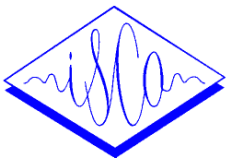

2nd International Conference on
Spoken Language Processing (ICSLP 92)
Banff, Alberta, Canada
October 12-16, 1992

RECOGNIZING PHONEMES VS. RECOGNIZING PHONES: A COMPARISON

Michael Riley and Andrej Ljolje

AT&T Bell Laboratories
Murray Hill, NJ 07974

ABSTRACT

This paper evaluates different ways of "spelling" a word in a speech recognizer's lexicon. In particular, we compare using, as the source of sub-words units for which we build acoustic models, (1) a coarse phonemic representation, (2) a single, fine phonetic realization, and (3) multiple phonetic realizations with associated likelihoods. We describe how we obtain these different pronunciations and we evaluate them on the DARPA Resource Management Task using the word-pair grammar (perplexity 60). We obtain 93.4% word accuracy using phonemic pronunciations, 94.1% using a single phonetic pronunciation per word, and 96.3% using multiple phonetic pronunciations per word with associated likelihoods.

1. INTRODUCTION

Current practice in speech recognition is to form word models from the concatenation of sub-word units. In other words, each word is spelled in terms of some finite alphabet of acoustic units. These units are variously referred to as "phones", "phonemes", or "phoneme-like units". How the spelling of a particular word is obtained is seldom discussed. In particular, whether that spelling is a coarse, abstract phonemic representation or a finer, more concrete phonetic one is not addressed. In this paper, we explore this issue and present recognition results comparing different solutions to this problem.

A specific example will help clarify the issue. Consider the word `bottle`. Its *phonemic* spelling, an abstract phonological representation like that found in dictionaries, is `/b aa t ax l/` (we use here ARPABET as the phonemic symbol set in examples [1]). This phonemic spelling, however, does not capture the finer sound variations that contextual, dialectic and speaker effects introduce. For example, in American English, the `/t/` in `bottle` will most likely be flapped `[dx]` and the `/ax l/` will most likely reduce to a syllabic `[el]`. So the most likely *phonetic* spelling of `bottle` is `[b aa dx el]`. In this paper, we use the TIMITBET symbols, a superset of the ARPABET symbols, for specifying phones [2].

Which spelling should we use for recognition purposes? If we use the phonemic spelling, we will, in effect, require the acoustic model, say, of `/t/` to handle all its allophones, which include `[t]`, `[dx]`, and `[q]`, and handle its deletion. On the other hand, if we use the phonetic spelling, we have the problem that if the speaker utters a likely, but not the *most* likely pronunciation of a word, e.g., `bottle` as `[b aa t el]`, then there is a mismatch between the permitted and true pronunciations for that rendition.

There is, of course, another choice and that is to allow for multiple phonetic spellings. So `[b aa dx el]`, `[b aa t el]`, `[b aa dx ax l]`, and `[b aa t ax l]` could all be allowed spellings of that word. This has the obvious cost of increasing the vocabulary size. It also raises the question of how many alternative spellings per word to allow. Too many and we risk recognizing an unlikely pronunciation of the wrong word over the likely pronunciation of the correct word; too few and we have a problem similar to using only the most likely pronunciation.

As a final refinement, we could attach a likelihood to each phonetic realization of word, so `bottle` is pronounced as `[b aa dx el]` with 75% probability, as `[b aa t el]` with 13%, `[b aa dx ax l]` with 10% , and `[b aa t ax l]` with 2%.

We still have to decide how many alternative pronunciations to allow, but probably now it is purely an issue of computational speed and space, since aincluding unlikely pronunciations should not hurt recognition accuracy if we take their likelihood into account. In this case, we have to have a scheme for not only finding the alternative pronunciations, but also for assigning likelihoods to them.

In the following sections, we will examine each of these possibilities. We shall, however, first briefly describe the the basic structure of the recognizer and the data set we shall use for the evaluation of these alternatives.

2. RECOGNIZER STRUCTURE

We have found a convenient way to evaluate different kinds of recognition strategies is by using a two-pass system. In the first-pass, a high-accuracy phone recognition module inputs the utterance and returns context-independent hypotheses of the phoneme (or, alternatively, phone) identities, locations and likelihoods. A lexical access module then takes this lattice and returns a word lattice based on a match be-

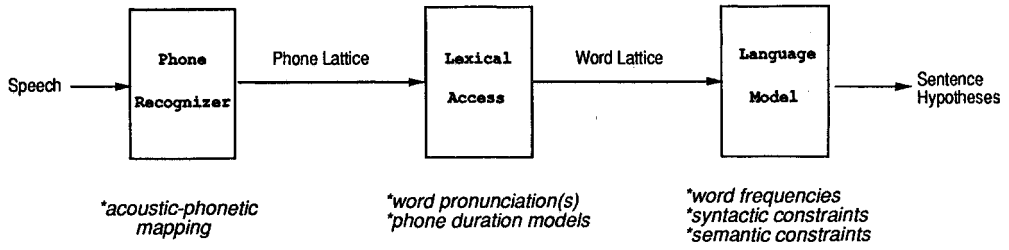

Figure 1. *Speech recognition system design: modular approach decomposes problem into three stages: (1) phone recognition, (2) lexical access, and (3) language model.*

tween the input lattice and a stored lexicon. Finally, the language model takes the word lattice and determines which paths through it are likely by combining the acoustic word likelihoods and the language model sentence priors. Figure 1 shows a block diagram of this part of the system. The final output of this first-pass is an N-best list of sentence hypotheses ordered by probability. We choose N to insure that the right answer is almost always in the N-best list. In [3,4], this system is described in more detail.

