

PREDICTION OF WORD CONFUSABILITIES FOR SPEECH RECOGNITION

David B. Roe and Michael D. Riley

AT&T Bell Laboratories
Murray Hill, NJ

ABSTRACT

Words which are similar in pronunciation cause errors by speech recognizers. In an application of speech recognition, the vocabulary should be chosen so as to avoid similar sounding words or phrases. Phonetically similar words (such as "wait" and "eight") or short words ("on" and "off") may be confused by the speech recognizer, with undesirable consequences.

We have developed a software tool, *word_confuse*, that detects confusable words. The confusability between pairs of words is calculated from two sources of information: the phonetic pronunciation of words as determined by the AT&T text-to-speech synthesizer, and the phonetic confusions exhibited by an AT&T phone-based speech recognizer. The calculation of confusability is based on searches through a finite state network that represents probabilistically the phonetic pronunciation of words. The metric of similarity is based on the Bhattacharyya distance. *Word_confuse* can be used to detect and eliminate confusable words from the vocabulary used in speech recognition applications.

1. INTRODUCTION

Confusable words are the bane of speech recognizers. It is well known that speech recognition accuracy on confusable vocabularies (such as the letters of the alphabet, including the subset B, C, D, E, G, P, T, V, Z) is much lower than on longer, easily-distinguishable words. In some applications of speech recognition, it is possible to select the set of vocabulary words to avoid words with similar pronunciation. Not all speaker-independent applications lend themselves to choosing a dissimilar set of vocabulary words - for instance, voice dictation and spoken language translation. But many telephone applications and computer control applications have restricted vocabulary sets from which confusable words can be eliminated.

Ideally an application designer would choose a vocabulary that is both easy for people to remember and that contained no pair of words that might be confused by a speech recognizer. Consider a hypothetical application involving the English digits and commands such as the word "wait". Since the command "wait" is similar to "eight", it may be prudent to substitute a different command with similar meaning, such as "stop" or "pause". In principle, the number of speech recognition errors should be reduced if all words in the vocabulary are chosen to be acoustically distinct.

There has been little previous work concerning automatic determination of word confusability for speech recognizers. The traditional approach has been to choose a vocabulary list based on experience, test the accuracy, and then correct the vocabulary list if there are too many recognition errors. At IBM there has been research on finding families of words which are similar to a target word [1], but their technique requires a complete set of trained HMM's for all words, and speech data from several speakers for the target word. For speaker-trained speech recognizers, a word that is

close to the pronunciation of a previously trained word can be detected, also based on acoustic comparison [2]. But it is desirable to be able to predict confusable words for speaker-independent applications *in advance* of collecting speech samples. Rabiner and Juang point the way to a phonetic approach in [3].

The basic idea behind predicting word confusability is simple. Text-to-speech systems can determine the phonetic pronunciation of words from text. Words that have similar phonetic pronunciations are likely to be confused by speech recognizers. Moreover, certain phones of English (such as *s* and *z*) are more likely to be mis-identified by speech recognizers than other pairs. Phonetic transcriptions that differ only in easily-confused phones are more likely to cause mis-recognitions than those that contain dissimilar phones. The phone-by-phone confusability can be measured experimentally by tabulating the errors made by phone-based speech recognizers.

This memorandum describes the theory of word confusability as determined by word pronunciations (Section 2), the implementation of a program named *word_confuse* which is based upon finite-state pronunciation networks to predict potentially confusable words (Section 3), and a comparison of the predictions of *word_confuse* to experimentally determined speech recognition errors for three different vocabulary sets (Section 4).

2. THEORY

Though there are several approaches for determining the acoustic similarity between words, we choose an approach based on the phonetic pronunciation and a measure of confusability of the phonetic units rather than acoustic examples [1,2] of the words themselves. Given two potentially similar words, we begin with their phonetic pronunciations from a text to speech synthesizer. Then we estimate the probability that the phonetic pronunciation of the first word will be misrecognized as the second word, rather than the first. This approach allows an estimate of confusability *before* recording speech utterances to find the actual pronunciations of the desired vocabulary. This approach also has the benefit of simplicity of calculation compared to estimates of similarity at the acoustic level. However, there is a potential drawback that actual pronunciations may not be represented accurately by the phonetic pronunciations from a dictionary, especially when coarticulation is important.

