

ON THE ROAD TO IMPROVED LEXICAL CONFUSABILITY METRICS

Eric Fosler-Lussier, Ingunn Amdal*, and Hong-Kwang Jeff Kuo

Bell Labs, Lucent Technologies
600 Mountain Ave.
Murray Hill, NJ 07974 USA
fosler,kuo@research.bell-labs.com

ABSTRACT

Pronunciation modeling in automatic speech recognition systems has had mixed results in the past; one likely reason for poor performance is the increased confusability in the lexicon from adding new pronunciation variants. In this work, we propose a new framework for determining lexically confusable words based on inverted finite state transducers (FSTs); we also present experiments designed to test some of the implementation details of this framework. The method is evaluated by looking at how well the algorithm predicts the errors in an ASR system. We see from the confusions learned in a training set that we are able to generalize this information to predict errors in an unseen test set.

1. INTRODUCTION

In a speech recognition system, the typical pronunciation model consists of one “baseform” pronunciation and possibly a set of alternative pronunciations for that word. Often, a pronunciation modeler must choose the number of pronunciations per word based on speed and accuracy issues: increasing the number of pronunciations within a system often increases decoding time; a large number of pronunciations can also cause a decrease in accuracy. Both of these problems are related to the concept of *confusability*: words with similar phonetic sequences can be confused with each other, unless the language model asserts its influence to disambiguate them.

Despite the inclusion of the acoustic model [1, 2] and language model influences [3] into the pronunciation modeling process, most models today lack a sense of how added alternative pronunciations will affect the overall decoding process. For example, allowing word-final deletion of the phone [t] can make the phrases *can’t elope*, *can elope* and *cantaloupe* homophonous. A method of quantifying the confusion inherent in a combined acoustic-lexical system is needed.

In preliminary work [4], we defined a confusion metric that gives bounds on the confusability in the lexicon. The metric is premised on the following idea: what if our acoustic models could produce a phonetic string that perfectly matches the pronunciations in the dictionary? We could then compute the set of word pronunciations that match any

phonetic substring in the data, producing a lattice of possible matching words (Figure 1) and counting how many words appear in correspondence with each phone. This metric, the “all confusion” metric, overestimates the number of possible confusions, since it doesn’t take into account that some words would be pruned during decoding because of a dead-end path in the word lattice: for example, the word *the* in the figure doesn’t have any appropriate following word in the lattice. An “exact confusion” metric ameliorates this somewhat by only counting confusions that occur at the word boundaries provided by the forced alignment – an underestimate of the amount of confusion in the lexicon. In our experiments [4], we found that this metric did not correlate well with the word error rate or the speed of ASR decoding; however, it was useful in selecting non-confusable pronunciation variants, providing an 8% reduction in word error rate.

There are some problems inherent in this metric. First, it only takes the Viterbi path into account in creating the confusion lattice. As discussed above, the overlap between acoustic models means that introducing a variant can create confusion with another model, even if the two phone strings do not completely match. A second problem is that unlikely paths in the confusion lattice are given as much weight as likely paths. Incorporating language model information into the lattice would provide a more accurate reflection of the decoding process, and hence a more accurate picture of the possible lexical confusions.

2. A PREDICTIVE WEIGHTED FST MODEL

Our proposal is to integrate possible acoustic model confusions, pronunciation modeling, and language model information into a single framework for describing the confusability of lexicons. Recent work [5, 6] has shown that the recognition process can be modeled with a sequence

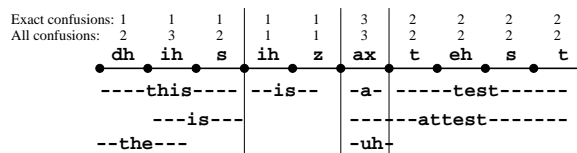


Fig. 1. Example of part of the lattice used to compute the preliminary confusion metric, from [4]

*Ingunn Amdal is currently at Telenor R&D, Norway, e-mail: ingunn.amdal@telenor.com. This work was done while visiting Bell Labs.

of weighted finite-state transducers (WFSTs).¹ An abstract representation of the Viterbi decoding process might be given as:

$$\hat{W} = \text{best path}(A \circ T \circ P \circ L) \quad (1)$$

where \hat{W} is the sequence of words corresponding to the best recognition hypothesis, A is a finite state automaton (FSA) containing the set of acoustic scores computed from an input utterance, T is a context-dependence FST, containing a mapping from acoustic states to triphones, P is the pronunciation model FST, containing a mapping from triphones to words, L is the language model FSA, which contains n-gram statistics, and \circ is the composition operator. All of these finite state machines are typically weighted, with the costs derived from the probabilities from the particular linguistic model.

