INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Improved Deep Duel Model for Rescoring N-best Speech Recognition List Using Backward LSTMLM and Ensemble Encoders

Atsunori Ogawa, Marc Delcroix, Shigeki Karita, Tomohiro Nakatani

NTT Communication Science Laboratories, NTT Corporation, Kyoto, Japan

atsunori.ogawa.gx@hco.ntt.co.jp

Abstract

We have proposed a neural network (NN) model called a deep duel model (DDM) for rescoring N-best speech recognition hypothesis lists. A DDM is composed of a long short-term memory (LSTM)-based encoder followed by a fully-connected linear layer-based binary-class classifier. Given the feature vector sequences of two hypotheses in an N-best list, the DDM encodes the features and selects the hypothesis that has the lower word error rate (WER) based on the output binary-class probabilities. By repeating this one-on-one hypothesis comparison (duel) for each hypothesis pair in the N-best list, we can find the oracle (lowest WER) hypothesis as the survivor of the duels. We showed that the DDM can exploit the score provided by a forward LSTM-based recurrent NN language model (LSTMLM) as an additional feature to accurately select the hypotheses. In this study, we further improve the selection performance by introducing two modifications, i.e. adding the score provided by a backward LSTMLM, which uses succeeding words to predict the current word, and employing ensemble encoders, which have a high feature encoding capability. By combining these two modifications, our DDM achieves an over 10% relative WER reduction from a strong baseline obtained using both the forward and backward LSTMLMs.

Index Terms: speech recognition, N-best rescoring, deep duel model, backward LSTMLM, ensemble encoders

1. Introduction

Based on the recent introduction of state-of-the-art neural network (NN) modeling, the performance of automatic speech recognition (ASR) has been greatly improved [1, 2] and various types of ASR-based applications, including voice search services and smart speakers, have been actively developed. Despite this great progress, in some situations such as performing ASR in noisy environments and/or performing ASR for conversational speech, the ASR accuracy remains at an unsatisfactory level [3–7].

A promising method for improving the ASR accuracy in such severe situations is to exploit multiple speech recognition hypotheses (word sequences), which are represented in the form of, e.g. an N-best list, a lattice, and a confusion network [8]. This is because the 1-best speech recognition hypothesis for an utterance may contain many errors, but a hypothesis that has a significantly lower word error rate (WER) than the 1-best hypothesis can be found in the multiple hypotheses if it is appropriately rescored (reranked). Actually, various types of rescoring methods have been developed and applied to noisy and/or conversational speech recognition [3–6, 9–12].

In this paper, we focus on the *rescoring of N-best speech recognition hypothesis lists*. Currently the most widely-used models for N-best rescoring are recurrent NN language models (RNNLMs) [13], and in particular, long short-term memory (LSTM)-based RNNLMs (LSTMLMs) [14]. An LSTMLM provides scores for each of the hypotheses in a given N-best list. These scores are interpolated with the ASR scores attached to each of the hypotheses, and these interpolated scores are used to rerank the hypotheses in the list. Note that, even though the LSTMLM performs well in N-best rescoring, it was originally developed to predict the current word (the vocabulary size is usually very large) given the preceding word sequence (context) and was not developed to perform N-best rescoring.

We have proposed a new NN model called a *deep duel model* (*DDM*) for N-best rescoring [15] (see Section 2). In contrast to the LSTMLMs, the DDM is developed to perform N-best rescoring. In this sense, the DDM can be categorized as a kind of discriminative LM (DLM) [16, 17], which is trained using N-best lists to find the oracle (lowest WER) hypotheses in the lists. However, in contrast to the sophisticated (but slightly complex) modeling of the DLMs, the DDM is modeled to provide only the minimum functionality to perform N-best rescoring. The functionality is, given two hypotheses in an N-best list, performing their *one-on-one comparison* (*duel*) in terms of the WER and selecting the hypothesis that has a lower WER. By repeating this duel for each hypothesis pair in the N-best list, the oracle hypothesis can be accurately found as the survivor of the duels. We can say that, by focusing on performing the duels of the given hypothesis pairs, the DDM simplifies N-best rescoring task compared with the conventional models, e.g. the LSTMLM that estimates probabilities for all the words in the large vocabulary to perform N-best rescoring.

