CONTEXT DEPENDENT PHONETIC STRING EDIT DISTANCE FOR AUTOMATIC SPEECH RECOGNITION

Jasha Droppo and Alex Acero

Speech Technology Group, Microsoft Research, Redmond, Washington, USA
jdroppo@microsoft.com, alexac@microsoft.com

ABSTRACT

An automatic speech recognition system searches for the word transcription with the highest overall score for a given acoustic observation sequence. This overall score is typically a weighted combination of a language model score and an acoustic model score. We propose including a third score, which measures the similarity of the word transcription's pronunciation to the output of a less constrained phonetic recognizer. We show how this phonetic string edit distance can be learned from data, and that including context in the model is essential for good performance. We demonstrate improved accuracy on a business search task.

Index Terms— speech recognition, acoustic modeling, string edit distance

1. INTRODUCTION

A standard automatic speech recognition system uses a language model and an acoustic model to find the best match between a sequence of acoustic observations and its corresponding word transcription. The language model score measures the relative likelihoods of different transcriptions independent of the acoustics, and the acoustic model score measures the likelihood of the acoustic observations for a given transcription's expected pronunciation. The best hypothesis is found using a weighted sum of these two scores.

In this paper, we introduce the use of a phonetic string edit distance (PSED) score, which measures the similarity between the phone sequence for a given word hypothesis and the phone sequence produced by a phonetic decoder, unconstrained by a pronunciation lexicon. We show how the parameters of the PSED can be easily learned from data, analyze the distribution of edits for real speech data, and demonstrate how context-dependent PSED models can improve the overall system accuracy.

The concept of learning a good string edit distance has been applied to handwriting recognition[1], pronunciation learning[2], and unit selection[3]. In [4], the authors presented a method for transducing from phones to words which used a context-independent error model. The novelty in our approach is in combining our string edit distance with the standard acoustic and language model scores, and in demonstrating the benefit of introducing context-dependent parameters.

This paper is organized as follows. Section 2 presents the theoretical background explaining how the new score fits into the standard framework. The detail of the model is presented in Section 3, as well as a description of how it is trained and evaluated. Experimental results are presented in Section 4, followed by a brief summary of the paper.

2. PHONETIC STRING EDIT DISTANCE

To understand the phonetic string edit distance, it is necessary to explore the difference between a phonetic recognition system and a word recognition system. Both systems use an acoustic model to assign a score to a sequence of acoustic observations X for a hypothesized string of phones P, but rely on different distributions for P.

2.1. Word Recognition

A simple word recognition system defines a joint score over three sequences: a word string[1] $W = \{w_1, \ldots, w_M\}$, a phonetic string $P = \{p_1, \ldots, p_N\}$, and a sequence of acoustic vectors $X = \{x_1, \ldots, x_T\}$.

The word symbols w_i are members of a known set of words. The language model $p(W) = p(w_1, \ldots, w_N)$ describes a prior distribution over word sequences of these words.

The phonetic symbols p_j are members of the phone alphabet, a small set of speech sounds that exist for the language. The pronunciation model

$$p(P|W) = p(p_1, \ldots, p_M|w_1, \ldots, w_N)$$

is a sparse distribution over the possible phone strings for every word string. It is typically described by a pronunciation lexicon, formed for each word w_i in W independently, and may consist of only one allowed pronunciation per word.

[1] A string is defined as an ordered sequence of symbols chosen from a set or alphabet.

The acoustic vectors are frame-based spectral measurements, such as mel-frequency cepstral coefficients (MFCC), and represent the audio to be recognized. The conditional likelihood $p(X|P) = p(x_1, \ldots, x_T|p_1, \ldots, p_M)$ is computed by the acoustic model, which produces a score for any possible phonetic string P from X. In modern large vocabulary speech recognition systems, this model is structured as a set of context dependent phone models.

Word recognition is performed by finding the most likely word sequence and pronunciation given the acoustic observations, as in Eq. 1.

$$\left(\hat{W}, \hat{P_w}\right) = \arg\max_{W,P} p(X|P)p(P|W)p(W) \quad (1)$$

Because the word recognition system is constrained to choose a phone string that can be decomposed into words, $\hat{P}_w$ is a string that is rigidly generated from a pronunciation lexicon, and may not be the same as a careful phonetic transcription of the utterance. Consequently, the acoustic model score $p(X|P)$ will not necessarily be maximized at $P = \hat{P}_w$. Instead, the acoustic model score indicates how close the acoustics are to the expected pronunciation of W.

