



An Approach to Efficient Generation of High-Accuracy and Compact Error-Corrective Models for Speech Recognition

Takanobu Oba, Takaaki Hori, Atsushi Nakamura

NTT Communication Science Laboratories, NTT Corporation,
2-4, Hikaridai, Seika-cho, Soraku-gun, Kyoto, Japan

{oba, hori, ats}@cslab.kecl.ntt.co.jp

Abstract

This paper focuses on an error-corrective method through reranking of hypotheses in speech recognition. Some recent work investigated corrective models that can be used to rescore hypotheses so that a hypothesis with a smaller error rate has a higher score. Discriminative training such as perceptron algorithm can be used to estimate such corrective models. In discriminative training, how to choose competitors is an important factor because the model parameters are estimated from the difference between the reference (or oracle hypothesis) and the competitors. In this paper, we investigate the way how to choose effective competitors for training corrective models. Particularly we focus on word error rate (WER) of each hypothesis and show that a higher WER hypothesis rather than the best-scored one works effectively as a competitor. In addition, we show that using only one competitor with the highest WER in an N-best list is very effective to generate accurate and compact corrective models in experiments with the Corpus of Spontaneous Japanese (CSJ).

1. Introduction

As the performance of automatic transcription by speech recognition has increased, a type of downstream processing using the transcripts as inputs becomes important for various speech applications. Examples of such downstream processing include correction of remaining recognition errors in the transcripts, extraction of linguistic and semantic information from the transcripts, etc. Discriminative training approaches such as the perceptron algorithm [1], adaboosting [2] and minimum sample risk [3] are widely used there. This paper focuses on the correction of errors as such a type of downstream processing.

Some previous studies have reported the effect of reranking approach on the correction of speech recognition results. Each hypothesis in a word N-best list or a word lattice of the speech recognition is rescored so that the score of a hypothesis with the lowest word error rate (WER) becomes larger than the score of the others.

A corrective method based on N-gram count features has been proposed recently [4, 5]. The hypotheses are reranked by rescored each hypothesis with the summation of weighted N-gram counts extracted from the hypothesis. The weights are estimated through the discriminative training. N-grams from the lowest-WER hypothesis, which is called the 'oracle' hypothesis, are given large weights, whereas N-grams from the other hypotheses competing against the oracle are given small weights. Thus the reranking process works so as to discriminate the oracle and competing hypotheses, and the resultant reranking model can play as an error-corrective model.

To build an error-corrective model, we generate N-best lists or word lattices from the whole training data, and then utilize the huge amount of hypotheses for the parameter estimation. As a result, a undue large corrective model is often generated after considerable computation. Hence it is important to limit the hypotheses to only ones which are effective for generating accurate and compact models. Our aim in this paper is to investigate how to find such hypotheses from each N-best list.

Through discriminative training, the corrective model acquires error patterns from the difference between the oracle and competing hypotheses. It is preferable that many error patterns are obtained, which could appear in recognizing unseen speech.

In [4, 5], the perceptron algorithm is used to discriminate the oracle hypothesis and the tentative recognition result, i.e. the best scored hypothesis after rescored with the corrective model at each iteration step in training. Since the best scored hypothesis becomes similar to the oracle hypothesis through the iteration steps, the corrective model is trained so as to discriminate the similar hypothesis from the oracle hypothesis. Even in the first iteration, only hypotheses similar to the oracle tend to be selected as competitors because hypotheses with high speech recognition score relatively have low WER, i.e. similar to the oracle. This is not necessarily suitable for correction of various errors in recognizing unseen speech.

In this paper, we study on the use of WER of each hypothesis rather than the score given to the hypothesis by a recognizer for choosing effective competitors. By choosing hypotheses with a higher WER, in particular, it is expected that acquired patterns of errors in competitors become more various.

Based on this idea, we modified the training method of corrective model using the perceptron algorithm so that it uses higher-WER hypotheses as competitors. In experiments using the Corpus of Spontaneous Japanese (CSJ) [6], it is shown that a higher-WER hypothesis rather than the best-scored one works effectively as a competitor in the perceptron algorithm. In addition, we show that the use of only a single competitor with the highest-WER in the N-best list is more effective to generate accurate and compact models.

This paper is organized as follows: in section 2, we explain the error correction procedure based on the reranking of speech recognition results and the parameter estimation method that employs the perceptron algorithm. In section 3, we explain our modification of the training procedure. Section 4 provides experimental results to find out the relation between the performance of the corrective models and WER of the sequences used for the training. Section 5 contains the discussion and describes future work. Section 6 concludes the paper.