A nice feature of finite state transducers is that they are invertible; instead of viewing the model M as “A produces B”, the system can be turned around (M^{-1}) to say that “B produces A.” Thus, one could determine the weighted set of all word sequences \mathbf{W} confusable with any word sequence W by composing the given word sequence with inverted transducers until acoustic scores are produced, and then reversing the process:²

$$\mathbf{W} = W \circ L \circ P^{-1} \circ T^{-1} \circ T \circ P \circ L \quad (2)$$

A consequence of this equation is that W is guaranteed to be a member of \mathbf{W} . Because the inverted mappings are one-to-many (especially T^{-1}), and because the word boundary information is lost with the composition of P^{-1} , the set \mathbf{W} will typically have many more members than W .

A drawback with these models is that they are expensive to compute for large-vocabulary systems; special methods, such as on-the-fly transduction [5] are required for efficient decoding. Furthermore, it is not straightforward to see how a non-transducer-based decoder can be exactly modeled by such a system.

Our approach is to assume that we can encapsulate the overlap in acoustic models using some form of summary information; in particular, we assume that the acoustic errors made by a recognizer can be captured by a confusion matrix between phones derived from recognition errors. We therefore use a confusion matrix represented by a FST C to model $T^{-1} \circ T$, thereby also avoiding the problems of non-transducer-based decoders. The confusion C will map each canonical phone, as given by a dictionary, to the phones that can be confused with that canonical phone. In this case, Equation 2 can be simplified (also eliminating the initial deterministic scaling by the n-gram grammar L):

$$\mathbf{W} = \hat{W} \circ P^{-1} \circ C \circ P \circ L \quad (3)$$

¹In this paper, all finite state machines are weighted; for notational convenience, however, we leave off the “weighted” designation.

² L is not inverted because it is a finite state automaton, and thus $L = L^{-1}$; moreover, the initial composition with L is not strictly necessary if L is a n-gram grammar, since it just scales the score for W deterministically.

Besides the inclusion of acoustic confusability and language model scoring in the process, this model has another advantage over the previous confusability model [4]: rather than over-counting or under-counting the possible confusions, here the model is constrained by the lexicon and language model. This means that only words that are part of complete paths in the decoding will be counted as confusions. Inclusion of the language model will also help weight paths appropriately in continuous speech recognition.

An example of the added ability of the new model compared to the old model, is that if the correct word in Figure 1 was in fact “attest”, the exact-confusion metric would not have counted “a test,” since the word boundaries of “a” and “test” do not correspond to word boundaries in the correct hypothesis. The all-confusion metric would have included these words, but would have also included the letter “S” and “Tess.” The output of the confusion matrix transformation ($\hat{W} \circ P^{-1} \circ C$) is a phone graph; when this phone graph is transduced with the pronunciation dictionary P , all paths in the phone graph that do not correspond to a complete word hypothesis are eliminated. If, for example, the deletion of final [t] were not present in the confusion matrix C , then the word hypothesis “a Tess” would be impossible, since there is no valid word pronunciation corresponding to the remaining [t]. The restriction of only allowing confusable words that are part of a complete path gives a more accurate approximation to the actual decoding process.

3. EXPERIMENTAL CORPUS

In order to concentrate on developing appropriate confusion matrices without the effect of language models, we chose to conduct our initial experiments with the Phonebook isolated word recognition task.

Typically, this corpus has been used to do vocabulary-independent acoustic model testing; recognition experiments are usually a forced one-of- n choice (where n ranges from 75 to 600). In this case, to maximize the confusability, we included the entire vocabulary of 7979 words in the recognizer (giving a rather difficult effective LM perplexity of 7979). The baseline lexicon was generated by the Bell-Labs TTS system with only one pronunciation per word.

We performed ASR transcription of the entire Phonebook corpus using the triphone acoustic models from our DARPA Communicator recognizer, which was trained on several general American English corpora. The resulting transcription had a word error rate (WER) of 20%.