We wish to estimate the probability of recognizing word W_1 given an utterance of word W_2 . This conditional probability, denoted $P(\text{rec}(W_1) | \text{utt}(W_2))$, is a measure of the confusability between W_1 and W_2 . To estimate this probability, information is available about the phonetic transcriptions of W_1 and W_2 , T_1 and T_2 . A transcription is a sequence of phones: $T = (p_1, p_2, \dots, p_n)$, where p_i are phonetic labels. Information is also available about phonetic errors that a speech recognizer is likely to make. The probability that a phone p_i will be recognized, given an utterance of p_j , is denoted $P(\text{rec}(p_i) | \text{utt}(p_j))$, and can be measured experimentally with speech recognizers. More generally, we may be able to use additional

conditioning information other than just $utt(p_j)$, such as the previous or following phone uttered, with sufficient training data and a suitable estimator. Let us associate the confusability of words with the probability of error P^E , defined in a symmetric way:

$$P^E = \frac{1}{2}(P(rec(W_1)|utt(W_2)) + P(rec(W_2)|utt(W_1)))$$

We estimate the confusability between W_1 and W_2 with a two-stage process, in which a phonetic transcription is intermediate between the spoken word and the recognized word. In this approach, phonetic transcription is viewed as a stochastic function of the uttered word. Instead, we could have stipulated that each word have only a single legal transcription (as in the following paragraph), but our approach allows us to model quantitatively that different pronunciations of a word should give rise to different transcriptions and that similar sounding words may result in the same transcription. Thus, confusability is estimated by summing appropriately summing over all possible phonetic transcriptions $T=(p_1, p_2, \dots, p_n)$. For each possible phonetic transcription T , we can estimate the probability $P(T|utt(W_2))$ that a spoken utterance $utt(W_2)$ will be transcribed by a phone recognizer as T . This approach has been used at BBN [4] for a different objective. We can further estimate the probability that the speech recognizer will interpret the transcription T as word W_1 , as opposed to some other word. Defining $P(rec(W_1)|T)$ to be the probability of recognizing W_1 given a transcription T , and combining these two terms over all T , the probability of confusion can be computed as:

$$P(rec(W_1)|utt(W_2)) = \sum_{\text{all } T} P(rec(W_1)|T) \times P(T|utt(W_2)) \quad (1)$$

The assumption is made that the only information available is contained in the transcription, that is, that

$$P(rec(W_1) | T \& utt(W_2)) = P(rec(W_1) | T).$$

(There is an alternate, but inferior, way to estimate confusability between W_1 and W_2 in a single-step process. It is possible to compute the probability that the phones T_2 , corresponding to W_2 , are transcribed as T_1 , corresponding to W_1 , by the phone recognizer. This probability, $P(T_1|utt(W_2))$, is high when T_1 is "close" to T_2 . However there are both theoretical and practical objections to this simple approach. First, this estimate of probability is flawed because it considers only one of the many transcriptions that might be produced by the phone recognizer. On a practical level, it is difficult to estimate the probability accurately from a finite-sized data set when the phones in the phonetic transcriptions T_1 and T_2 are dissimilar.)

The second term of Equation 1, $P(T|utt(W_2))$, is estimated by producing the transcription T_2 of the "correct" pronunciation, then estimating the probability that T_2 will be transcribed as T by a speech recognizer. The pronunciation component of the AT&T text-to-speech synthesizer [5] handles phrases as well as isolated words. In some cases, alternate pronunciations may be generated to account for words pronounced differently in different contexts. For instance, "the" is pronounced either "dh ax" or "dh iy". From the correct pronunciation T_2 , we estimate the probability that another transcription T will result during speech recognition, knowing the errors characteristic of speech recognizers. Associated with each phone in this transcription T_2 there are characteristic substitutions, insertions, and deletions that a speech recognizer is likely to make. The probability of these characteristic errors depends on the specific error. For instance, a speech recognizer is more likely to substitute the phone n for m , than to delete the phone er . The probability of phone substitutions, deletions, and insertions has been measured by Ljolje and Riley [6] for one particular phone-based recognizer. The technique is to use HMM-based phone recognition combined with a decision tree learning algorithm to estimate the phone confusions in their phonetic contexts [7].

