemes, but add a few extra categories to fully specify all the phones. Perhaps a useful additional input feature would try to capture speech rate; we have not tried this yet.

The standard deviation in the residual in the prediction of the durations of phones output by the phone recognizer is 29 msec. This compares with an overall 45 msec. standard deviation in the phones themselves. The best cross-validated tree-length is about 300 terminal nodes.

For the lexical access, we need to represent the *probability distribution* of the durations. To do so, we can fit a gamma distribution to the data at each terminal node in the tree.

6. IMPLEMENTATION OF LEXICAL ACCESS

With these trees it is straight-forward to take a word sequence and phone sequence and estimate the likelihood that the word sequence gives rise to the phone sequence. We use the pronunciation trees to predict the first factor in Eq. 2.2 and the duration trees to predict the second factor. This simple-minded generate-and-test algorithm, of course, is not acceptable during recognition since the number of legal sentences is enormous. Instead, we have to find a more efficient way compute the exact same thing or a close approximation.

The simplest approach to an efficient implementation is to use the decision trees to form pronunciation and duration networks for each word in the vocabulary ahead of time. Then, for every possible starting phone and every possible stopping phone in the recognized phone sequence we match to the pronunciation network for each word in the vocabulary. To allow for insertions and deletions, this essentially becomes a string match with costs in terms of log likelihoods in the probabilistic model [cf. 3]. Dynamic programming permits an efficient match here [6].

This approach presents one disadvantage; word coarticulation information is mostly lost, since the individual word pronuncation model would need to be created without knowing the lexical context. To get around this, we can create multiple word models per word keyed to different lexical contexts.

7. RESULTS

At this time we have a simple version of the model described here running. We have not yet implemented the word-coarticulation component and the lexical likelihood model has the form:

$$P(\mathbf{y},\mathbf{d}|\mathbf{w}) \approx P(\mathbf{y}|\mathbf{w})P(\mathbf{d}|\mathbf{w}) \qquad (7.1).$$

In other words, the duration model does not include the phone sequence only the phoneme sequence (cf. Eq. 2.2).

Testing the model on the February '89 DARPA resource management test set and using the word-pair grammar, we achieved 85.7% word correct and 83.2% word accurary. Word insertion were 2.4% and deletions were 3.5%. This is with a phone recognizer that is achieves an estimated 81.5% phone correct and 76.0% phone accuracy on the same test set, using automatically derived phonetic transcriptions [see 1].

We are encouraged by this since it is a considerable improvement over this system's progenitor and approaching the best results reported for phone-based recognition. This improvement is due both to much better phone recognition and to improved lexical access with this approach.

We believe considerable further improvement will come when we include better duration information, word co-articulation, and, most importantly, when we input a phone lattice with recognizer alternatives rather than just the best guess. We have, in fact, implemented a crude version of a lattice in which the segmentation produced by the best guess is used, but alternative phones and their likelihoods are included. This performed 88.5% phone correct and 87.2% phone accuracy on the the Feb '89 test set. We are now implementing a structure that allows a true lattice that will allow alternative segmentations.

9. REFERENCES

[1] Ljolje, A. 1990. Phone classification using high order phonatactic constraints: preliminary results. *NATO ASI Speech Recognition and Understanding.* July 1990, Cetraro, Italy.

[2] Zue,V., Glass, J., Phillips, M. and Seneff, S. 1989. The MIT Summit Speech Recognition System: a progress report. *Proc. DARPA Speech and Natural Language Workshop.* Feb 1989, pp. 179-189.

[3] Levinson, S.E., Ljolje, A. and Miller, L. 1990. Continuous speech recognition from a phonetic transcription. *Proc ICASSP '90.* pp 93-96.

[4] Coker, C.. 1985. A dictionary-intensive letter-to-sound program. *J. Acoust. Soc. Am.* **78**, Suppl. 1, S7.

[5] Fisher, W., Zue, V., Bernstein, D. and Pallet, D. 1987. An acoustic-phonetic data base. *J. Acoust. Soc. Am.* **81**. Suppl. 1.

[6] Kruskal, J. 1983. An overview of sequence comparison. In *Time Warps, String Edits, and Macromolecules: the Theory and Practice of Sequence Comparison.* D. Sankoff and J. Kruskal, eds. Reading, MA: Addison Wesley. pp. 1-44.

[7] Brieman, L., et. al. 1984. *Classification and Regression Trees.* Monterey, CA: Wadsworth & Brooks.

[8] Chen, F. 1990. Identification of contextual factors for pronunciation networks. *Proc. ICASSP '90.* S14.9.

[9] Riley, M. 1989. Statistical tree-based modeling of phonetic segment durations. *J. Acoust. Soc. Am.* **85**. S44.

[10] Randolph, M. 1990. A data-driven method for discovering and predicting allophonic variation. *Proc. ICASSP '90.* S14.10.

[11] Riley, M. 1989. Some applications of tree-based modelling to speech and language. *Proc. DARPA Speech and Natural Language Workshop.* Oct 1989, Cape Cod, MA, pp. 339-352.