We divided this recognized data into speaker-disjoint training and test sets (Table 1). The number of overlapping words in two sets was relatively small – only 819 (19%) different words (in terms of number of utterances, that is 5967 of 45739, 13%). Note that by training set, we mean the set we used to train the phone confusion matrices.

4. METHOD

In this section, we provide a simplified example of the confusion matrix training process for the Phonebook corpus. To start the training, we compute an alignment be-

Set	No. spkrs	No. utts	Vocab size
All	1358	93667	7979
Test set	660	45739	4300
Training set	698	47928	4498

Table 1. Training and test set of Phonebook

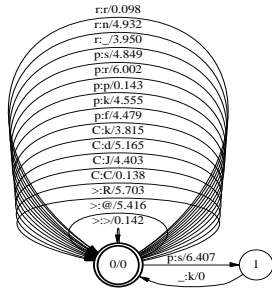


Fig. 2. Phone confusion matrix C_{mini}

tween the canonical and the recognized (“actual”) transcription for each utterance. For example, if “porch” were misrecognized as “forge”, the resulting alignment would be [p:f >: > r:r C:J]. The alignment procedure is a fully automatic dynamic programming technique with substitution costs based on a phonetic distance metric. Costs for deletions and insertions are adjusted to give reasonable alignments. Some alignments are more challenging than others, one example is the correct word “egghead” aligned with the recognized word “uneducated” [e:Λne g:Jʊk h:At e:i d:d]. Insertions are represented by a phone-to-phones mapping, e.g. g:Jʊk. The total number of mappings in the confusion matrix may therefore be higher than the number of phones squared.

The cost of each phone-to-phon(e)s mapping between canonical and realized transcriptions ($\text{cost}(w_r|w_c)$) is set to the negative log likelihood of observing the recognized phone(s) w_r given the correct phone w_c ; this is estimated by counting occurrences of these pairs in the training set.

$$\begin{aligned} \text{cost}(w_r|w_c) &= -\log \left[\hat{P}(w_r|w_c) \right] \\ &= -\log \left[\frac{\text{count}(w_c : w_r)}{\text{count}(w_c)} \right] \end{aligned} \quad (4)$$

A small example confusion matrix C_{mini} is shown in Figure 2. The arc labels show the mapping and the cost: “ $w_c : w_r / \text{cost}(w_r|w_c)$ ”. We can see that there is one insertion here, p : sk, and one deletion, r : _.

The probabilities of insertions are computed as a multiple-unit substitution (e.g., $\hat{P}(\text{sk}|p)$), but for inclusion into the FSM framework, the mapping is broken into several individual maps (e.g., p : s, _ : k), with the first pair carrying the cost.

The next step is to find the pronunciations that are confusable with each correct word. We compose the phones of each word with the confusion matrix; the result of the

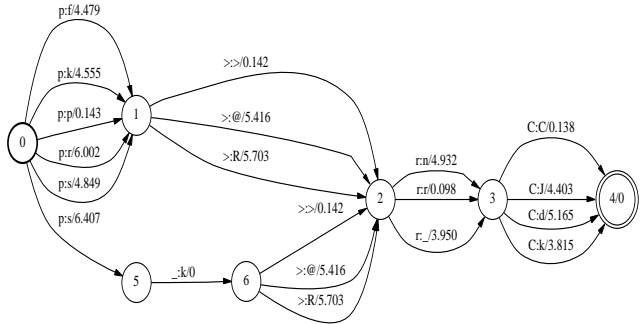


Fig. 3. Graph of confusable phones

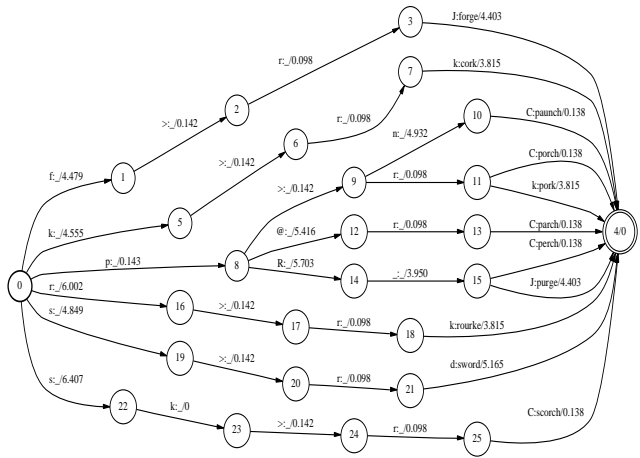


Fig. 4. Graph of confusable phones that form vocabulary words

composition of the phones in “porch” and confusion matrix, “porch” $\circ P^{-1} \circ C_{\text{mini}}$, is shown in Figure 3.