In this paper, we make two modifications to our DDM. The first modification is applied to the feature to be used (see Section 3.1). The DDM performs a duel by encoding the feature vector sequences of two hypotheses to be compared. In our previous study [15], we greatly reduced the WER by using the score provided by a forward (normal order) LSTMLM as an additional feature. In this study, we further reduce the WER by also using the *backward* (*reverse order*) *LSTMLM* score [5, 6, 10]. The backward LSTMLM exploits succeeding words to predict the current word and has a complementary word prediction ability with the forward LSTMLM. The second modification is applied to the LSTM-based encoder, which is the most important component of the DDM (Section 3.2). We steadily reduce the WER by introducing an ensemble method [18, 19] to the encoder, i.e. *ensemble encoders*, to enhance its feature encoding ability. To stabilize the training of the ensemble encoder-based DDM, we employ a multitask learning (MTL) framework [2, 20]. By combining these two modifications in experiments using a large-scale speech corpus (Section 4), our DDM achieves an over 10% relative WER reduction from a strong baseline N-best rescoring result obtained using both the forward and backward LSTMLMs.

2. Deep duel model

Given two hypotheses in an N-best list, a DDM performs their one-on-one comparison (duel) in terms of the WER. In the following, we describe the structure of the DDM, its efficient training procedure, and its usage in N-best rescoring [15].

2.1. Structure of DDM

Let $W^u = w_1^u, w_2^u, \cdots, w_{L(W^u)}^u$ be the uth hypothesis (word sequence) of length (number of words) $L(W^u)$ in a given N-

http://dx.doi.org/10.21437/Interspeech.2019-1949

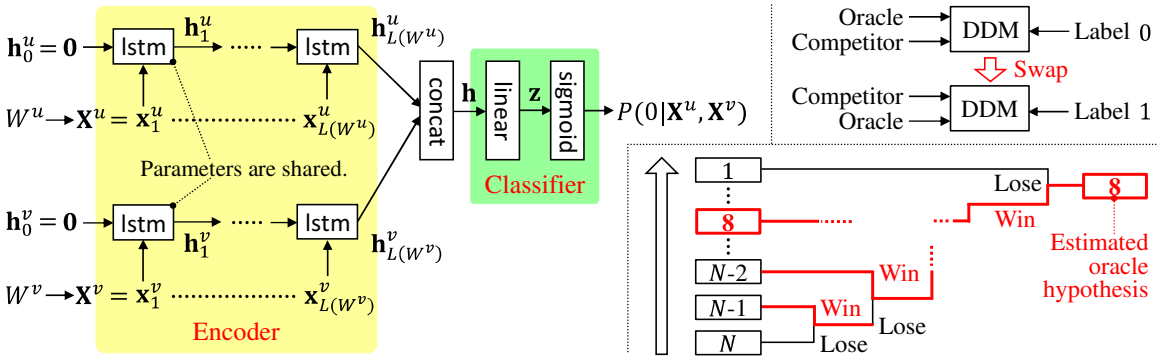

Figure 1: *(Left) Structure of DDM. (Top right) Core part of DDM training. (Bottom right) Duels on N-best list to find oracle hypothesis.*

best list. $\mathbf{A}^u = \mathbf{a}^u_1, \mathbf{a}^u_2, \cdots, \mathbf{a}^u_{L(W^u)}$ is the auxiliary feature vector sequence that corresponds to W^u. The auxiliary feature vector $\mathbf{a}^u_i$ for the word w^u_i is, for example, composed of an acoustic score (log likelihood) and a linguistic score (log probability) provided by an ASR system for the word w^u_i. $\mathbf{X}^u = \mathbf{x}^u_1, \mathbf{x}^u_2, \cdots, \mathbf{x}^u_{L(W^u)}$ is the feature vector sequence that corresponds to W^u. The feature vector $\mathbf{x}^u_i$ of the word w^u_i is obtained as $\mathbf{x}^u_i = \texttt{concat}(\texttt{embed}(w^u_i), \mathbf{a}^u_i)$, where $\texttt{embed}(\cdot)$ denotes the NN-based word embedding operation and $\texttt{concat}(\cdot)$ denotes the vector concatenation operation.

