RESCORING MULTIPLE PRONUNCIATIONS GENERATED FROM SPELLED WORDS

Roland Kuhn, Jean-Claude Junqua, and Philip D. Martzen

Panasonic Technologies Inc., Speech Technology Laboratory
3888 State Street, Suite 202, Santa Barbara, CA 93105, U.S.A.
email: kuhn, jcj@research.panasonic.com; philip.martzen@aero.org

1. ABSTRACT

Building on earlier work [2], we show how a set of binary decision trees grown by means of the Gelfand-Ravishankar-Delp algorithm [8] can be trained to generate an ordered list of possible pronunciations from a spelled word. Training is carried out on a database consisting of spelled words paired with their pronunciations (in a particular language). We show how phonotactic information can be learned by a second set of decision trees, which reorder the multiple pronunciations generated by the first set. The paper defines the "inclusion" metric for scoring phoneticizers that generate multiple pronunciations. Experimental results employing this metric indicate that phonotactic reordering yields a slight improvement when only the top pronunciation is retained, and a large improvement when more than one hypothesis is retained. Isolated-word recognition results which show good performance for automatically-generated pronunciations are given.

2. INTRODUCTION AND RELATED WORK

Many applications of speech recognition and speech synthesis would benefit from the ability to generate pronunciations of words from their spelling [10]. The direct predecessor of the current paper is [2], which described a two-step approach to training a letter-to-sound system. In the first step, letters and phonemes are aligned via a Viterbi algorithm that inserts letter or phoneme nulls [3]. In the second step, either a binary decision tree or a Trie lookup data structure learns letter-to-phoneme rules from the aligned strings.

The current paper shows how phonotactic information can be brought into play. In the new approach, we grow not only the original "letter-only" decision trees, but also "letter-phoneme" or "mixed" trees which evaluate the probability that a given phoneme is generated from a particular letter based on both the letter sequence and the rest of the phoneme sequence. To generate an ordered list of pronunciations for a spelled word, one first generates the N most likely pronunciations from the "letter-only" trees, and then rescores and reorders these N pronunciations with the "mixed" trees. In the experiments we carried out, the top candidate after rescoring was always more likely to be correct than the top candidate before rescoring.

The recent work that has the most similarity to ours is [9]. Like us, these researchers employ letter-only decision trees, with questions about categories of letters (e.g. fricatives) as well as about singleton letters, to generate pronunciations. Unlike us, they weight candidate questions according to their distance from the current phoneme in order to favour questions that pertain to the nearby context; they also smooth the probabilities in tree leaves with probabilities in ancestral nodes, and combine probability estimates generated by trees grown on different subsets of the data. Like us, they rescore pronunciations generated by letter-only trees with phonotactic information: however, they use phonemic trigram rescoring.

Two other recent papers on the letter-to-sound problem are [1], [5]; both use neural nets, though the authors of [5] are currently experimenting with a mixed approach, in which decision trees play a part. A particularly interesting and unusual approach to the letter-to-sound problem is described in [11].

3. METHODOLOGY

Figure 1 shows a decision tree for the pronunciation of the letter 'e', with questions based only on the letters in the word in which the 'e' occurs. The numbers in the questions refer to positions relative to the 'e', with negative numbers denoting letters to the left and positive numbers denoting letters to the right. '#' is a special symbol meaning "beginning or end of word". For instance, the question in the root "+1L==r?" means "is the position after the 'e' occupied by 'r'?"; the question "-2L==#?" means "is the beginning of the word two positions to the left of the 'e'?" The rectangular boxes are the leaves of the tree; they contain probabilities that the 'e' will generate certain phonemes. The symbol for the null phoneme is '-'. The tree shown here was extracted for illustrative purposes from the upper portions of a much larger tree for 'e'.

Once 26 trees (one per letter of the alphabet) of this "letter-only" type have been grown, one submits each letter of a new word to the appropriate tree. From the information in the leaves reached, one can construct the N most probable phoneme sequences for the word via dynamic programming. However, examination of the output of this N-best letter-only phoneticizer turned up many phonotactically unlikely pronunciations. For instance, "Achilles" generated *ae k ih l l z* (reference pronunciation is *ax k ih l iy z*), and "Aaberg" generated *aa ax b er g* (reference is *aa b er g*).