Many of the paths in this graph will not form complete vocabulary words; to eliminate invalid paths, we compose this result with the lexicon, “porch” $\circ P^{-1} \circ C_{\text{mini}} \circ P$, and obtain the resulting confusable words shown in Figure 4.

To assess how well we predict errors, we have looked at the rank of the recognized word in the confusion graph for the correct word; the score for a word is computed by summing the negated phone costs over the path, and therefore represents the log likelihood of the word confusion. Table 2 shows the score for the confusable words from Figure 4. As we can observe the confusion matrix predicts that “forge” will be confusable with “porch,” along with several other words.

The two last columns show the scores from two different confusion matrices, the first (all) trained on all recognized utterances, and the second (error) trained on errors only. As we can see, the correct word gets a closer score to the competitors in the case of the “error”-confusion matrix. This is because identity-mappings get a much lower weight.³ The

³There will be identity mappings in the error alignments also. We use

Rank	Word	Phone-to-phones(s) mappings	Score C_{all}	Score C_{error}
1	porch	p:p >: > r:r C:C	-0.52	-5.42 (1)
2	pork	p:p >: > r:r C:k	-4.20	-5.46 (2)
3	paunch	p:p >: > r:n C:C	-5.36	-7.60 (5)
4	parch	p:p >:@ r:r C:C	-5.80	-8.08 (6)
5	scorch	p:sk >: > r:r C:C	-6.79	-8.59 (8)
6	cork	p:k >: > r:r C:k	-8.61	-6.77 (3)
7	forge	p:f >: > r:r C:J	-9.12	-7.28 (4)
8	perch	p:p >:R r:_ C:C	-9.94	-9.56 (9)
9	rourke	p:r >: > r:r C:k	-10.06	-8.22 (7)

Table 2. Rank and score of words confusable with “porch”

rank of the competitors will also be different, in this case our recognized word “forge” (which is in the training set) would get a better rank using the errors only in training. This is generally not the case.

5. EXPERIMENTS IN FORMING THE CONFUSION MATRIX

There are many different factors that can go into the development of a confusion matrix; in this section, we outline four experiments that are investigated in this work. The first three (choice of training set, choice of model type, and choice of transcriptions) are fundamental questions in how to build the training set; the fourth experiment starts to refine these models.

We have used the derived confusion matrices to predict the errors made by the ASR system. For each error utterance, we have built the phone confusion graph and sorted the vocabulary words it contains by rank as shown in Table 2.

One way to judge the quality of prediction from a confusion matrix is to look at how often a misrecognized word has a rank below a given threshold; we represent this as a graph showing the cumulative percentage of error words falling below a particular rank threshold (Figures 5–7). If the x-axis of the graph were expanded to 7979, all curves would meet at 1. The ideal result for this rank-assessment would be that the correct word gets rank 1 and the recognized word (if different from the correct word) gets rank 2. Analyzing the errors more closely we see that this is not possible, the same word may introduce several errors and the recognized words in these errors are generally not the same. For all the experiments shown the correct word will have rank 1 if not stated otherwise. The figures of cumulative rank is therefore only shown for the words where the recognized word is different from the correct word; the errorful words.

First we show the results on the training set in order to see how well the confusion matrix is modeling the seen confusions. In Figure 5 we show the cumulative distribution of ranks of the errorful words tested on the training set. We see that using all utterances in the confusion matrix training (solid line) the rank of the recognized word is 1000 or better in 81.4% of the cases.

The results for the test set is shown in Figure 6. The prediction of the errors is worse than on the training set as

all phones in the alignment, not only the erroneous ones.