2. Corrective Model with Perceptron

Given speech data indexed by k , we denote the hypotheses of speech recognition for the k -th data as \mathbf{Hyps}_k and the n -th hypothesis as $w_{k,n}$. The recognition score of $w_{k,n}$ is denoted as $\log P_{k,n}$. \mathbf{Hyps}_k is generally assumed to be an N-best list or word lattice.

The corrective model reranks the hypotheses based on linear interpolation,

$$w_k^* = \arg \max_{w_{k,n} \in \mathbf{Hyps}_k} \{\lambda \log P_{k,n} + \phi(w_{k,n}) \cdot \alpha\}. \quad (1)$$

The recognition score is compensated by an inner product of an N-gram count vector $\phi(w_{k,n})$ and a weight vector α . λ is a parameter for adjusting the scaling between the recognition score and the score of the corrective model.

We can use different discriminative training methods to estimate parameter α . Here, we explain the perceptron algorithm depicted in Figure 1. T is the upper limit of the iterations and K is the total number of training data sets. w_k^{oracle} indicates the oracle hypothesis. In this training procedure, w_k^* is selected as the best hypothesis after reranking under certain α , then the two sequences, namely w_k^* and w_k^{oracle} , are discriminated. The update of α is based on the N-gram counts.

The averaged parameter $\alpha_{ave} = \sum_{t,k} \alpha_k^t / KT$ performs accurately in the actual analysis for test sets. α_k^t is the parameter α after processing the k -th training data in the t -th iteration.

```

1: Set  $\alpha = [0, \dots, 0]^T$ 
2: For  $t = 1, \dots, T$            #iteration
3: For  $k = 1, \dots, K$          #data set
4:  $w_k^* = \arg \max_{w_{k,n} \in \mathbf{Hyps}_k} \{\lambda \log P_{k,n} + \phi(w_{k,n}) \cdot \alpha\}$ 
5:  $\alpha = \alpha + \phi(w_k^{oracle}) - \phi(w_k^*)$ 

```

Figure 1: Perceptron algorithm.

3. Modification of the Perceptron Algorithm

We describe two ideas to modify the perceptron algorithm so that higher-WER hypotheses are used as competitors and the model size becomes small.

In section 1, we have mentioned that choosing the competitor by the recognition score is not necessarily suitable for the collection of various errors in recognizing unseen speech. One idea to cancel the effect of the score is to set $\lambda = 0$, where we assume the competitor is randomly selected when there are equivalent competitors with the same score. Especially when $\lambda = 0$ and α has many zero components, i.e. early steps of training, it is randomly selected from all the hypotheses in \mathbf{Hyps}_k . By canceling the recognition score, competitors including more errors can be selected. For that matter, more erroneous competitors might be obtained if we set $\lambda < 0$. However, λ needs to be set as a positive constant in the correction of speech recognition for test sets because the recognition score is the most important factor. For disambiguation, we denote λ for training by λ^{train} and that for testing by λ^{test} , respectively. In other words, our first idea is to introduce $\lambda^{train} \leq 0$ and $\lambda^{test} > 0$.

On the other hand, although more error patterns will be acquired by setting $\lambda^{train} \leq 0$, it will increase the number of parameters of the corrective model, where the number of parameters means the number of non-zero components in α . The second idea is to restrict \mathbf{Hyps}_k to higher-WER hypotheses for the generation of a compact model.

Now an n -best list is reranked in ascending order of WER and this order is denoted by a superscript index as $w_k^1, w_k^2, \dots, w_k^n$; specifically $w_k^1 = w_k^{oracle}$ and w_k^n is the highest-WER hypothesis in the N-best list. We denote the subset of the n -best $\{w_k^i : i = 1, x \leq i \leq y\}$ as $\mathbf{Hyps}_k(x, y)$. This is the subset including the oracle hypothesis of each utterance. Specifically, $\mathbf{Hyps}_k(2, n)$ consists of all the hypotheses in the n -best and $\mathbf{Hyps}_k(n, n)$ consists of only the oracle and the highest-WER hypothesis, that is $\{w_k^1, w_k^n\}$. By replacing \mathbf{Hyps}_k with $\mathbf{Hyps}_k(x, y)$ in Figure 1, the WER-based competitor restriction is introduced in the algorithm.

