Minimum Hypothesis Phone Error as a Decoding Method for Speech Recognition

Haihua Xu[1],Daniel Povey[2], Jie Zhu[1] and Guanyong Wu[1]

[1]Department of Electronic Engineering
Shanghai Jiao Tong University,Shanghai,200240,China
[2]Microsoft Research, Redmond, WA, USA
{haihua_xu,zhujie,wuguanyong }@sjtu.edu.cn, dpovey@microsoft.com

Abstract

In this paper we show how methods for approximating phone error as normally used for Minimum Phone Error (MPE) discriminative training, can be used instead as a decoding criterion for lattice rescoring. This is an alternative to Confusion Networks (CN) which are commonly used in speech recognition. The standard (Maximum A Posteriori) decoding approach is a Minimum Bayes Risk estimate with respect to the Sentence Error Rate (SER); however, we are typically more interested in the Word Error Rate (WER). Methods such as CN and our proposed Minimum Hypothesis Phone Error (MHPE) aim to get closer to minimizing the expected WER. Based on preliminary experiments we find that our approach gives more improvement than CN, and is conceptually simpler.

Index Terms: Minimum Bayes Risk (MBR), MPE, Confusion Networks, Speech Recognition, Lattice Rescoring

1. Introduction

The standard decoding formula used in speech recognition is Maximum A Posteriori (MAP) which takes:

$$\begin{aligned} W^* &= \mathbf{argmax}_W P(W|\mathcal{O}) && (1) \\ &= \mathbf{argmax}_W P(W)p(\mathcal{O}|W), && (2) \end{aligned}$$

where W is the word-sequence and $\mathcal{O}$ is the observation sequence. Assuming our models are correct, this is the decoding rule that minimizes the Bayes' Risk with respect to the sentence– that is, it minimizes the probability of choosing the wrong sentence. If we choose some W, then the Bayes' Risk equals $1 - P(W|\mathcal{O})$ (i.e. the sum of all the others' probabilities), so clearly we can minimize this by choosing the W with the largest value of $P(W|\mathcal{O})$. However, speech recognizers are typically evaluated in terms of their Word Error Rate (WER), which is the Levenshtein distance (number of insertions plus deletions plus substitutions) between the decoded output and the reference, normalized by the length of the reference. It was shown by an example in [7] that maximizing the sentence error rate does not always maximize the WER.

1.1. Confusion networks

Currently there is a widely used technique, Confusion Network Decoding [1], that attempts to exploit this difference. Confusion networks are useful because they are quick to produce from lattices[1], permit combination of the outputs of different decoders [2], and produce better word error rates than MAP decoding or (for combination of different systems) ROVER [2]. But confusion networks are conceptually rather unsatisfying as they rely on representing the alternative word sequences as a "sausage string" which is like a lossy compression of a lattice.

[1]Lattices are graphs which compactly represent the list of alternative word sequences W that correspond to an utterance, together with their timing information

1.2. Other attempts to minimize the word error rate

There have been previous attempts to minimize the expected Word Error Rate using techniques other than Confusion Networks. In [7], a method was introduced based on N-best lists. This minimized the expected word error by computing the edit distance of each pair of elements in the N-best list, and returning the element of the N-best list that minimized the weighted edit distance. Naively implemented, if the sentence was about L words long, this algorithm would take about time N^2L^2 which is probably too slow, although speedups are possible. Worthwhile word error rate improvements were reported in that paper. The decoding rule can be summarized as:

$$W^* = \mathbf{argmin}_{W_i} \sum_{j=1}^{N} P(W_j|\mathcal{O})E(W_i|W_j), \qquad (3)$$

where $E(W_i|W_j)$ is the number of word errors (the Levenshtein distance). The difficulty is that (3) is hard to compute in a lattice.

In [3], an approximation was made which made it possible to do the computation with lattices rather than N-best lists. Rather than the word error, the "time frame error" was used which (in its most basic form) is simply the number of frames on which a hypothesis differs from a reference. This makes it quite easy to compute the best path using only a lattice. The authors also introduced a parameter α which, as $\alpha \to 1$, attempts to normalize for the length of the words. Results in that paper show an lowering of word error rate, and also as expected the sentence error rate increases.