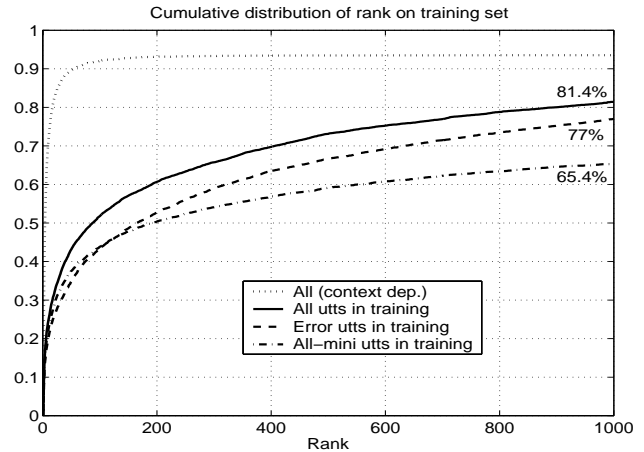


Fig. 5. Predicted rank of recognized errors in training set

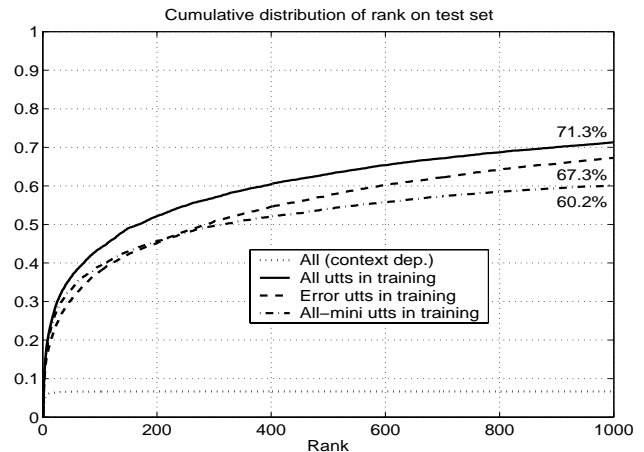


Fig. 6. Predicted rank of recognized errors in test set

expected, with a difference of about 5–10%. Still, the results show that the training set results generalize pretty well to the unseen test set: 71.3% of the errorful words are predicted within rank 1000. Discussion of the results using context-dependent mappings (dotted lines) and all-mini utterances (dot-dashed lines) will be given later.

When looking closer at the predicted errors, we observe that when the correct and recognized words are similar they have a better rank as expected. The “porch/forge” pair in Table 2 is an example of this. When we get a rank outside 1000, the correct and recognized words are typically quite different, as for example “duty/julianne,” “gobblers/novelist,” and “handmaid/envisioning.” These pairs are very hard to predict in general.

We have investigated a threshold on the minimum number of occurrences of each mapping when building the confusion matrix in Figure 7. The performance on the training set is as expected because learning all mappings on the same set as testing on should be beneficial. The result on the test set is also best when using all mappings. For mappings occurring less frequently the cost will be higher and these mappings will have less influence. We therefore counted all

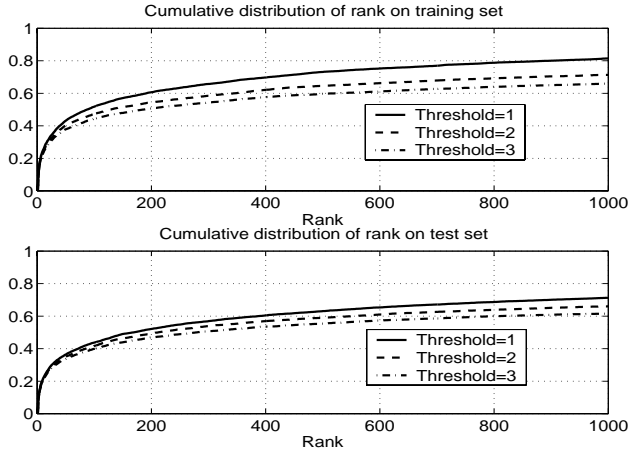


Fig. 7. Predicted rank of recognized errors using a threshold on the minimum number of occurrences for training.

mappings when building the confusion matrices.

5.1. Training set material: all words or errorful words?

As discussed in Section 4, we can choose to select only errorful transcriptions to train the confusion matrix, or include all of the data. We built two confusion matrices on the training set, either using only errorful utterances (error) or all of the utterances (all). The cumulative distribution graph for the errorful words in the test set can be seen in Figure 6.