We use the widely-available DARPA Resource Management Task for our evaluations using the simple word-pair grammar (perplexity 60) as the language model; we test on the February 1989 test set. Under these circumstances, we find, for example, that with just $N = 10$, the correct sentence is present 97% of the time in our optimal system.

In the second-pass, we rescore these alternative sentences one at a time. With this scheme we have all contextual information readily available. Thus, in the rescoring we use context-dependent models including across word boundaries. We are also able, as described in Section 4, to modify the phonetic spelling with the cross-word context. Since the correct sentence is almost always present in the N-best output of the first-pass, the second-pass should give results comparable to the more conventional integrated search recognizers, other things being equal, since in the second-pass there is no quantization of boundaries or other pruning. In [4], this system is described in more detail. In [5, 6], the N-best approach is advocated and algorithms for it are described.

In the following sections, the factor that we will vary is how the words are spelled in the lexicon of the lexical access module of the first-pass and in the N-best rescoring of the second-pass. We reiterate that while we have adopted this two-pass, lattice-based system for its ease of experimentation, we are confident that our results to apply directly to more conventional, one-pass, integrated search systems.

3. PHONEMIC PRONUNCIATIONS

Let us now turn to the central question of this paper, how to spell a word for the purposes of speech recognition. Our baseline solution is to use a simple phonemic representation as found in an on-line version of the Collins American English Dictionary. When a word is not found in the dictionary, morphological rules and, failing those, letter-to-sound rules are brought to bear. We have, in fact, very briefly described the pronunciation component of the Bell Labs Text-to-Speech system, which is what we use to obtain the phonemic spellings [8]. There is usually only one allowed pronunciation per word. An exception would be, for instance, `data` (`/d ey t ax/` and `/d ae t ax/`).

With such a phonemic lexicon, we are able to achieve 93.4% word accuracy on the DARPA RM Feb '89 test set. This is remarkable given the relative abstractness of the representation. Apparently, the HMM acoustic models with Gaussian mixture distributions we use are able to model to some degree the natural allophonic variation. Nonetheless, we shall see that we can do better.

4. PHONETIC PRONUNCIATIONS

Our first modification will be to expand each phonemic spelling into its most likely phonetic spelling. Thus, for example, the `/t/` in `bottle` would be transcribed as `[dx]`, its most likely allophone in this context (for American English).

How shall we determine, in general, the appropriate phonetic transcription? A trained phonetician could do this by hand, although it would be tedious for large vocabularies. Further, when we later deal with alternative pronunciations, we will need to have likelihoods associated with the different pronunciation alternatives, which would probably be difficult task to do accurately by intuition.

We instead take a corpus-based approach. Recent work by one of the authors using statistical classification trees trained on the TIMIT phonetic database (6000 hand-segmented and labelled utterances by 600 speakers) has shown that both the most likely pronunciation and pronunciation alternatives with likelihoods can be reliably predicted [8]. Using factors such as phonemic context, lexical stress, and location of word boundaries, the decision trees assign a probability to any phonetic realization of a phoneme in context.

Figure 2 shows an example of the phone realization alternatives for the word `bottle`. The first column gives the phoneme to realize. Pairs of probabilities and phones follow. For example, the phoneme `/t/` is predicted to realize as a flap, `[dx]`, with a 71% probability and as a released `/t/` with probability 12%. Phone realizations with less than 10% probability are pruned from this figure.

To characterize the quality of this pronunciation model, we can say that it predicts the correct realization of a phoneme on held-out data from the same corpus 83% of the time, contains the correct phone in the top 5 guesses 99% of the time, and has a conditional entropy of .8 bits. This compares to the null model, in which only the phoneme to realize is used, which predicts the correct phone 69% of the time, contains the correct phone in the top 10 guesses 99% of the time, and has a conditional entropy of 1.5 bits. The

PHONEME	PHONE1	PHONE2	CONTEXT
b	1.00 b		
aa	0.92 aa		
t	0.71 dx	0.12 t	
uh	0.78 el	0.12 ax	
l	0.95 l		(if uh→ax)
	0.98 -		(if uh→el)

Figure 2. *Pronunciation network for* `bottle`. *The first column gives the phoneme to realize. Pairs of probabilities and phones follow. For example, the phoneme* `/t/` *is predicted to realize as* `[dx]` *with 71% probability and as released* `[t]` *with 12% probability. Realizations with less than 10% probability are pruned from this figure. Some realizations depend on the realization of the previous phoneme. For example, the phoneme* `/l/` *will delete with 98% probability if the previous* `/uh/` *was realized as* `[el]` *but will appear with 95% probability as* `[l]` *if the* `/uh/` *was realized as* `[ax]`

reader is referred to [8] for more precise details of the how this particular pronunciation model is generated.