2. Minimum Hypothesis Phone Error

Our approach is to take the approximations normally used for Minimum Phone Error discriminative training [4] and use them to efficiently approximate (3).

Our decoding approach is:

$$W^* = \mathbf{argmax}_W \sum_{W'} P^{\kappa}(W'|\mathcal{O})Acc(W'|W), \qquad (4)$$

where $\mathbf{argmax}_W$ is taken over an N-best list that we derive from a lattice, and $\sum_{W'}$ is taken over the lattice itself.

$Acc(W'|W)$ is the approximated accuracy used in MPE, which we will explain in detail below. The acoustic scale κ is the acoustic scaling factor from MPE, which is normally taken to scale the (pre-scaled) language model term back down to 1.0 and leave a scale on the acoustics only, so $P^{\kappa}(W'|\mathcal{O})$ would typically be equivalent to $P(W')p(\mathcal{O}|W')^{\kappa}$ times a normalizing factor to make all sequences in the lattice sum to one.

2.1. Use of accuracy rather than error

The essential difference between our Equation (4) and Equation (3) is (after harmonizing the notation) the replacement of $E(W|W')$ with $-Acc(W'|W)$. The Levenshtein/edit distance is symmetric so $E(W|W') = E(W'|W)$, and accuracy is defined as the length of the reference minus the error, so (ignoring the approximate calculation of accuracy) we would have $Acc(W'|W) = |W| - E(W'|W) = |W| - E(W|W')$. Thus, the use of an accuracy instead of an error has the effect of biasing decoding towards longer utterances, which leads to an interesting qualitative difference between our results (which lead to more insertions and fewer deletions) and all the standard approaches we compare with which have the opposite effect. Note that the reason why accuracy rather than error was used in MPE [4] was because it was easier to approximate; the same issue applies here. This is preliminary work and we have yet to investigate canceling or partly canceling the term $|W|$ (length of W) which is implicitly introduced by the use of an accuracy. As a side note, the reader may have noticed that we could get rid of the implicit term $|W|$ by simply reversing our accuracy expression to $Acc(W|W')$, since the averaged value of $|W'|$ does not depend on W. However, due to the way we approximate the accuracy this would make our lattice-based calculation impossible.

2.2. Approximation of accuracy

Our accuracy approximation is done at a phone level (we have also done it at the word level, as the same approach applies). The phone accuracy of a word-sequence is a sum over contributions of all phone arcs q in the sequence, and the phone accuracy of an arc q is defined as:

$$\mathrm{Acc}(q|W) = \max_{z \in W} \left\{ \begin{array}{l} q, z \text{ same phone} \rightarrow -1 + 2e(q|z) \\ q, z \text{ different} \rightarrow -1 + e(q|z) \end{array} \right\}, \tag{5}$$

where z is an arc in the reference sequence W and $e(q|z)$ is the extent of overlap in time of q and z, divided by the length of z. We can compute this efficiently by pre-computing for each frame a list of arcs z that include that frame. This equation is an approximation to the notion that this accuracy needs to be -1 for an insertion, 0 for a substitution and 1 for a correct phone (which is surprising but correct given the definition of accuracy).

For any proposed word-sequence W (derived from the N-best list) in Equation (4), we can work out the expression $\sum_{W'} P^{\kappa}(W'|\mathcal{O})Acc(W'|W)$ quite easily as follows: we compute the posteriors $P^{\kappa}(q|\mathcal{O})$ of each arc in the lattice using appropriately weighted acoustic and language model scores (note that the superscript κ refers to the use of these appropriately weighted scores in the computation, not taking the final probability to a power), we compute the approximated arc accuracy $\mathrm{Acc}(q|W)$ for each arc, and our answer is:

$$\sum_{W'} P^{\kappa}(W'|\mathcal{O})Acc(W'|W) = \sum_{q} P^{\kappa}(q|\mathcal{O})Acc(q|W). \tag{6}$$

2.3. Use of N-best lists

Ideally we would like to take our $\mathbf{argmax}_W$ over all sequences in the lattice rather than just the N-best list. Because Equation (5) takes the max over a whole reference sequence rather than referring to just one reference arc, this is not possible to do trivially. It might be possible to pre-expand the lattice so that the phone identity and boundaries in a sufficiently wide context before and after any arc were always fixed. However, this would require some work to implement correctly. Currently we use N-best lists, and we find that after around $N = 20$ or $N = 40$ we see very little change in WER. This is consistent with [7], in which it was observed that the chosen hypothesis is almost always in the top 10. (Note that this is expected to vary with sentence length).