As shown in Fig. 1 (left), the DDM receives $\mathbf{X}^u$ as the upper input and $\mathbf{X}^v$ as the lower input. Here, y is the binary-class symbol, i.e. $y = \{0, 1\}$. $y = 0$ denotes that the WER of W^u (upper input) is not higher than that of W^v (lower input) and, conversely, $y = 1$ denotes that the WER of W^u is higher than that of W^v. The DDM is designed to output $P(0|\mathbf{X}^u, \mathbf{X}^v)$, i.e. the probability of $y = 0$ given $\mathbf{X}^u$ and $\mathbf{X}^v$ as,

$$\begin{cases} P(0|\mathbf{X}^u, \mathbf{X}^v) \geq 0.5 & \text{if WER of } W^u \leq \text{WER of } W^v, \\ P(0|\mathbf{X}^u, \mathbf{X}^v) < 0.5 & \text{otherwise,} \end{cases} \tag{1}$$

where $\sum_y P(y|\mathbf{X}^u, \mathbf{X}^v) = 1$. If the upper inequality is satisfied, the DDM estimates that the WER of W^u is not higher than that of W^v ($y = 0$). Conversely, if the lower inequality is satisfied, the DDM estimates that the WER of W^u is higher than that of W^v ($y = 1$).

The lengths of W^u and W^v, i.e. $L(W^u)$ and $L(W^v)$, may differ. To deal with hypotheses with different lengths, we employ a unidirectional LSTM-based encoder [21–23]. By using an encoder, these two hypotheses can be represented with fixed-length encoded hidden state vectors, and can be fairly compared by using their encoded vectors. Given the feature vector $\mathbf{x}^u_i$ of the current word w^u_i and the previous hidden state vector $\mathbf{h}^u_{i-1}$, an LSTM unit provides the current hidden state vector $\mathbf{h}^u_i$ as,

$$\mathbf{h}^u_i = \texttt{lstm}(\mathbf{x}^u_i, \mathbf{h}^u_{i-1}), \tag{2}$$

where $\texttt{lstm}(\cdot)$ denotes the operation of the LSTM unit ($\mathbf{h}^u_0 = \mathbf{0}$). $\mathbf{h}^u_i$ encodes the feature vector sequence $\mathbf{x}^u_1, \mathbf{x}^u_2, \cdots, \mathbf{x}^u_i$ of the word sequence $w^u_1, w^u_2, \cdots, w^u_i$. By repeating this operation for each feature vector $\mathbf{x}^u_i$ in $\mathbf{X}^u$, we can obtain the hidden state vector $\mathbf{h}^u_{L(W^u)}$ that encodes $\mathbf{X}^u$. By also applying this operation to $\mathbf{X}^v$, we can obtain the encoded hidden state vector $\mathbf{h}^v_{L(W^v)}$ of $\mathbf{X}^v$. Note that the parameters are shared between the LSTMs used for encoding $\mathbf{X}^u$ and $\mathbf{X}^v$, i.e. the DDM has only one encoder. Then, we concatenate $\mathbf{h}^u_{L(W^u)}$ and $\mathbf{h}^v_{L(W^v)}$ to obtain the concatenated encoded hidden state vector $\mathbf{h}$ as,

$$\mathbf{h} = \texttt{concat}(\mathbf{h}^u_{L(W^u)}, \mathbf{h}^v_{L(W^v)}). \tag{3}$$

A classifier follows the encoder. It is composed of a one-layer fully-connected linear layer followed by the sigmoid activation function. The concatenated encoded hidden state vector $\mathbf{h}$ provided by the encoder is input into the classifier, and we obtain the probability $P(0|\mathbf{X}^u, \mathbf{X}^v)$ as,

$$\begin{aligned} \mathbf{z} &= \texttt{linear}(\mathbf{h}), \\ P(0|\mathbf{X}^u, \mathbf{X}^v) &= \texttt{sigmoid}(\mathbf{z}), \end{aligned} \tag{4}$$

where $\texttt{linear}(\cdot)$ and $\texttt{sigmoid}(\cdot)$ denote the operations of the linear layer and the sigmoid activation function, respectively.