We therefore determined to grow a second type of decision tree (shown in Figure 2) which estimates the probability of a phoneme being generated by a given letter, based on both the other letters in the word and the other phonemes in the pronunciation. Figure 2 again deals with the letter 'e'. Questions with an 'L' in them refer to the letter sequence of the word, questions with a 'P' to the phoneme sequence. The root question in Figure

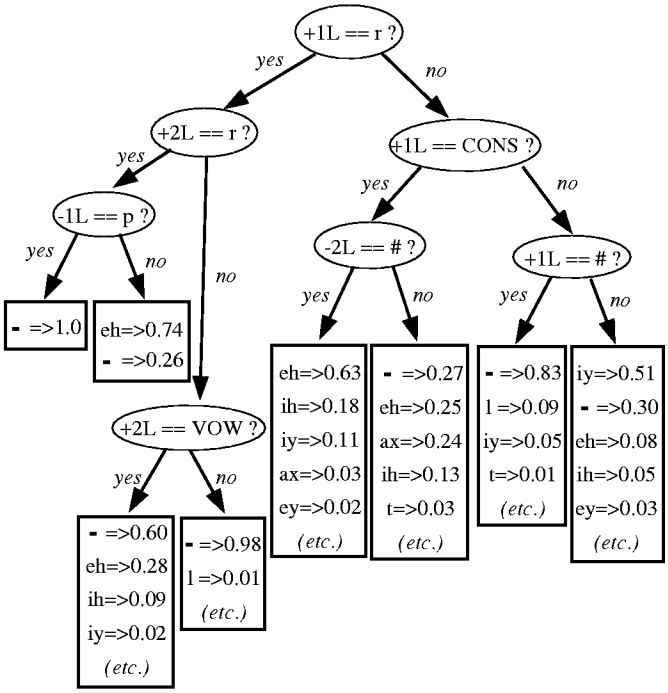

Figure 1: **Letter-only tree for 'e'**

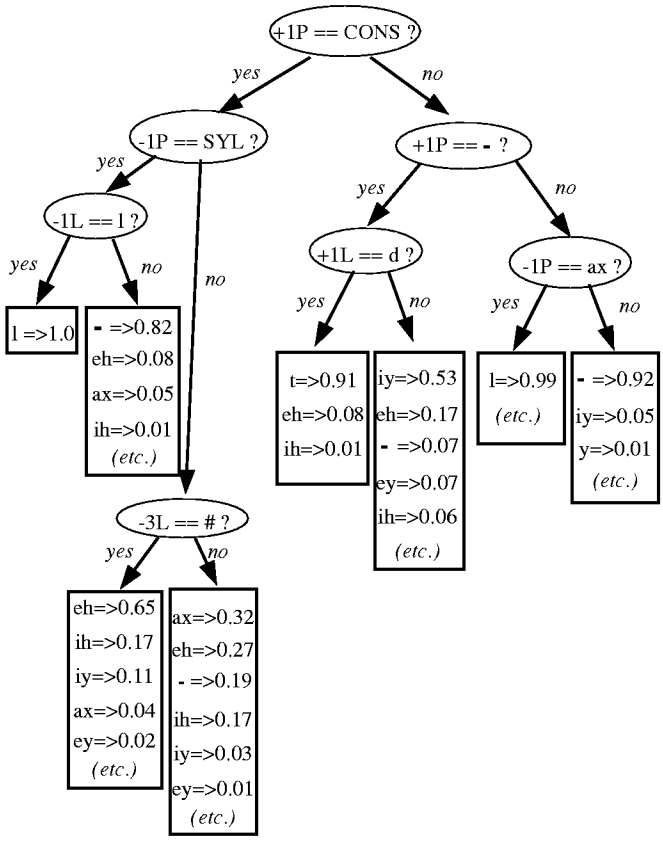

Figure 2: **Mixed tree for 'e'**

2 means: "is the next phoneme **after** the phoneme generated by 'e' a consonant?" The question in the root's *yes* child means: "is the preceding phoneme syllabic?"; the question in the root's *no* child means: "is the next phoneme the phonetic null?"

4. INITIAL EXPERIMENTS

We had originally planned to carry out experiments on both Spanish and English data. For Spanish, we carried out an initial experiment in which letter-only trees grown on 40957 spelling-pronunciation pairs were tested on 4551 spellings not in the training data. These Spanish data came from the LDC Spanish Lexicon [7]. The letter-only phoneticizer achieved string accuracy of 99.6% - that is, only 19 of the 4551 words were wrongly phoneticized. 15 of these 19 errors involved words and names of English or American origin (e.g., "Andrew", "Chrysler", "standard"). In view of the high success rate of the letter-only approach for Spanish, we decided not to carry out experiments with mixed trees for that language.