3. Experimental setup

Experiments are performed on two different corpora, named *Corpus-A* and *Corpus-B*. *Corpus-A* is a data set merged from two separate sets, one of which is Microsoft Research Asia (MSRA) corpus that has $31.5h$ of training data, while the other is China Project-863(P-863) corpus that has $113h$ of training data [8, 9]. Because data from MSRA is gender-dependent, uniformly spoken by young males, while data from P-863 is gender-independent with narrower distribution in terms of triphone coverage, we combined them into a $144.5h$ corpus to train a gender-independent cross-word triphone system using HTK [12]. *Corpus-B* is a broadcast speech data set collected from Net TV with $12.0h$ length in total, in which $10.63h$ data is employed for MPE based MAP training while $1.37h$ data is used as test data. Data in *Corpus-B* is derived from more than 10 TV channels in China, during the period July to October 2008.

Our data is coded as Mel frequency cepstral coefficients (MFCCs) with energy rather than C0 used, plus delta and delta-delta features giving the standard 39 dimensional feature vector. The system was first trained with MLE using data from *Corpus-A*, leading to an HMM set with 10 Gaussians per state in each HMM and $6k$ shared states in total. Then MPE criterion is applied to train the HMMs with four iterations using the smoothing-to-previous-iteration *I-smoothing* method [10] with $\tau = 80$. All training lattices for MPE are generated using toned-syllable bigram language models (LMs) [8]. The acoustic scaling factor κ for MPE training is set to $1/12$, the inverse of the LM scale. To generate a system for broadcast speech recognition, the HMMs trained with MPE are adapted with MAP using $10.63h$ data from *Corpus-B* with $\tau_{map} = 20$ for MAP smoothing, and then further iterations of MPE training are performed on *Corpus-B*.

Recognition LMs are toned-syllable bigram LMs. For *Corpus-A* all training transcriptions in the corpus are used to build LMs, while for the broadcast speech, transcriptions from both corpora are merged to build LMs.

Evaluation is performed on two test sets. One is MSRA test data that has 500 utterances with $0.74h$ data in total and 19.1 syllable words per utterance; the other is a $1.37h$ broadcast news dataset with 1212 utterances, with on average 19.9 syllable words per utterance. The former is gender dependent, in-office recorded and clean (read by 25 young males and 20 utterances for each), while the latter (which we call BDC) is gender independent, and contains news broadcast, review, and conversations [9]. Baseline results from Viterbi rescoring of the lattices are shown in Table 1. These are essentially the same as if we had obtained the one-best path directly from the decoder.

Note that WER results are neither true word error rate results as normally reported for English, or Character Error Rate results as normally reported for Chinese, but are based on toned-syllable errors. Syllables roughly correspond to Chinese characters and the results are therefore similar to Character Error Rate results. The word unit used for Consensus experiments is the toned-syllable unit.

Insertion penalties and language model weights were tuned to optimize the baseline Word Error Rate (as defined above). The values used were 12 for the LM scale and a word insertion penalty of 20.

Table 1: Baseline decoding results (WER)

Training methods	MSRA	BDC
MLE	26.41%	-
MPE	23.85%	-
MPE+MAP	-	33.81%
MPE+MAP+MPE	-	30.60%

Table 2: Properties of the test lattices

Test sets	WGDensity	WGDepth	GER
MSRA	58.21	41.96	6.16%
BDC	88.67	72.65	9.61%

Table 2 shows the properties of the lattices for the two test sets. WGDensity means word graph density, defined as overall number of word arcs contained in the lattice divided by the number of actually spoken words [11], while WGDepth means word graph depth, defined as the average number of word arcs per time frame [5]; and GER is graph error rate, sometimes referred to as Oracle error rate[2]. These lattices should be considered medium sized based on [5, 11].