2.2. Phonetic Recognition

A phonetic recognition system relates a phonetic string P to an acoustic sequence X, without reference to any underling word string W.

In this system, a phonetic language model $p(P)$ describes a prior distribution over phone strings. As in the word recognition system, an acoustic model $p(X|P)$ calculates the likelihood of the acoustic sequence from the phone string. Recognition is performed by finding the phonetic sequence that is most likely under the phonetic language and acoustic models, as in Eq. 2.

$$\hat{P}_x = \arg\max_P p(P|X) = \arg\max_P p(X|P)p(P) \quad (2)$$

Because the phonetic recognition system is less constrained, it is free to search among all possible phone strings P for the one that matches the acoustics X and has a reasonable phonetic language model score. As a result, although $P(X|\hat{P}_x)$ will probably be larger than $P(X|\hat{P}_w)$ from the word recognizer, $\hat{P}_x$ will not necessarily consist of pronunciations from the lexicon.

2.3. A Third Score

As described above (Eq. 1), a word recognition system relies on two different scores to rank the possible (W, P) hypotheses: the acoustic model $p(X|P)$, and the combined pronunciation and language model $p(P, W)$. In practice, better recognition is achieved by using a logarithmic weighting of these two scores. By convention, a weight γ is applied to the language model score.

$$\ln p(X, P, W) = \ln p(X|P) + \gamma \ln p(P, W) \quad (3)$$

In the proposed system, we incorporate the most likely phone string $\hat{P}_x$ from a phonetic recognizer, which is unconstrained by word pronunciations. To do this, Eq. 3 is augmented with a term measuring the similarity between the hypothesized phone string P and $\hat{P}_x$. This is similar to the approach in [4], which uses a language model score and a context-independent phonetic error model, but omits the acoustic model score. In Eq. 4, this term is shown weighed by a phonetic model weight β.

$$\ln q(X, P, W) = \ln p(X|P) + \gamma \ln p(P, W) - \beta D(P, \hat{P}_x) \quad (4)$$

d	r	eh	g	ax	n	ϵ	k	ae	l	ax	s
d	r	ae	g	ax	n	sil	p	ae	l	ih	s
d	r	ae	g	ax	n	sil	p	ϵ	l	ey	s

Fig. 1. Alignment between the expected pronunciation of "Dragon Palace" (center) and recognized phone string for an utterance of "Dragon Palace" (top) and "Dragon Place" (bottom).

3. PHONETIC STRING EDIT DISTANCE

Intuitively, a word string W is likely to be correct when its associated pronunciation string P is identical to the string found by our phonetic decoder P_x. Conversely, if the two phone strings are quite different, then W is probably not correct. Therefore, the phonetic string edit distance should assign a high score to similar strings, and a low score to dissimilar strings.

A common string similarity measure is the Levenshtein distance, which is defined as the minimum number of edit operations (insertions, deletions, or substitutions) needed to transform one string into another. It has the desired property that identical string pairs have zero distance, and the distance increases with the number of edit operations needed to transform one string into another.

But, as Figure 1 shows, this distance measure is insufficient. It shows how the expected pronunciation for "Dragon Palace" (center) aligns with a phonetic recognition of "Dragon Palace" (top) and "Dragon Place" (bottom). In the matched case where the user actually said "Dragon Palace", there are four edits (ae $\rightarrow$ eh, p $\rightarrow$ k, ih $\rightarrow$ ax, and sil $\rightarrow \epsilon$). In the mismatched case, there are only two edits (ae $\rightarrow \epsilon$, and ih $\rightarrow$ ey).

The problem with the Levenshtein distance is that it ignores the relative likelihood errors. It would be better if the model understood that some edits (e.g., substituting ax for ih)

are relatively common and benign, while others (e.g., substituting ey for ih) are unlikely and should be penalized. Because the differences between $\hat{P}_x$ and the expected pronunciation of the true word string are caused by acoustic confusability, accented speech, noise, or unexpected pronunciations, these patterns should be systematic and learnable.