4. Experiments

CSJ includes many lecture speech data and their transcriptions. To generate speech recognition hypotheses, we prepared an acoustic model (AM) and two language models (LM). The AM is MCE trained tri-phone HMM with 5,000 states-32 Gaussians and each LM is a back-off tri-gram model with Kneser-Ney smoothing. We used the speech recognition system SOLON, which has been developed at NTT Communication Science Labs, to generate the word N-best lists. It is a decoder based on a weighted finite state transducer and it can provide an efficient fast search by using a fast on-the-fly composition algorithm [7].

4.1. Data set

We first divided 2,692 lecture data into four groups as shown in Table 1. In addition, for the three groups that we would generate N-best lists, we divided each lecture into utterances.

Table 1: Data size.

	lectures	utterances
train1	300	56,622
train2	2,372	-
test1	10	1,293
test2	10	1,156

train1 is a data set for training corrective models. train2 is a data set to make a LM. This LM is used to generate an N-best list for each utterance of train1. To generate N-best lists for test1 and test2, we made another LM using both train1 and train2. The vocabulary size was 100,808.

We generated word 100-best lists for train1 and used them to train corrective models given different (x, y) and λ^{train} . The number of training iterations T was set at 10 for all experiments. For test1 and test2, we generated and reranked the 100-best list of each utterance.

4.2. Investigation of effective hypothesis for generating accurate model

Figures 2 and 3 depict the WERs for test1 and test2, respectively. 'before reranking' denotes the WER of the hypothesis with the top recognition score before applying the corrective model. 'baseline' denotes the results where the corrective models are trained using all the hypotheses in the 100-best list and are used on the constraint $\lambda^{test} = \lambda^{train}$. The others denote the WERs, when each 100-best list is reranked by using the

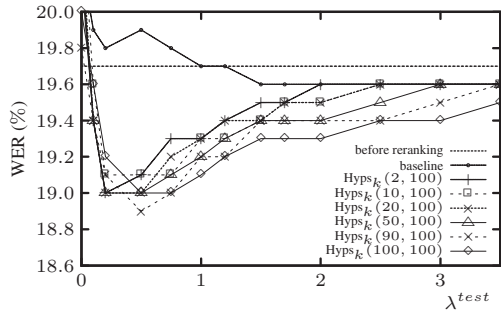


Figure 2: WER for test1 after reranking using the model trained under $\mathbf{Hyps}_k(x, 100)$.

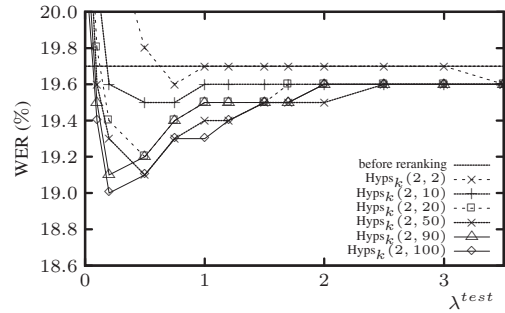


Figure 4: WER for test1 after reranking using the model trained under $\mathbf{Hyps}_k(2, y)$.

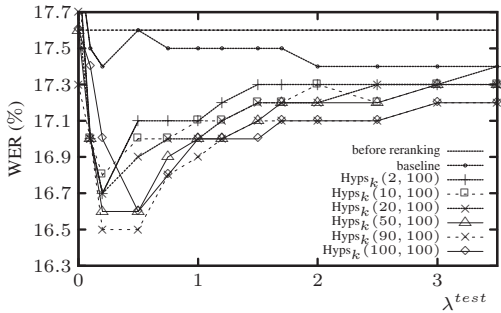


Figure 3: WER for test2 after reranking using the model trained under $\mathbf{Hyps}_k(x, 100)$.

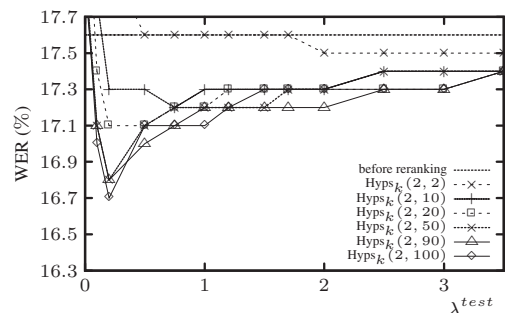


Figure 5: WER for test2 after reranking using the model trained under $\mathbf{Hyps}_k(2, y)$.

corrective model trained given $\lambda^{train} = 0$ and the hypothesis subset $\mathbf{Hyps}_k(x, 100)$, keeping $y = 100$.

