MINIMUM CLASSIFICATION ERROR FOR LARGE SCALE SPEECH RECOGNITION TASKS USING WEIGHTED FINITE STATE TRANSDUCERS

Erik McDermott & Shigeru Katagiri

NTT Communication Science Laboratories
NTT Corporation, Kyoto 619-0237, Japan

ABSTRACT

This article describes recent results obtained for two challenging large-vocabulary speech recognition tasks using the Minimum Classification Error (MCE) approach to discriminative training. Weighted Finite State Transducers (WFSTs) are used throughout to represent correct and competing string candidates. The primary task examined is a 22K word, real-world, telephone-based name recognition task. Lattice-derived WFSTs were used successfully to speed up the MCE training procedure. The results on this difficult task follow the classic picture of discriminative training: small acoustic models trained with MCE outperform much larger baseline models trained with Maximum Likelihood; MCE training substantially improves the performance of the larger models as well. We also present preliminary results on the 30K word Corpus of Spontaneous Japanese (CSJ) lecture speech transcription task, with a training set of 190 hours of audio.

1. INTRODUCTION

Elaborating a study first presented in [1], we here describe the use of discriminative training based on Minimum Classification Error (MCE) to obtain significantly more effective acoustic models, raising recognition accuracy and reducing model size. This is the classic MCE vs. Maximum Likelihood Estimation (MLE) picture [2], but for a much more challenging task than used in most MCE studies to date. We also present preliminary results on the Corpus of Spontaneous Japanese (CSJ) lecture speech transcription task that has been used by several groups in Japan [3]. The task vocabularies used here, 22K and 30K words, and the training set size for the CSJ task (190 hours), are the largest reported for any MCE study to our knowledge. These results, together with other recent work [4], should lay to rest concerns that MCE could only be used for small vocabulary or noise-free tasks.

When training on large data sets, the heavy computational load of discriminative training creates the need for approximations that speed up training substantially. One such approximation is to use lattices from a previous recognition pass for a given acoustic model as the recognition grammar for each utterance. This has been used for the MMI framework for years now [5, 6, 7]. Surprisingly, no one to our knowledge has reported results for lattice-based MCE training – though discussion of this can be found in [8]. The results presented here show that this method is effective for MCE as well. Note that lattices are only used to speed up training; the parameter update just uses the top string or the top N incorrect strings.

In our approach, Weighted Finite State Transducers (WFSTs) [9] are used throughout to model both the recognition grammar (from which the top incorrect strings will be found) and the correct word sequence. The recognition grammar is either the target language model, or a subset thereof, such as a unigram or an utterance-dependent lattice derived from a previous recognition pass (with segmentation information removed). The same decoder can then be used to generate the correct and best incorrect string scores required by the MCE training procedure. For the name recognition task, we describe how modeling the various hesitations and fillers as hidden variables – through "enriching" the WFST for the correct string – can allow the MCE training procedure to focus more directly on optimizing name recognition error.

2. MCE TRAINING

The MCE framework for HMMs has been described in several sources [2, 10]. For every training token $\mathbf{x}_1^T$, a misclassification measure compares a log-likelihood based discriminant function $g_k(\mathbf{x}_1^T, \mathbf{\Lambda})$ for the correct string category S_k with a weighted sum over the discriminant functions for the incorrect strings:

$$d_k(\mathbf{x}_1^T, \mathbf{\Lambda}) = -g_k(\mathbf{x}_1^T, \mathbf{\Lambda}) + \log \left[\frac{1}{M-1} \sum_{j \neq k}^{M} e^{g_j(\mathbf{x}_1^T, \mathbf{\Lambda})\psi} \right]^{\frac{1}{\psi}}. \quad (1)$$

The discriminant functions include both acoustic and language model scores. This measure is then input to a 0-1 loss function, typically a sigmoid. If we take ψ to be very large, only the best incorrect category is used, the sign of the misclassification measure reflects the correctness or incorrectness of the classification decision for training token $\mathbf{x}_1^T$, and the loss function reflects string-level classification error. Small values of ψ can be used to "unweight" the top incorrect categories, which can help generalization. This is examined in Section 3.8; in all other experiments, only the top incorrect string was used.