4. Experimental results

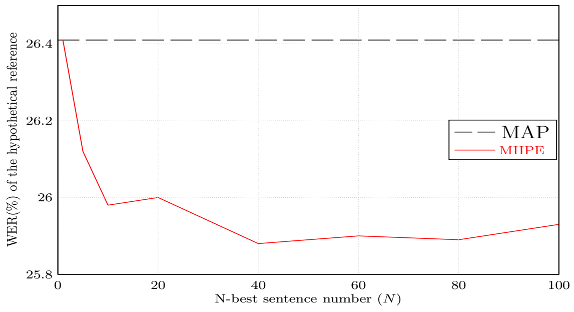
Figure 1 illustrates MHPE performance with varying N on the two sets, using our more basic models (ML for MSRA, and MPE+MAP for BDC). We can see that after around $N = 40$, further improvements are inconsistent. Results in the tables are given with $N = 40$.

Table 3: Baseline vs. MHPE on MSRA test data.

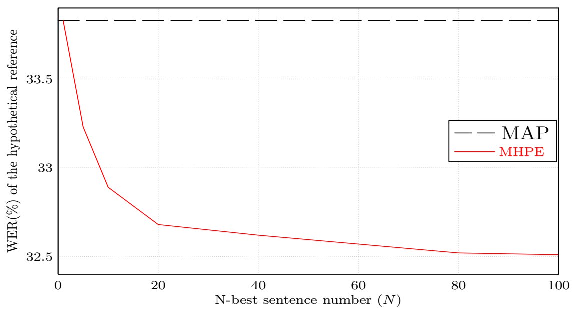
Methods	MSRA (MLE, N=40)				
	#ins	#sub	#del	SER	WER
Base	24	2333	171	95.40%	26.41%
MHPE	51	2319	107	95.40%	25.88%
Δ	+27	-12	-64	-0.0%	-0.53%

From Tables 3 and 4, the N-best implementation for the proposed MHPE lattice scoring method gave 2.0% and 3.6% relative Word Error Rate (WER) reduction on the two test data sets respectively. Note that the ratio of insertions to deletions increases, which is opposite to the normal case for these kinds of methods but expected based on the nature of our scoring function. The sentence error rate did not show a consistent change, unlike [1, 3] where it increased.

[2]GERs are obtained by the posterior probability based CN method, and are overestimates of the true GER.



(a) *MSRA test set (train: Corpus-A, MLE)*



(b) *BDC test set (train: Corpus-A MPE, MAP→ Corpus-B)*

Figure 1: WER versus N.

Table 4: Baseline vs. MHPE on BDC test data.

Methods	BDC (MAP,N=40)				
	#ins	#sub	#del	SER	WER
Base	114	7065	963	98.02%	33.83%
MHPE	187	7014	651	98.18%	32.62%
Δ	+73	-51	-312	+0.16%	-1.21%

Tables 5 and 6 show results on MPE trained models for the MSRA and BDC datasets respectively. We see an even larger improvement than on the ML trained models, with 4.0% and 5.0% relative WER reductions over the baseline decoding method. We also do not see a sentence error rate degradation.

Table 7 shows the comparison of all of our MHPE results with the CN technique. In all cases the MHPE result is better than the CN result. The insertion and deletion rates are not shown, but as expected the CN results have more deletions than the baseline, opposite to the MHPE case. The language model scale was tuned to optimize the baseline performance; in the future we intend to do MHPE and CN experiments at a range of language model scales as we anticipate that this will affect the relative performance.

5. Conclusion and further work

We have introduced a new decoding method for lattice rescoring that aims to get closer to the Minimum Bayes Risk decision rule with respect to the Word Error Rate. We have overcome some of the computational problems that these types of techniques suffer from, by using the same approximations used in MPE training. We still use an N-best list, but in a much more manageable way than, for example [7].

We have shown promising initial results, basically demonstrating that our technique gives substantial improvements versus the traditional “MAP rule” decoding approach. Further

Table 5: Baseline versus MHPE on MSRA test data.