The first term of the right side of Equation 1 can be estimated by assuming a minimum error decision criterion. Given a transcription T , the decision criterion that minimizes P^E for two words W_1 and W_2 is:

$$P(rec(W_1)|T) = \begin{cases} 1 & \text{if } P(T|utt(W_1)) > P(T|utt(W_2)) \\ 0 & \text{otherwise} \end{cases}$$

Let S_1 be the set of transcriptions for W_1 that are "close" to T_1 :

$$S_1 \equiv \{ T \mid P(T|utt(W_1)) > P(T|utt(W_2)) \}$$

(The situation when more than two words are in the vocabulary can be handled similarly.) Then,

$$P(rec(W_1)|utt(W_2)) = \sum_{T \in S_1} P(T|utt(W_2))$$

and

$$P^E_{\min} = \frac{1}{2} \sum_{\text{all } T} \text{Min}(P(T|utt(W_1)), P(T|utt(W_2))) \quad (2)$$

Unfortunately, Equation 2 is impractical because it is too time consuming to calculate. A convenient approximation to the right hand side of Equation 2 is ρ , the Bhattacharyya coefficient:

$$\rho_{1,2} \equiv \sum_{\text{all } T} \sqrt{P(T|utt(W_2)) \times P(T|utt(W_1))} \quad (3)$$

Chen [8] gives the following bounds on the validity of the approximation of Equation 2 by Equation 3:

$$(\rho/2)^2 \leq P^E_{\min} \leq \rho/2$$

Note that the definition of ρ is symmetric so that

$$P(rec(W_1)|utt(W_2)) \approx \rho_{1,2} = \rho_{2,1} \approx P(rec(W_2)|utt(W_1))$$

Equation 3 has the further advantage that the confusability of a word with itself is always normalized to unity, since

$$\rho_{i,i} \approx \sum_{\text{all } T} P(T|utt(W_i)) = 1$$

Although the Bhattacharyya coefficient ρ is only an approximation, we believe its computational advantages outweigh the lack of precision, especially in view of the more serious uncertainties in estimating $P(T|utt(W_i))$ (see section 3.1).

3. IMPLEMENTATION WITH FINITE STATE NETWORKS

To calculate the conditional probability of a transcription T given a word W_i , a finite state network with costs on the arcs is convenient. From the correct transcription T_i and the estimator of phone confusions, we construct a statistical finite state network of probable phonetic transcriptions. Each arc in the network represents a phone in a transcription, together with a probability associated with that phone arc. A description of these finite state networks and a set of programs to manipulate them may be found in Reference [9]. By way of example, a finite state network for the word "the" is shown in Figure 1. Arcs with probabilities less than 0.02 are not shown in this example, though they may be important in contributing to alternate pronunciations.

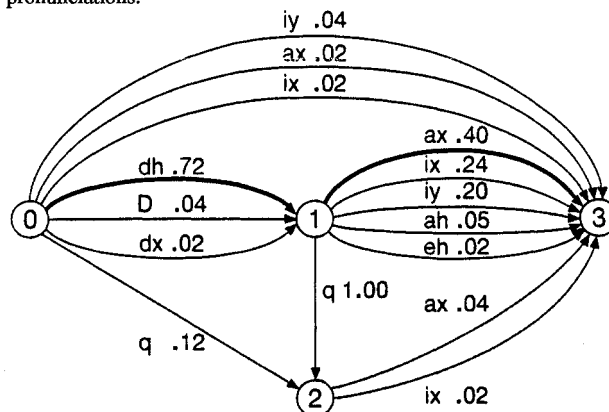


Figure 1. Finite state network for the pronunciation of "the". The common pronunciation "dh ax" is shown in bold.

The majority of the pairs are highly dissimilar; the median distance score is 14.4. In a database prepared for speech recognition experiments, 100 test subjects spoke 25 of the town names over the telephone network. In a preliminary test of an isolated word, phone-based speech recognizer, Ljolje and Riley observed 379 errors [7]. We tabulated these confused pairs of town names, along with the confusability score predicted by *word_confuse*. Because of the huge number of possible confusions, only a handful of the substitution errors (such as "Oakland"- "Oaklyn") were observed more than once. Unfortunately, some of the speech recognition errors are totally inexplicable, for instance, "Cape May Court House"- "Cheesequake".

This data was analyzed by dividing possible word pairs into 10 classes depending on their predicted distance score. There are nine equally spaced classes, with distances ranging from 0 to 2, from 2 to 4, etc. All pairs with scores above 18.0 are included in the tenth class. For pairs in each class, we compute the probability that that pair was actually confused by the speech recognizer in one of the 2500 recognition trials. The ratio for each class, (number of pairs substituted) / (total number of pairs) is plotted in Figure 3. As expected, words in a pair with low predicted confusability are rarely substituted for one another by this speech recognizer. In fact, a pair of town names for which the predicted confusability is high is more than 1000 times as likely to be confused than than a pair with a large distance. The agreement in this experiment appears to be very good.