For all the experiments described here, the overall MCE loss function was optimized in batch mode using the Quickprop algorithm, implemented in parallel over many machines [2, 4].

3. TELEPHONE-BASED NAME RECOGNITION

The task is that of recognizing Japanese names spoken over the telephone, in the real-world setting of people telephoning a call center to request that information (catalogs, pamphlets, etc.) be mailed to them. The goal was to evaluate the performance of an off-line speech recognition system used to transcribe the contents of each call, in particular the caller's name.

3.1. Database and data characteristics

A database of more than 40 hours of utterances in this real-world setting was collected and transcribed. Most utterances are from different callers. Every utterance contains a family name and a given name – we do not address the problem of handling utterances that do not conform to this pattern. There is great variation in speaking style. A large proportion of the calls were made from cellular phones and noisy acoustic environments. Another feature of the data is the presence of false starts, hesitations and various fillers ("eeto...", "ano..." etc. at the beginning of the utterance; "desu", "to mooshimasu", etc., at the end) that can bracket the caller's name.

From the overall set, a training set of 35,500 utterances (about 39 hours of audio) and a test set of 6,428 utterances were selected. All utterances were converted to sequences of 21-dimensional feature vectors, consisting of RASTA-filtered MFCCs, their deltas, and delta energy.

3.2. WFST-based recognizer / language model

The experiments described here are based on the SOLON recognizer designed at NTT Communication Science Laboratories [11]. SOLON uses a time-synchronous beam search through a WFST for decoding speech input. It has been applied to language models with vocabularies of up to 1.8 million words [11].

A WFST was designed to model 16576 family names and 5744 given names, about 99.8% of the family and given names found in a corpus of 130,000 name listings. Unigram probabilities were estimated from the corpus and incorporated into the WFST. This set of names covers all utterances in the training set described above – out-of-vocabulary (OOV) utterances were either removed from the training set, or the corresponding names added to the lexicon. In addition to the set of names, some simple types of hesitations and pauses are modeled, along the lines of the patterns described in Section 3.1. The results presented in the following sections include a small OOV error rate.

Via composition with a triphone connection network, the recognition WFST models both cross-word and within-word triphones. After weight-pushing and network optimization, the final network, *CLG*, contained 489,756 nodes and 1,349,430 arcs. The size of the vocabulary modeled (22,320 names in all), and the fact that both cross-word and within-word triphones were used, are significant improvements compared to the initial study, [1].

3.3. Context-dependent model design using decision trees

Triphone models of several sizes were clustered using phonetic decision trees. After tree construction, the number of Gaussians was increased iteratively by performing Viterbi training on the training set and splitting the Gaussian with the largest mixing weight. Models were created using two different log-likelihood increase thresholds during tree construction, resulting in two sets of trees with different tree depths and sizes. The shallower tree set, containing 187 triphone states, was used to make three models with, respectively, 4, 12 and 20 Gaussians per leaf node mixture; the deeper tree set, containing 547 triphone states, was used to make four models with, respectively 12, 20, 36 and 50 Gaussians per leaf node mixture. Thus, in all, seven HMM configurations were used, with respectively, 748, 2244, 3740, 6564, 10940, 19692 and 27350 Gaussian pdfs.

Table 1. Name error rate (NER) and total number of Gaussians for different HMM configurations trained with MLE/Viterbi-training

# Gaussians	NER (%)
748	43.07
2244	37.74
3740	35.42
6564	32.85
10940	31.39
19692	30.05
27350	29.58

Table 2. Name error rates for various configurations and MCE design methods

# Gaussians	Tri-loop	FullLM	Lattice	Lattice-RTr
748	35.44	31.38	32.78	31.04
2244	31.69	27.22	28.06	27.88
3740	31.04	-	-	26.77
6564	29.69	26.84	28.21	26.38
10940	27.57	-	-	26.33
19692	26.83	-	-	25.62

Given these models, name recognition accuracy was evaluated on the test set using beam search through the language model WFST. The recognition results for each configuration are shown in Table 1.