Methods	MSRA (MPE, N=40)				
	#ins	#sub	#del	SER	WER
Base	26	2074	183	94.60%	23.85%
MHPE	47	2035	108	93.40%	22.90%
Δ	+21	-39	-75	-1.20%	-0.95%

Table 6: Baseline versus MHPE on BDC test data.

Methods	BDC (MPE,N=40)				
	#ins	#sub	#del	SER	WER
Base	109	6296	959	96.45%	30.60%
MHPE	178	6254	568	96.20%	29.08%
Δ	+69	-42	-391	-0.25%	-1.52%

work needs to be done to see in more detail how the performance compares with Confusion Networks (CN) [1] and with frame-by-frame approximations such as [3]. We also need to investigate what effect our use of an accuracy versus an error in Equation (4) has and what happens if we attempt to cancel the effect by including a term in our objective function proportional to the length of the word-sequence. Another thing that needs more investigation is the effect of using phone versus word accuracy; preliminary experiments have not shown any clear difference.

In order to have the same utility as CN, an approach like this needs to be able to support lattice combination, as in CNC [2]. This might seem simple in principle – e.g. in (3) and (4), sum over the word-sequences in all the parallel lattices rather than just one. However, in practice it might be difficult to do because word alignments will differ between systems which means so our time-based approximations may not work, and also because we cannot limit ourselves to choosing hypotheses from the individual N-best lists but need to choose "combined" hypotheses. However, these problems are solvable; for instance, to remove the overly strict dependence on time alignment one could consider ways to efficiently calculate or approximate the Levenshtein distance in a sequence-vs-lattice context, and we have some ideas on how to do this. The aim would be to get the same or better results as Confusion Networks (CN) and Confusion Network Combination (CNC) but with less severe approximations.

Table 7: MHPE versus CN

Test sets	Base	CN	MHPE
MSRA(MLE)	26.41%	25.92%	25.88%
BDC(MAP)	33.83%	32.80%	32.62%
MSRA(MPE)	23.85%	23.42%	22.90%
BDC(MPE)	30.60%	29.41%	29.08%

6. References

[1] L. Mangu, E. Brill, A. Stolcke, "Findng consensus in speech recognition: word error minimization and other applications of confusion networks", Computer Speech and Language 14 (4),291–294,2000.

[2] G. Evermann, P.C. Woodland, "Posterior probability decoding, confidence estimation and system combination", Proc. Speech Transcription Workshop, 2000.

[3] F. Wessel, R. Schlüter, H. Ney," Explicit word error minimization using word hypothesis posterior probabilities", in Proc. ICASSP'01, Salt Lake City, UT,pp.33–36,2001.

[4] D. Povey, P. C. Woodland, "Minimum phone error and I-smoothing for improved discriminative training", in Proceedings of the ICASSP'02, Vol. 1, Florida, USA, pp. 105–108,2002.

[5] D. Povey, "Discriminative training for large vocabulary speech recognition", Ph.D. thesis, Cambridge University, 2004.

[6] D. Povey, K. Brian, "Evaluation of proposed modifications to MPE for large scale discriminative training", Proc. ICASSP'07, Vol. 4, Hawaii, USA,pp. 321–324,2007.

[7] A. Stolcke, Y. König, M. Weintraub, "Explicit word error minimization in N-best list rescoring", in Procceedings of the 5th European Conference on Speech Communication and Technology, Vol. 1, Rhodes, Greece, pp. 163–166,1997.

[8] H. Xu, J. Zhu, "Towards more efficient and accurate methods for mandarin LVCSR discriminative training", in Proc. ICME'08, Hannover, Germany, pp. 981–984,2008.

[9] H. Xu, J. Zhu, G. Wu, "An efficient multistage rover method for automatic speech recognition", in Proc. ICME'09, 2009.

[10] D. Povey, D. Kanevsky, et al., "Boosted MMI for model and feature-space discriminative training", Proc. ICASSP'08, pp. 4057–4060, 2008.

[11] S. Ortmanns, H. Ney, "A word graph algorithm for large vocabulary continuous speech recognition", Computer Speech and Language 11 (1), 43–72, 1997.

[12] S. Young, et al., "The HTK Book, 3rd Edition", Cambridge University, 2006.