The models trained under $\lambda^{train} = 0$ can efficiently reduce word errors, while the effect of baseline is very small in our experimental environment. The model trained under $\lambda^{train} = 0$ and $\mathbf{Hyps}_k(90, 100)$ gives the minimum WERs, 18.9% and 16.5% for test1 and test2, respectively.

The important result is that training using $\mathbf{Hyps}_k(100, 100) = \{w_k^1, w_k^{100}\}$, which constitutes of only two sequences, can generate an accurate corrective model. The amount of word error reduction is largely independent of x . The model trained given a small value of x , that is where low-WER hypotheses can be used to update α , does not lead to any additional word error reduction and furthermore becomes sensitive to λ^{test} .

On the other hand, to confirm the influence of high-WER hypotheses on a performance improvement, we trained some corrective models under different values of y while maintaining $x = 2$ and λ^{train} was 0. Figures 4 and 5 depict the WERs for test1 and test2. The larger the value of y becomes, the more errors are reduced. This means that an accurate corrective model can be constructed by training using high-WER hypotheses.

Note the WERs are 12.4% for test1 and 11.0% for test2 if the oracle hypothesis is correctly extracted from each 100-best list. The corrective model using N-gram counts has a small effect on improvement of speech recognition result.

4.3. Effect of the recognition score in training

Different values of λ^{train} were used to make models using all the hypotheses in the 100-best list, that is $\mathbf{Hyps}_k(2, 100)$. Figures 6 and 7 show the WERs for test1 and test2. The optimiza-

tion of $\lambda^{test} (> 0)$ was done for each model so as to give the minimum WER.

The effect of the corrective model changes greatly around $\lambda^{train} = 0$. Training under $\lambda^{train} < 0$, which is discriminative training using hypotheses with low recognition score, can lead to the generation of an accurate corrective model. Normally λ^{train} is set at a value greater than 0. However, we obtained the result showing the superiority of a model trained intentionally using hypotheses with low speech recognition scores.

This result can be explained by linking it to the WER. Figure 8 shows the relationship with the WER and the order of the recognition score in the N-best list of train1. WER is high when the order is low. We can recognize the low order hypotheses as high-WER hypotheses. Since discriminative training using high-WER hypotheses can generate an accurate model as shown in the previous experiment, a model trained using hypotheses with low-recognition-scores can provide a greater error reduction than a model trained using high-scoring hypotheses.

We compared two corrective models to show which is better as a criterion to make an accurate model between WER and speech recognition score. One is trained using $\mathbf{Hyps}_k(100, 100)$, that is solely the oracle and the highest WER hypothesis. This model is trained given $\lambda^{train} = 0$. The other is trained using the oracle and the hypothesis that has the lowest recognition score in the N-best list. For both, λ^{test} was set at the value that gave minimum WER for the test sets. In the latter case, λ^{train} also was decided in the same manner. The WERs for test1 and test2 after using the models to rerank the N-best lists were 19.0% and 16.6% in the former case, and 19.2% and 16.7% in the latter case, respectively. Training based on the WER is better than that based on the speech recognition score,

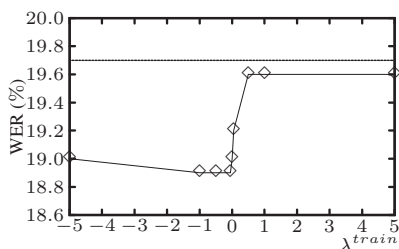


Figure 6: λ^{train} and WER for test1.

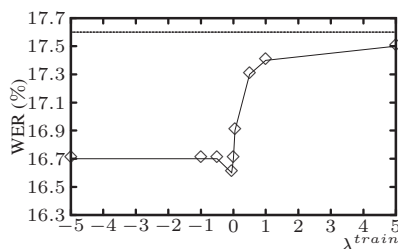


Figure 7: λ^{train} and WER for test2.

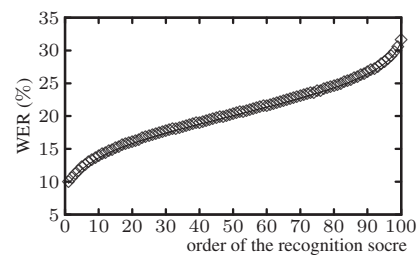


Figure 8: Order of recognition hypotheses and WER for train1.

for the generation of an accurate corrective model.