3.4. Training with a triphone loop

Using a connected triphone loop, constrained by a phoneme bigram, as the language model during training can be a simple way of obtaining significant improvements over the MLE / Viterbi-training baseline. The hope is that improved phoneme recognition accuracy will translate into better word recognition when given the target language model.

MCE training with a bigram-constrained triphone loop, using Quickprop, was carried out for six of the MLE / Viterbi-training baseline configurations. The name recognition results for the MCE-trained model are shown in Table 2 in the column for *Tri-loop*.

3.5. Training with the full name grammar and flat transcription WFSTs

The use of MCE with the full 22K language model WFST was then evaluated. Doing so results in a long training time, but is directly aimed at the recognition target. The approach is simply to use the full LM WFST with the SOLON decoder to generate the top incorrect strings required in implementing the derivative of Equ. (1).

Equ. (1) also requires a model for the discriminant function for the correct string category (often referred to as the "numerator" term in MMI studies), including both acoustic and language model scores. There are several ways of implementing this. The approach adopted here, with a view to minimizing the possibility of mismatch of both LM and acoustic score calculation, was to represent the utterance transcription as a WFST derived directly from the full language model WFST. Each of the transcription WFSTs was obtained via composition of the full language model WFST,

CLG, with the transcribed word sequence *W* for each utterance. The result is a simple and usually flat WFST, *CLGW*, modeling the utterance's transcribed word/phonetic content and containing the correct LM score. (Note that the output symbol alphabet used in both *CLG* and *W* contains both names and fillers). The SOLON decoder can then be used for recognition, yielding the top competitors, as well as for the correct strings. The resulting scores and segmentations can then be used to implement the MCE optimization procedure based on Equ. (1). This WFST-based implementation of MCE is similar to that used in [4] with the MIT GALAXY system.

Accordingly, three of the MLE-trained model configurations, with 748, 2244 and 6564 Gaussian pdfs, were used as initial models for string-level MCE training via Quickprop. The test set results are shown in Table 2 in the *FullLM* column.

3.6. Training with lattice-derived WFSTs and flat word transcription WFSTs

The training procedure can be significantly sped up by using lattices generated for all training utterances from a previous recognition pass. The idea is that each lattice embodies a subset of likely recognition candidates that will not change much as long as the acoustic model does not change much. Using lattice-derived WFSTs as the utterance-dependent language model can be much faster than using the full language model. Other than the dynamic swapping in of a WFST for each utterance, this entails no changes to the MCE training procedure.

Recognition lattices were generated for all training utterances, for six of the MLE-trained model configurations. Each lattice is made during beam search through the full LM WFST. The lattice gets a new arc for every output-emitting arc activated during search. This usually happens somewhere in the word once a phone sequence becomes unique. When generating the lattices, beam width was set to be about three times larger than that used during training with the full LM. Segmentation time information was removed from the lattice, which was then saved in WFST format. This can then be used by the decoder instead of the full LM.

Across the configurations used, the resulting set of lattice-derived WFSTs had on average 800 arcs each, in contrast with 1,349,430 arcs for the full LM. MCE training was then run for the same three model configurations and same settings as for the full LM training described in Section 3.5 – same Quickprop parameters and same number of iterations. Training is now much faster than training with the full LM, by a factor of about three. The test set results are shown in Table 2 in the *Lattice* column . The results are very close to those for MCE training with the full LM.

3.7. Training with lattices and "rich transcription" WFSTs

Having established that using lattices as the recognition grammar during MCE training is a practical way of significantly speeding up the learning procedure, we next considered a variant that we thought could further boost performance on this task.

The target is to recognize the family name and the given name in each utterance. As described in Section 3.2, the full LM models a number of fillers and hesitations that can bracket the names. The presence or absence of these extra terms is indicated in the utterance transcription; that transcription will then be taken as the correct string for the MCE training procedure. However, given that it does not matter whether the filler terms are recognized correctly as long as the names are recognized, it is reasonable to wonder if the training procedure couldn't be modified to focus just on name recognition.