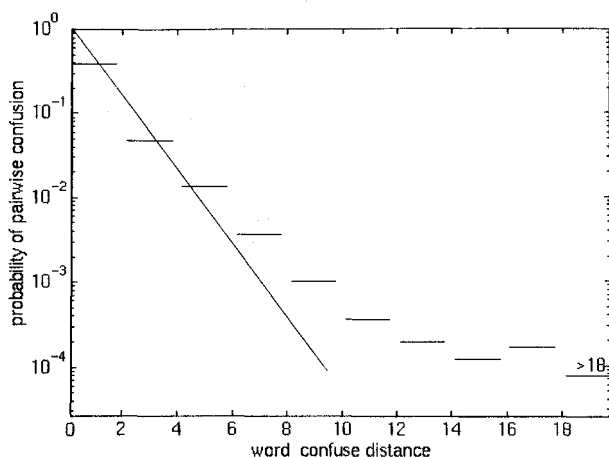


Figure 3. Substitution Probabilities for Pairs of NJ Town Names. The probability of substitution error for pairs of town names, classified by predicted similarity from *word_confuse*. Probabilities of pairwise confusion are plotted on a logarithmic scale. The expected behavior is an exponential decrease in probability as the distance increases.

In these (and other) experiments, words that had low Bhattacharyya distances were, as a class, more often confused than pairs of words with high distances. Therefore, speech recognition errors can be reduced by choosing dissimilar vocabularies. Because there are a huge number of pairs of words that have low but finite probability of confusion, we reluctantly conclude that it is *not* possible to eliminate all speech recognition errors by eliminating highly confusable words from the vocabulary list.

5. SUMMARY

This memorandum describes an efficient way to calculate the acoustic confusability between pairs of words using the Bhattacharyya distance. The program *word_confuse* computes an approximate index of similar pronunciation between all pairs of words in a list. There is general agreement between the predictions of *word_confuse* and the errors generated by speech recognizers on three vocabulary sets that were tested: letters of the alphabet, English digits, and names of towns in New Jersey. However, there are inexplicable speech recognition errors which *word_confuse* fails to predict adequately. Despite this shortcoming, speech recognition errors can be reduced by eliminating pairs of words that are likely to be confused from the vocabulary.

ACKNOWLEDGEMENTS

The authors wish to thank Barry Lively for asking the questions that stimulated this work, Andrej Ljolje for providing data from speech recognition experiments, and Allen Gorin for making helpful comments about an early version of *word_confuse*.

REFERENCES

- [1] L. R. Bahl, P. de Souza, P. S. Gopalakrishnan, D. Kanevsky, D. Nahamoo, "Constructing Groups of Acoustically Confusable Words", Proc. ICASSP 1990, pp 85-88, April 1990.
- [2] B. A. Dautrich, T. W. Goeddel, D. B. Roe, "Speaker-trained speech recognizer having the capability of detecting confusingly similar vocabulary words", U.S. Patent #4972485, Nov. 20, 1990.
- [3] L. R. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*, pp. 294-296, Prentice Hall, 1993.
- [4] A. Asadi, R. Schwartz, J. Makhoul, "Automatic modeling for adding new words to a large-vocabulary continuous speech recognition system", Proc. IEEE ICASSP-91, pp.305-308, April 1991.
- [5] C. Coker, K. Church, and M. Liberman, "Morphology and rhyming: Two powerful alternatives to letter-to-sound rules for speech synthesis", in Gerard Bailly and Christian Benoit, editors, Proceedings of the ESCA Workshop on Speech Synthesis, pages 83--86, 1990.
- [5] M. D. Riley, "A statistical model for generating pronunciation networks", Proceedings of ICASSP'91, Toronto, Canada, May 1991.
- [6] M. D. Riley and A. Ljolje, "Lexical access with a statistically-derived phonetic network", Proc. Eurospeech '91, Genoa, Italy, Sept. 1991.
- [7] A. Ljolje and M. Riley, "The AT&T phonetic-based speech recognition system", Proc. DARPA Continuous Speech Recognition Workshop, Palo Alto CA, Sept. 1992.
- [8] C. Chen, *Statistical Pattern Recognition*, pp. 57-59, Hayden, Rochelle Park, NJ, 1973.
- [9] F. Pereira, M. D. Riley, and R. W. Sproat, "Weighted Rational Transductions and Application to Human Language Processing", DARPA Workshop on HLT, March 1994, Plainsboro, NJ.