4.4. Model parameter size

Finally, we compare the number of parameters that have a value other than 0. This parameter is the averaged parameter α_{ave} . Table 2 shows the relationships between the number of parameters and the WER. We were able to construct an accurate and compact corrective model by training using $\mathbf{Hyps}_k(100, 100)$, which is just the oracle and the highest-WER hypothesis. In comparison with such a compact model, ones trained given all the hypotheses and a certain small λ^{train} became extremely large without significant improvements.

Table 2: Model size and WERs.

λ^{train}	$\mathbf{Hyps}_k(x, y)$	# of parameters	WER test1	WER test2
0	(100, 100)	448, 338	19.0	16.6
0	(90, 100)	742, 640	18.9	16.5
-0.05	(2, 100)	1, 024, 573	18.9	16.6
0	(2, 100)	980, 652	19.0	16.7
0.05	(2, 100)	895, 885	19.2	16.9
0.5	(2, 100)	753, 388	19.6	17.3

5. Discussion and Future Work

For the correction of speech recognized text by reranking based on N-gram counts, it is intuitive that the model should be trained to give low weights to the N-gram features from the hypothesis whose recognition score is higher than the score of the oracle hypothesis. This is because it is a direct approach reflecting the idea of the reranking. However, we obtained a result that the training to give low weights to the features from the high-WER hypothesis was better than an approach that took the speech recognition score into consideration. It was possible to generate an accurate model without giving weights to the N-gram features from the hypothesis that has higher recognition score than that of the oracle hypothesis.

We also obtained a result that the effect of the discrimination from low-WER hypotheses was very small. We attribute this result to low discrimination ability of N-gram features. The WER of the oracle hypotheses in test sets is much smaller than the WER after employing a corrective model based on N-gram counts. To reduce the word errors further, we need to discriminate the oracle from similar word sequences robustly for test sets. We believe there is a need for more efficient features than N-grams.

On the other hand, we used the 100-best list in our experiments. We need to undertake a study to determine whether we

can obtain the same results with a word lattice, which includes word sequences with higher WERs. In addition, we need to confirm that the results in this paper are robust for different tasks. The discrimination of the oracle and the high-WER hypotheses can work efficiently for the reduction of word errors. And we were able to construct a small and accurate corrective model by limiting the sequences used for the training. In this case, it suggests that the selection of hypotheses based on the WER can be more effective for the generation of an accurate model than selection based on the speech recognition score.

6. Conclusion

We focused on a method for correcting speech recognition results that is achieved by reranking hypotheses based on N-gram counts. The discrimination of the oracle and the high-WER hypotheses can work efficiently for the reduction of word errors. And we were able to construct a small and accurate corrective model by limiting the sequences used for the training. In this case, it suggests that the selection of hypotheses based on the WER can be more effective for the generation of an accurate model than selection based on the speech recognition score.

7. References

- [1] Michael Collins, "Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms," *Proc. EMNLP*, pp. 1-8, 2002.
- [2] Yoav Freund and Robert E. Schapire, "A Decision-Theoretic Generalization of On-line Learning and An Application to Boosting," *Journal of Computer and System Sciences*, 55(1), pp. 119-1139, 1997.
- [3] Jianfeng Gao, Hao Yu Wei Yuan and Peng Xu, "Minimum Sample Risk Methods for Language Modeling," *Proc. HLTC and EMNLP*, pp. 209-216, 2005.
- [4] Brian Roark, Murat Saraclar and Michael Collins, "Corrective Language Modeling for Large Vocabulary ASR with the Perceptron Algorithm," *Proc. ICASSP*, vol. 1, pp. 749-752, 2004.
- [5] Zhengyu Zhou, Jianfeng Gao, Frank K. Soong and Helen Meng, "A Comparative Study of Discriminative Methods for Reranking LVCSR N-Best Hypotheses in Domain Adaptation and Generalization," *Proc. ICASSP*, vol. 1, pp. 141-144, 2006.
- [6] Kikuo Maekawa, Hanae Koiso, Sadaoki Furui and Hitoshi Isahara, "Spontaneous Speech Corpus of Japanese," *Proc. LREC*, pp. 947-952, 2000.
- [7] Takaaki Hori and Atsushi Nakamura, "Generalized Fast on-the-fly Composition Algorithm for WFST-based Speech Recognition," *Proc. Interspeech 2005*, pp. 284-289, 2005.