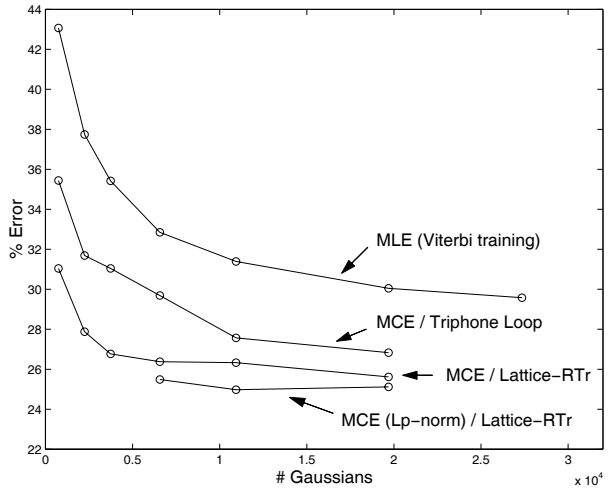

Fig. 1. Name error rates for MLE- and MCE- trained recognizers

Rather than adhere to the utterance transcriptions, a WFST were created for each utterance so as to represent the transcription names, bracketed by all possible fillers in the LM. These "rich transcription" WFSTs were generated simply by removing the filler output symbols in the full LM WFST, *CLG*, and composing the result with a small WFST made from just the two names of interest, *N*. The resulting WFST, *CLGN*, models the two names of interest, as well as all possible fillers around those names. Though a simpler method could avoid composition with the full LM WFST, this approach reduces the possibility of mismatch. As with the flat transcription WFSTs used in Sections 3.5 and 3.6, the rich transcription WFSTs contain the LM scores of the correct names.

The rich transcription WFST is then used with the decoder to calculate the MCE discriminant function for the correct string; the discriminant function for the top incorrect string is obtained from name-based filtering of the top recognition candidates. The resulting MCE loss function now corresponds directly to family name and given name string recognition error, and does not reflect filler recognition error. This is not the case when using a flat word transcription WFST to model the correct string. The training procedure will now focus on the names in question, and not on accurate recognition of the fillers. If a filler is recognized, correctly or not, by the language model (be it full LM or lattice LM), it will tend to be recognized by the rich transcription WFST as well. The MCE gradient for these terms will tend to cancel out, and focus only on the matter of interest, the name content.

Accordingly, MCE training was run using the rich transcription WFSTs to model the correct string instead of the flat word transcription WFSTs used in the previous experiments. The same set of lattice-derived WFSTs was used to speed up the search for the top incorrect string categories. Learning proceeded more quickly than before, rapidly raising the recognition rate. For some of the configurations, lattices had to be re-generated half-way through the training procedure. The test results are shown in Table 2 in the column for *Lattice-RTr*.

Table 3. Name error rates after MCE training using N L_p-normed incorrect strings

# Gaussians	6564	10940	19692
NER	25.49	24.98	25.12

Table 4. Error rates for MCE vs. MLE baseline on the Corpus of Spontaneous Japanese

# Gaussians	MLE	MCE / Unigram
23856	24.7	23.8

3.8. Use of L_p normed N-best incorrect candidates

All the experiments described so far used the single best incorrect candidate in implementing Equ (1). Using more incorrect candidates, and a choice of ψ that prevents a few top incorrect categories from dominating the L_p-norm in Equ (1), may help generalization to test data. This is related to "acoustic scaling" used in MMI studies [6]. Using $\psi = 0.1$ and the top 30 incorrect strings during lattice-based MCE training with rich transcription WFSTs resulted in additional gains in performance, as shown in Table 3. Fig. 1 illustrates these and other results on the telephone-based name recognition task.