We define the phonetic string edit distance (PSED) from a hypothesized phone string P to $\hat{P}_x$ as the conditional probability of the edits required to produce $\hat{P}_x$ from P after aligning the two strings to minimize the Levenshtein distance.[2]

$$p(\hat{P}_x|P) = p(o_1, o_2, \ldots, o_N|h_1, h_2, \ldots, h_N).$$

Unfortunately, this joint model over strings of phones has far too many parameters to train. So, instead, we restrict the model to consider left context on both the observed phone string and the hypothesized phone string only up to order j.

$$p(\hat{P}_x|P) \approx \prod_{t=1}^{N} p(o_t|o_{t-1}, \ldots; o_{t-j}, h_t, \ldots, h_{t-j}) \quad (5)$$

The model parameters are trained using two parallel sets of phonetic strings generated from the training data. For each utterance, the correct phone string P is generated by forced alignment of the reference word string using a standard pronunciation model. The observed phone string $\hat{P}_x$ is generated by running phonetic recognition on each utterance. After aligning $\hat{P}_x$ and P to minimize the number of edits between the strings, we accumulate a table that contains the necessary counts. This table is then used to estimate maximum likelihood parameters for the model. For example, to compute the likelihood of recognizing A after B when the reference contains symbols C and D,

$$p(A|B;C,D) = \frac{\#(A,B;C,D)}{\sum_i \#(i,B;C,D)},$$

where $\#(A,B;C,D)$ is the number of times in the training data that the symbols A and B occurred in order aligned with the symbols C and D. The sum over i is meant to compute how many times the same pattern occurred with any symbol in place of A.

To evaluate the model, we align $\hat{P}_x$ with the various P being considered, and assign a score to each P according to Eq. 5.

3.1. How Much Context is Appropriate

In this section, we demonstrate that the usefulness of context in modeling our data. For illustrative purposes, we collapse all possible edit operations into two classes: correct (C), which include all match edits, and incorrect (E), which includes all substitution, insertion, and deletion edits. If the edits are context independent, then the probability of seeing one error after another, $p(E|E)$ should be equal to seeing one error regardless of context, $p(E)$.

[2]Under simplifying assumptions, our phonetic string distance is just the maximum over all possible edits of the probability that string $\hat{P}_x$ is observed when the intended phone string was $\hat{P}$.

	Context Length				
	0	1	2	3	4
$P(C\|C^i)$	67.8%	78.4%	82.6%	83.5%	84.8%
$P(E\|E^i)$	32.2%	54.0%	55.1%	57.0%	60.3%

Table 1. The conditional probability of C-type and E-type edits changes dramatically with context. The longer the string of C-type edits, the more likely another C-type edit becomes. The same is true for E-type edits.

This, however, is not the case. Table 1 illustrates how the likelihood of C-type and E-type errors changes based on context. In the table, the shorthand E^i represents a string of E repeated i times. It is clear from the data that a conditionally independent modeling assumption would be inadequate for this data.

The example demonstrates how errors tend to cluster together, which indicates a model with insufficient context will misestimate the PSED. In general, there may be other patterns present in the data that we should capture. To take these possibilities into account, we trained the full model up to a context of 3 (order $j = 2$), and leave the question of model smoothing for future work.

4. RECOGNITION EXPERIMENTS

The system is evaluated on the Windows Live Search for Mobile voice search task [5]. This data contains about 1.5 million business listing queries taken from a deployed voice search application. The data is divided into 1.47 million training, 8777 development, and 12758 test utterances. The same data was used to train both the acoustic model and the phonetic string edit distance model. We used the development set to tune the model parameters and configuration, and the test set to report final accuracy improvements.

A single acoustic model was used for generating both word and phonetic string hypotheses. This model contains 27760 diagonal Gaussian components, shared among 863 states, which are in turn shared by 2612 hidden Markov models (HMM). These HMM represent the context-dependent versions of 40 phone units, as well as two silence models and three garbage models. The phonetic vocabulary size is 46, and is modeled by a trigram language model with 16 thousand n-grams. The word vocabulary is 65 thousand, and is modeled by a trigram language model with 4.7 million n-grams. The pronunciation lexicon has 92 thousand entries, with an average of 1.4 pronunciations per word. Both systems were trained from in-domain data.