For the other experiments in this section, we generated $N = 20$ pronunciations from each spelled word using the letter-only trees, then rescored with mixed trees grown on exactly the same training data as the letter-only trees.

4.1. Databases

Table 1 gives the number of pronunciations (phonetic transcriptions) on which training and testing took place. The aligned Nettalk data we used for EXP1 were kindly provided to us by F. Yvon; this version of Nettalk is the same as that used in his thesis [11]. CMU-STL is our in-house version of the CMU pronouncing dictionary, obtained by transforming Version 0.4 of CMU [4] into STL's set of 42 phonemes and then editing out some errors. We aligned CMU-STL with Viterbi [3] and then carried out two sets of experiments, varying the relative sizes of training and test corpora.

EXP #	DATABASE	TRAINING	TEST
1	Nettalk	17,940	2,000
2	CMU-STL	10,989	98,898
3	CMU-STL	98,898	10,989

Table 1: Sources of training and test data

4.2. Tree Variants

In addition to questions about individual letters and phonemes in the context of the current letter and phoneme, we can define classes of letters and classes of phonemes about which questions can be asked. All trees used the same four letter classes: VOW ('a', 'e', 'i', 'o', 'u'), VOY (VOW plus 'y'), CONS (consonants without 'y') and CONY (consonants plus 'y'). The pseudo-letter '#', denoting beginning or end of word, does not belong to a class.

For mixed trees, we tried two schemes for defining phoneme classes: a simple scheme with only classes VOW and CONS defined, and a more complex scheme with 21 phoneme classes. Another aspect of growing mixed trees is independent of the

class definitions: should phonetic nulls in the neighbourhood of the current phoneme be kept or eliminated (pulling non-null phonemes closer)? We have tried both possibilities.

We thus experimented with four different variants of mixed-tree-growing: variant A (phonetic nulls kept, simple question scheme); variant B (nulls kept, complex question scheme); variant C (nulls discarded, simple question scheme); and variant D (nulls discarded, complex question scheme). Figure 2 was obtained under variant B.

In all experiments shown here, the letter and phoneme questions spanned positions between 5 to the left and 5 to the right of the current position.

4.3. Performance Results

All results shown below were obtained by comparing the phoneticizer's top candidate to all pronunciations for that word in the test corpus. This is a harsh criterion, since M pronunciations in the test corpus for a given word cause the phoneticizer's output to be scored wrong at least $M-1$ times. The "% WD. CORR." column shows the percentage of phoneticizer-generated pronunciations that are completely correct by this measure; the other columns show the number of phoneme substitutions, insertions, and deletions in the top candidate as a percentage of the number of phonemes in the reference. For Nettalk, six runs were carried out per variant; for all CMU-STL results (EXP2 and EXP3), two runs were carried out per variant. Recall from Table 1 that the EXP3 phoneticizer was trained on nine times as much data as the one in EXP2; as expected, EXP3 performance is better.

VARIANT	% WD CORR.	% PH. SUBS	% PH. INS	% PH. DEL
letter-only	54.2	8.81	0.92	0.78
A	57.6	8.81	0.89	1.06
B	57.5	8.98	0.86	1.03
C	57.6	8.97	0.93	1.07
D	57.9	8.67	0.79	1.06

Table 2: Nettalk results (EXP1)

VARIANT	% WD. CORR.	% PH. SUBS	% PH. INS	% PH. DEL
letter-only	46.6	10.47	1.10	1.76
A	48.1	10.91	1.08	1.79
B	48.0	10.91	1.15	1.70
C	47.5	11.02	1.11	1.83
D	47.4	11.06	1.10	1.82

Table 3: CMU-STL results (EXP2)

In the mixed trees, about 33% of the questions tended to be of 'P' (phoneme) type, and 67% of 'L' (letter) type. The percentage of times in which the top hypothesis generated by the letter-only trees was still the top hypothesis after rescoring lay in the range $58\% - 68\%$, with the lowest value (most active rescorer) obtained by variant B of EXP2 and the highest value (most conservative rescorer) obtained by variant A of EXP3; the EXP1 (Nettalk) values clustered around 61%.