4. CORPUS OF SPONTANEOUS JAPANESE

We very briefly summarize preliminary work on the Corpus of Spontaneous Japanese (CSJ) lecture speech transcription task. This is a large-scale, spontaneous speech recognition task [3]. The standard male *A* set consisting of 154,000 utterances (approximately 190 hours of audio) was used for training. The trained models were tested on the standard test set of 10 lecture speeches, each from a different speaker, comprising 130 minutes of audio in total. The trigram language model WFST used in the recognition tests models 30,000 words, and contains 6,138,702 arcs. The feature vectors used consist of 38 MFCCs, deltas, and delta-deltas. A single acoustic model was used, with approximately 3000 states and 8 Gaussians per state, for a total of 24,000 Gaussian pdfs. MCE training was performed using the unigram WFST for this task, containing 494,845 arcs. Beam search through the full unigram is very fast, about 3-5 times real time; MCE training could be carried out without resorting to lattices. The "rich transcription" WFST approach described in Section 3.7 was used to model the correct transcription while allowing the optional insertion of silence models between words. On the training set, word recognition accuracy with the unigram rose from 43.6 % to 55.4 %, over 11 iterations. Parallelized over 40 processors, training time was about 16 hours. The test results, using the trigram language model, for both the MCE-trained model and the baseline, are shown in Table 4.

5. CONCLUSION

This study evaluated discriminative training based on the Minimum Classification Error framework in the context of difficult large-vocabulary speech recognition tasks. Focusing on a telephone based name recognition task, several model configurations were used to compare WFST-based MCE training with an MLE / Viterbi Training baseline. Strikingly, the smallest acoustic model, with 748 Gaussians, trained with MCE, outperforms the baseline models with up to 10940 Gaussians. The best MCE-trained model, with 10940 Gaussians, yielded a relative performance improvement of 20.4%. Lattices were successfully used to speed up the training procedure; training with a simple triphone loop yielded good results too. Preliminary results were described on the 190 hour, 30K word Corpus of Spontaneous Japanese lecture transcription task, with MCE (trained on a unigram LM but tested on a trigram LM) yielding a 3.6% relative reduction in error.

6. ACKNOWLEDGMENT

We would like to thank the MIT Spoken Language Systems Group for providing many of the WFST tools used in this work.

7. REFERENCES

[1] E. McDermott, A. Biem, S. Tenpaku, and S. Katagiri, "Discriminative training for large vocabulary telephone-based name recognition," in *Proc. IEEE ICASSP*, 2000, vol. 6, pp. 3739–3742.

[2] E. McDermott, *Discriminative Training for Speech Recognition*, Ph.D. thesis, Waseda University, School of Engineering, March 1997.

[3] T. Kawahara, H. Nanjo, T. Shinozaki, and S. Furui, "Benchmark test for speech recognition using the corpus of spontaneous japanese," in *Proc. of the Spontaneous Speech Processing & Recognition Workshop*, Tokyo, 2003, pp. 135–138.

[4] E. McDermott and T. J. Hazen, "Minimum Classification Error training of landmark models for real-time continuous speech recognition," in *Proc. IEEE ICASSP*, 2004, vol. 1, pp. 937–940.

[5] Y. Normandin, R. Lacouture, and R. Cardin, "MMIE Training for Large Vocabulary Continuous Speech Recognition," in *International Conference on Spoken Language Processing*, 1994, vol. 3, pp. 1367–1370.

[6] D. Povey and P.C. Woodland, "Minimum Phone Error and I-smoothing for improved discriminative training," in *Proc. IEEE ICASSP*, 2002, vol. 1, pp. 105–108.

[7] L. Nguyen and B. Xiang, "Light supervision in acoustic model training," in *Proc. IEEE ICASSP*, 2004, vol. 1, pp. 185–188.

[8] R. Schluter, W. Macherey, B. Muller, and H. Ney, "Comparison of discriminative training criteria and optimization methods for speech recognition," *Speech Communication*, vol. 34(3), pp. 287–310, 2001.

[9] M. Mohri, F. Pereira, and M. Riley, "Weighted Finite State Transducers in Speech Recognition," in *Proc. of Automatic Speech Recognition Workshop*, 2000, pp. 97–106.

[10] S. Katagiri, C-H. Lee, and B.-H. Juang, "New discriminative training algorithms based on the generalized descent method," in *Proc. IEEE Worshop on Neural Networks for Signal Processing*, 1991, pp. 299–308.

[11] T. Hori, C. Hori, and Y. Minami, "Fast on-the-fly composition for weighted finite-state transducers in 1.8 million-word vocabulary continuous speech recognition," in *International Conference on Spoken Language Processing*, 2004.