In order to predict the most likely phonetic spelling of an utterance, we simply find the most likely path through its corresponding phonetic network (e.g., Figure 2). With such a phonetic lexicon, having one pronunciation per word, we achieve 94.1% word accuracy on the Feb '89 test set. his is a 0.7% improvement over using the phonemic spelling. This modest improvement suggests that while we gain something by using a finer, more precise symbol set (and thus sharper acoustic models for it), we probably lose something by not correctly guessing the appropriate phonetic transcription. We believe that usually in such cases, it is not that our technique incorrectly predicts the a priori most likely pronunciation, but that the speaker uses a less likely variant.

5. ALTERNATIVE PRONUNCIATIONS

Our next modification is to multiple phonetic pronunciations per word. In order to predict the N most likely phonetic spellings we could simply find the N most likely paths through its corresponding phonetic network. We have, however, found it inconvenient to always have to expand such a network into distinct phonetic strings, but would sometimes rather use the transistion network directly. Therefore, we limit the number of pronunciations alternatives by placing a threshold p on the minimum likelihood of a single phone realization. For example, in Figure 2, $p = 0.1$; there are no phone realizations with probability less than that. We add the proviso that we always keep the most likely phone realization for a given phoneme even if it is below the threshold so that we do not disconnect the network when p is large. For each value of p, we can consider all paths through the (truncated) pronunciation network to be the number of alternative pronunciations of that word/utterance. Note that when $p = 1.0$, we select the single, most likely pronunciation and as $p \rightarrow 0$, we get more and more pronunciation alternatives.

It is clear that unless we factor the realization likelihoods into our recognition score, that as $p \rightarrow 0$, very unlikely pronunciations with good acoustic match for incorrect words will degrade our recognition performance. To avoid this, we use Bayes Theorem to combine the acoustic and pronunciation alternative likelihoods into a single word/utterance likelihood [3,4].

Figure 3 shows word accuracy vs. p for the Feb '89 test set using an alternative pronunciation lexicon with likelihoods. We see that performance improves as we increase the bushiness of the pronunciation networks. At the best value, $p = .05$, we get 96.3% word accuracy, 2.9% better than with phonemes alone.

We note with $p = .05$, if we were to expand the phonetic network for each word in the lexicon into distinct pronunciation strings, we would get on average about 17 pronunciations per word. The short words, of course, have relatively few pronunciations, while the longer ones considerably more.

We conclude by noting that the word accuracy of just the context-independent first-pass is 95.1%. This remarkable result shows that even *context-independent* phone models considerably exceed *context-dependent* phoneme models in performance.

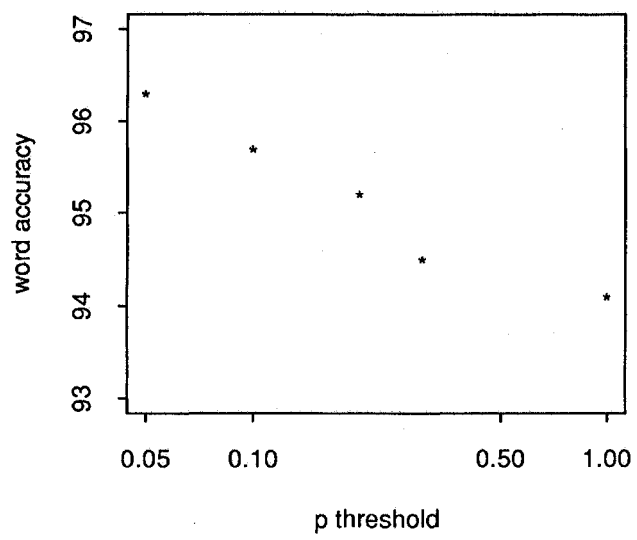

Figure 3. *Word accuracy vs. probability threshold, p, on phone realization likelihoods.*

6. REFERENCES

[1] Shoup, J. Phonological aspects of speech recognition. In *Trends in Speech Recognition*. W. Lea, ed. NY: Prentice-Hall. pp. 125-138. 1990.

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

[3] Riley, M.D. and Ljolje, A. Lexical access with a statistically-derived phonetic network, *Eurospeech '91.* Genoa, Italy. Sept. 1991.

[4] Ljolje, A. and Riley, M.D. Optimal speech recognition using phone recognition and lexical access. *ICLSP '92.* Banff, Canada and Natural Language Conference. Oct 1992.

[5] Chow, Y. and Schwartz, R. The N-best algorithm: an efficient procedure for finding the top N sentence hypotheses. *Proc. ICASSP '90.* pp 81-84. New York. Apr. 1990.

[6] Soong, F. and Huang, E. A fast tree-trellis search for finding the N-best sentence hypotheses in continuous speech recognition. Proc. ICSLP '90. pp.709-712. Kobe, Japan. Nov. 1990.

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

[8] Riley, M. A statistical model for generating pronunciation networks. *Proceedings of ICASSP '91.* Toronto, Canada. May 1991.