VARIANT	% WD CORR.	% PH. SUBS	% PH. INS	% PH. DEL
letter-only	59.0	7.67	0.76	0.99
A	59.9	8.01	0.72	1.38
B	59.7	8.11	0.72	1.36
C	61.3	7.80	0.76	1.14
D	59.9	8.12	0.80	1.26

Table 4: CMU-STL results (EXP3)

5. LATER EXPERIMENTS

The above results showed that rescoring pronunciations obtained from letter-only trees with mixed trees slightly improves the frequency with which the phoneticizer's top hypothesis matches the reference. In practice, one would often wish to retain more than one hypothesis from the phoneticizer; we therefore require a diagnostic for multiple pronunciations.

We decided to take the top N hypotheses for a word and compare each of them to the reference pronunciation. If a distance measure between pronunciations is defined, dynamic programming can be used to find the hypothesis that is closest to the reference pronunciation. In our experiments, the distance measure was the sum of substitutions, insertions, and deletions required to turn a hypothesis into the reference. The usual comparisons are then carried out between the reference and the chosen hypothesis. We call this the "inclusion" metric. For instance, if $N = 3$ and the reference pronunciation matches one of the three hypotheses perfectly, no errors are recorded for the word. Thus, comparisons across different values of N are meaningless (the higher N is, the more forgiving the metric) but methods can be compared for fixed N.

Table 5 shows inclusion results for different values of N, for the letter-only version of the phoneticizer (*l.-only*) and for two versions in which mixed trees of type C above rescore hypotheses from the letter-only phoneticizer. In the *mix20* version, the first 20 hypotheses from the letter-only phoneticizer are rescored; in *mix100*, 100 hypotheses are rescored. Experimental conditions are as in EXP3 above ($98,898$ pronunciations from CMU-STL as training, $10,989$ as test). Note that for any value of N, the *mix20* version yields better pronunciations than the other two versions according to the inclusion metric. Clearly, guessing correct pronunciations for an English word is a delicate balance: one should neither place too much emphasis on the sounds of individual letters (as the *l.-only* version does) nor on phonotactic information (as the *mix100* version does). The *mix20* version achieves this balance better than the other two do.

Finally, we carried out isolated-word recognition experiments on three sets of 225 words from the Phonebook lexicon, with transcriptions produced by the same CMU-STL-trained phoneticizer as in EXP3 and Table 5. Table 6 shows the results, averaged over the three word sets. The line "N=1: *manual*" gives results for transcriptions produced by a linguist (one per word); since these are the reference pronunciations, the inclusion metric ("Incl. %") on this line is 100.0% by definition. The last column shows the recognition word percent correct ("Rec. %"). Note that if we take the top three pronunciations output by the *mix20* phoneticizer, the recognition result is almost as good as that for the manually-generated pronunciations (95.0% vs. 97.1%). However, with a larger lexicon three pronunciations per word might be dangerous (there would be more potentially confusable words).

VARIANT	% WD CORR.	% PH. SUBS	% PH. INS	% PH. DEL
N=1: *l.-only*	59.0	7.67	0.76	0.99
N=1: *mix20*	61.3	7.80	0.76	1.14
N=1: *mix100*	58.7	8.49	0.93	1.51
N=2: *l.-only*	70.8	5.13	0.56	0.79
N=2: *mix20*	75.1	4.67	0.43	0.59
N=2: *mix100*	67.6	6.58	0.51	0.72
N=3: *l.-only*	75.7	4.02	0.45	0.69
N=3: *mix20*	80.7	3.44	0.36	0.47
N=3: *mix100*	79.4	3.71	0.32	0.50
N=4: *l.-only*	79.3	3.34	0.39	0.58
N=4: *mix20*	83.5	2.82	0.33	0.41
N=4: *mix100*	82.9	2.93	0.28	0.43
N=5: *l.-only*	81.4	2.96	0.34	0.53
N=5: *mix20*	85.4	2.43	0.31	0.37
N=5: *mix100*	85.1	2.48	0.26	0.40

Table 5: CMU-STL: Inclusion results

VARIANT	Incl. %	Rec. %
N=1: *manual*	100.0	97.1
N=1: *l.-only*	51.9	90.2
N=1: *mix20*	52.7	90.7
N=1: *mix100*	50.2	89.9
N=2: *l.-only*	61.6	93.2
N=2: *mix20*	63.7	93.4
N=2: *mix100*	63.0	93.3
N=3: *l.-only*	65.5	94.0
N=3: *mix20*	68.4	95.0
N=3: *mix100*	68.1	94.0
N=4: *l.-only*	68.1	93.1
N=4: *mix20*	70.7	93.2
N=4: *mix100*	71.1	93.2
N=5: *l.-only*	69.5	89.7
N=5: *mix20*	71.6	89.4
N=5: *mix100*	72.4	89.2