At first glance, it appears to be better to train the matrix on all of the data (solid line), rather than focusing on errorful data (dashed line), even though the test set contains only errorful words! The main difference between the two matrices is the scaling of the cost of identity mappings compared to the other mappings as shown in Table 2. Since the error confusion matrix is trained on less data, we derived a third matrix, all-mini, trained on a similar-sized random sample (20%) of the training data (dot-dashed line). This performed worse than the error matrix, this suggests that focusing on the errors through reweighting of the mappings may prove fruitful in further experiments.

5.2. Model type: context independent or context dependent?

The confusion matrix may be represented as a set of context-independent mappings, or we may choose to include context; for example, we may predict that a particular triphone will be confused with some monophone other than the corresponding monophone of the triphone. Context has been shown to be very important in predicting pronunciation variation [7], but inclusion of context gives many more parameters to estimate and we may encounter over-training.

We have tried a simple context-dependent mapping scheme using only the identity of preceding and succeeding phone (no clustering). As expected, the context-dependent confusion matrix gives much improved performance on the training set, see dotted line in Figure 5, however, when testing, we observed that the context-dependent mappings are

not able to generalize well (roughly 8% of the misrecognitions found within rank 1000, dotted line in Figure 6). We also tested a “combination” where both the context-dependent and context-independent paths are maintained (and a path may consist of both context-dependent and independent arcs). This gave the same performance as the context-independent matrix on the test set.

This experiment suggests that naive context-dependent confusion matrices suffer from a lack of training data. However, we believe that using decision-tree clustering of the context elements can improve the performance of the confusion matrix.

5.3. Transcriptions: recognized words or phone recognition?

The above example uses the phonetic alignment of the recognized word as the “alternative” phone transcription, which is aligned to the canonical transcription. This allows the lexicon to play a part in determining the confusion pairs, but may not produce a matrix that generalizes well to unseen data. In the other extreme, a phone-loop recognition with no grammar would produce phonetic strings unbiased by the lexicon, but these strings may not form valid candidates according to the lexicon. In between these two extremes, one could also do phone recognition with a bigram or trigram grammar trained on the lexicon, which would have the effect of reintroducing some of these constraints.

When using phone-loop transcriptions, we got much larger confusion matrices. This is expected since many more mappings will be available than when only vocabulary word transcriptions are used. The performance was not as good as with starting from word recognition transcriptions, but not way off: 68% of the recognized words was within rank 1000 (*cf.* 71% with word recognition transcriptions). More sophisticated pruning and context-dependent mapping may be needed in this case. Using a phone bigram or phone trigram grammar made no big difference.

Since the common wisdom has been that monophone models show pronunciation variation better in automatic transcriptions, in later experiments we also started from transcriptions from monophone acoustic models trained on the same acoustic training set (30% WER vs. 20% WER for triphone acoustic models). However, for the phone transcriptions derived from monophone acoustic models, we found lower performance: 60% of errors were predicted within rank 1000 (*cf.* 68% for phone transcriptions from the triphone model).

5.4. Discriminative training of matrices

In Section 5.1, we saw that focusing the matrix built from all data more towards errors may prove beneficial; one method of achieving this is through discriminative minimum classification error training. In [8], we presented a discriminative technique for language model optimization. We have reformulated this technique to train our confusion matrix.

We want the recognized word to advance in the ranked list from the confusion graph for the correct word. This can be achieved by defining an empirical loss function

dependent on the confusion matrix weights for the mappings. First we define a word-confusion log probability $g(W|W_c, C)$ where W_c is the correct word and C the confusion matrix. If W_r is the recognized word, the set $\mathcal{W} = \{W_1, W_2 \dots W_N\}$ contains the competitors, in this case the words from the confusion rank list that is not W_t . A possible misclassification measure is then:

$$d(W) = -g(W_r|W_c, C) + \log \left[\frac{1}{N} \sum_{n=1}^N e^{g(W_n|W_c, C)\eta} \right]^{1/\eta} \quad (5)$$