Although it should be possible to build a system that incorporates both the acoustic and the phonetic score into the

Baseline	$j = 0$	$j = 1$	$j = 2$
5522	5563	5578	5591

Table 2. Number of correct utterances in the development set increases with model context.

search algorithm, the prototype system presented here is implemented using n-best rescoring. For each utterance in the development and test sets, we used the word recognition baseline system to find a list of up to 40 hypothesis word strings, along with their corresponding language model and acoustic model scores. We then computed the phonetic string edit distance for each hypothesized word string by measuring the PSED between its pronunciation and the output of a phonetic recognition system.

As expected, including more context in the PSED model improves overall system accuracy. Table 2 reports the maximum number of correctly recognized utterances in the development data as the model history variable j is increased. Because the accuracy is still increasing at $j = 2$, it is possible that increasing the context further could still improve the system, but at that point some sort of model parameter smoothing would probably be necessary.

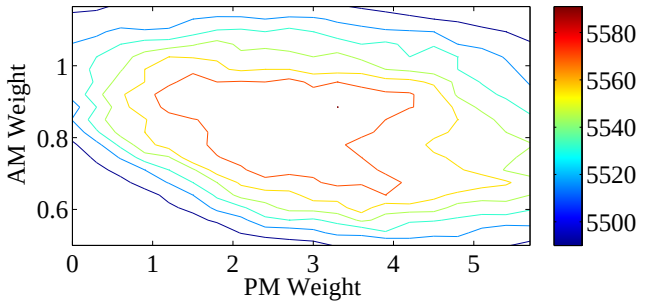

Fig. 2. Number of correct utterances in the development set, as a function of acoustic and phonetic model weights.

Using the development data, we performed a grid search for optimal values of the acoustic model weight and phonetic model weight, while holding the language model weight constant at 15. Figure 2 shows how the accuracy on the development set changes as a function of these weights when the context is set to $j = 2$. According to this figure, if the phonetic model weight is zero, the optimal acoustic model weight is 0.95. If we use a nonzero phonetic weight, the peak of Figure 2 occurs near an acoustic model weight of 0.85 and a phonetic model weight of 3.0.

Table 3 shows the result of evaluating the system on the development and test data. Out of the 12758 utterances in the test set, the oracle accuracy is only 10219 utterances, or 80%. That means, for 20% of the data, the correct hypothesis is not in the top 40 word strings identified by the word recognizer, and our rescoring system has no chance of producing a correct result.

	Weight			Correct	
System	LM	AM	PM	Dev	Test
Baseline	15.0	0.95	0.0	5522	7754
Proposed	15.0	0.85	3.0	5591	7847
Oracle	-	-	-	7084	10219

Table 3. Number of correct utterances for the baseline and the proposed system, which adds a PM score.

On the test data, the proposed score corrects 3.8% of the errors that the oracle system is able to correct with respect to the baseline. When we consider the entire data set, including utterances that the oracle system is unable to correct, the proposed system reduces the utterance error rate by 1.8%. We expect the real gain for integrating the PSED directly into the recognition process to be somewhere between these extremes.

5. CONCLUSION

In this paper, we have proposed adding a phonetic string edit distance to the standard language model and acoustic model scores in a speech recognition system. The phonetic string edit distance is trained to learn the relative likelihood of phonetic recognition strings given an expected pronunciation. We built a prototype that consists of two stages: standard n-best word recognition followed by rescoring with the new objective function, and showed that by incorporating this into a recognition system, the utterance error rate can be improved. The present gain is 1.8% relative, but with a full integration of the PSED we expect a gain between 1.8% and 3.8%.

6. REFERENCES

[1] J. Oncina and M. Sebban, "Learning stochastic edit distance: application in handwritten character recognition," *Pattern Recognition*, vol. 39, no. 9, pp. 1575–1587, 2006.

[2] E. S. Ristad and P. N. Yianilos, "Learning string edit distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 5, pp. 522–532, May 1998.

[3] G. Zweig and P. Nguyen, "Maximum mutual information multi-phone units in direct modeling," in *Proceedings of Interspeech*. ISCA, 2009.

[4] G. Zweig and J. Nedel, "Emprical properties of multilingual phone-to-word transduction," in *Proceedings of ICASSP*. IEEE, 2008.

[5] A. Acero, N. Bernstein, R. Chambers, Y. C. Ju, X. Li, J. Odell, P. Nguyen, O. Scholz, and G. Zweig, "Live search for mobile: Web services by voice on the cellphone," in *Proceesings of ICASSP*. IEEE, 2008.