Table 6: Phonebook: inclusion and recognition results

6. DISCUSSION

We have shown that phonotactic rescoring by mixed trees of the ordered list of pronunciations generated by letter-only trees yields a better list, especially when more than one pronunciation is kept. We also defined a metric for systems that generate multiple pronunciations. Because it relies on the notion of a reference pronunciation, this metric may still underestimate the practical usefulness of the phoneticizer - practically, we may be just as interested in generating probable "wrong" pronunciations as right ones.

Does rescoring make the phoneticizer more likely to mispronounce words in the way a native speaker would? This question cannot be answered quantitatively at present; in our opinion, however, the answer is "yes". We have studied rescoring errors, where the rescorer rejected the correct candidate provided by the letter-only trees; these errors are often similar to those a poorly educated speaker of standard American would make. From EXP3, variant C, we have the following examples (reference in brackets): "airfare" => *eh r f er* (*eh r f eh r*), "analyzes" => *ae n ax l iy z ih z* (*ae n ax l ay z ih z*), "chiseled" => *ch ay z ax l d* (*ch ih z ax l d*).

The Microsoft research described in [9] has some similarities to ours, though the phoneme set for the CMU-derived dictionary on which experiments were carried out, the amount of training data, and the nature of the baseline system all differ. Furthermore, the Microsoft group did not report results for generation of multiple hypotheses. Thus, we have not been able to compare directly phonemic trigram rescoring with mixed-tree rescoring. Our intuition (it is no more than that) is that mixed-tree rescoring is probably superior, because it permits class-based questions covering a wider context. However, the other Microsoft innovations - such as distance-weighted question-choosing criteria, smoothing of leaf probabilities, and combination of probabilities from multiple trees - probably yield a system with overall performance better than ours. In future work, we hope to experiment with these ideas, and with other European languages, particularly German and French.

7. REFERENCES

1. M. Adamson and R. Damper, "A Recurrent Network that Learns to Pronounce English Text", *ICSLP '96*, V. 3, pp. 1704-1707, Oct. 1996
2. O. Andersen, R. Kuhn, *et al.*, "Comparison of Two Tree-Structured Approaches for Grapheme-to-Phoneme Conversion", *ICSLP '96*, V. 3, pp. 1700-1703, Oct. 1996
3. O. Andersen and P. Dalsgaard, "Multi-lingual testing of a self-learning approach to phonemic transcription of orthography", *Eurospeech '95*, pp. 1117-1120, Sept. 1995.
4. *CMU Pronouncing Dictionary*, Carnegie-Mellon University, http://www.speech.cs.cmu.edu/cgi-bin/cmudict
5. N. Deshmukh, J. Ngan, J. Hamaker, and J. Picone, "An Advanced System to Generate Pronunciations of Proper Nouns", *ICASSP-97*, V. II, pp. 1467-1470, April 1997
6. T. Fukada and Y. Sagisaka, "Automatic Generation of a Pronunciation Dictionary Based on a Pronunciation Network", *Eurospeech '97*, V. 5, pp. 2471-2474, Sept. 1997
7. S. Garrett, T. Morton, and C. McLemore, "LDC Spanish Lexicon", *Linguistic Data Consortium*, Philadelphia, Pennsylvania, 1997
8. S. Gelfand, C. Ravishankar, and E. Delp, "An Iterative Growing and Pruning Algorithm for Classification Tree Design", *IEEE Pattern Analysis and Machine Intelligence*, pp. 163-174, Feb. 1991
9. L. Jiang, H.-W. Hon, and X. Huang, "Improvements on a Trainable Letter-to-Sound Converter", *Eurospeech '97*, V. 2, pp. 605-608, Sept. 1997
10. L. Lamel and G. Adda, "On Designing Pronunciation Lexicons for Large Vocabulary, Continuous Speech Recognition", *ICSLP-96*, V. 1, pp. 6-9, Oct. 1996
11. F. Yvon, "Prononcer par analogie: motivation, formalisation, et évaluation", *Thèse de doctorat (ENST96)*, École Nationale Supérieure des Télécommunications, France, May 14th 1996

We wish to thank F. Yvon of ENST, France, for providing us with his Nettalk data.