For each word in the training set we build a confusion graph giving us the phone-to-phone(s) mappings from the correct word to the predicted word. The word-confusion log probability can be expressed by summing over the phone pairs $w_c : w_n$ in the alignment. The sum of scores for the mappings can be reformulated to summing over the score for each unique mapping c_{ij} multiplied by the number of occurrences of this mapping N_{ij}^n (for word n).

$$\begin{aligned} g(W_n|W_c, C) &= \sum_{\text{alignment length}(n,c)} \log P(w_n|w_c, C) \\ &= \sum_{\# \text{ diff. mappings}(i,j)} c_{ij} N_{ij}^n \end{aligned} \quad (6)$$

The empirical loss can be found by feeding the misclassification measure through a sigmoid loss function. By differentiating the empirical loss with respect to the confusion matrix weights we get the formula for updating the weights of a mapping dependent on the number of mappings N_{ij}^n . The resulting formula enhances the phone-to-phones mapping of the recognized word and punishes the phone-to-phones mappings of competitors.

There are a number of issues to be considered, first of all it will be computational infeasible to use all competitors, the N must be reduced to some reasonable value. What do we do if the recognized word is outside this window? The competing words will also generally have multiple possible phone-pair alignments, should all or only the best be used? There are also a number of parameters to be adjusted. We believe that discriminative training is a promising method to improve the confusion matrices and that more research needs to be done.

6. CONCLUSIONS AND FUTURE WORK

We have shown that the confusion matrix formalism is able to reasonably predict recognition errors in the test set at least better than chance. This means there is a pattern in the confusions that a simple phone-to-phone mapping can capture. The phone confusion matrix can likely be used to predict which words are most confusable given the correct word.

We would also like to say something about how likely the confusions are. The confusion matrix gives us a score for each confusable word; using this score we could derive

a metric for comparing the probability of confusion for different words and also different lexica.

This formalism lends itself to extensions easily; for example, in future work, we would like to incorporate errorful hypotheses from n-best recognition, utilize acoustic scores in the confusion matrix, and integrate decision-tree modeling for contextual variation. The current model, of course, is very simple (by design): it has no concept of durations or long-term context. However, it provides a good baseline for future development of error-predictive technology, as well as the elusive lexical confusion metric. It is not currently clear what form this metric will take, but this work suggests several possibilities, including finding how many confusable words fall into a search beamwidth of the top word, or calculating an entropy-like distribution over the top n words. Future experiments will look for correlations between these types of metrics and word error rate.

Extending this work to deal with continuous word recognition is theoretically very simple: all that is required is the addition of the language model through transducer composition. In practice, this will probably be rather computationally expensive because the phone and word graphs will become large, and beamwidth search techniques will be required to efficiently determine n-best hypotheses. However, the resulting solution should be similar to regular ASR decoding, save the need for evaluating acoustic models.

7. REFERENCES

- [1] T. Holter and T. Svendsen, "Maximum likelihood modelling of pronunciation variation," *Speech Communication*, vol. 29, pp. 177–191, 1999.
- [2] E. Fosler-Lussier and G. Williams, "Not just what, but also when: Guided automatic pronunciation modeling for broadcast news," in *DARPA Broadcast News Workshop*, (Herndon, Virginia), March 1999.
- [3] J. M. Kessens, M. Wester, and H. Strik, "Improving the performance of a Dutch CSR by modeling within-word and cross-word pronunciation variation," *Speech Communication*, vol. 29, no. 2–4, pp. 193–207, 1999.
- [4] M. Wester and E. Fosler-Lussier, "A comparison of data-derived and knowledge-based modeling of pronunciation variation," in *ICSLP*, (Beijing, China), 2000.
- [5] M. Mohri, M. Riley, D. Hindle, A. Ljolje, and F. Pereira, "Full expansion of context-dependent networks in large vocabulary speech recognition," in *ICASSP*, (Seattle, Washington), 1998.
- [6] X. Mou, S. Seneff, and V. Zue, "Context-dependent probabilistic hierarchical sub-lexical modelling using finite state transducers," in *Eurospeech*, (Aalborg, Denmark), 2001.
- [7] M. Riley, "A statistical model for generating pronunciation networks," in *ICASSP*, pp. 737–740, 1991.
- [8] H.-K. Kuo, E. Fosler-Lussier, H. Jiang, and C.-H. Lee, "Discriminative training of language models for speech recognition," in *ICASSP*, (Orlando, Florida), May 2002.