2.2. Training of DDM

Figure 1 (top right) shows the core part of the DDM training procedure. The main purpose of N-best rescoring is to find the oracle hypothesis from a given N-best list, and thus we train the DDM by performing duels between the oracle hypothesis and competitor hypotheses (the oracle hypothesis has to beat all the competitor hypotheses).

We form a pair consisting of the oracle hypothesis and a competitor hypothesis from a given N-best list and feed it to the DDM with the corresponding binary-class teacher label, i.e. 0 or 1. As shown in Fig. 1 (top right), we first feed the oracle hypothesis to the DDM as the upper input and the competitor hypothesis as the lower input. In this case, the DDM should output the probability of the symbol 0 as $P(0|\mathbf{X}^u, \mathbf{X}^v) = 1$, and thus we give the teacher label 1. We next swap the oracle and competitor hypotheses and feed them to the DDM with the teacher label 0.

If we repeat this core training procedure for all of the competitor hypotheses in the N-best (e.g. 100-best) list, for all of the N-best lists (utterances) in the training data, and for a certain number of epochs, the training cost becomes very high. To reduce the training cost, we select the competitor hypotheses. We preferentially select the following four hypotheses: 1) the 1-best (highest ASR score) hypothesis, 2) the second lowest WER hypothesis, 3) the lowest rank (lowest ASR score) hypothesis, and 4) the highest WER hypothesis. Hypotheses 1) and 2) are selected as strong competitors that are difficult to beat. Conversely, hypotheses 3) and 4) are selected as the weak competitors that are easy to beat. To increase the variety of the duels, in addition to the above four hypotheses, we also select hypotheses from the N-best list with an equal rank interval (the total number of competitor hypotheses is, e.g. 20).

2.3. N-best rescoring using DDM

In the evaluation phase, we use the trained DDM to perform N-best rescoring (find the oracle hypothesis) for a given N-best list as shown in Fig. 1 (bottom right). As with the first step of the bubble sort [24], we repeat duels from the bottom to the top of the list. Then, we select the survivor as the estimated oracle hypothesis. We start the duel from the bottom of the list because the oracle hypothesis tends to exist at a higher rank in the list, and so higher rank hypotheses should be seeded (evaluated at later stages) in this step-ladder tournament.

Note that, as with N-best rescoring using the conventional N-best rescoring models, we can use ASR scores attached to each of the hypotheses in the given N-best list. In this case, we combine the binary-class symbol log-probabilities provided by the DDM with the corresponding ASR scores (log likelihoods). We can obtain the scores of W^u (upper input) and W^v (lower input), i.e. $s(W^u)$ and $s(W^v)$, used for the duel as,

$$\begin{cases} s(W^u) = (1-\lambda)\log p(W^u|\mathbf{O}) + \lambda \log P(0|\mathbf{X}^u, \mathbf{X}^v), \\ s(W^v) = (1-\lambda)\log p(W^v|\mathbf{O}) + \lambda \log P(1|\mathbf{X}^u, \mathbf{X}^v), \end{cases} \tag{5}$$

where $\mathbf{O}$ is the acoustic feature vector sequence of the input utterance, $\log p(W^r|\mathbf{O})$ is the ASR score provided for the rth hypothesis W^r, λ is the weight for the DDM scores ($0 \leq \lambda \leq 1$), and $P(1|\mathbf{X}^u, \mathbf{X}^v)$ equals $1 - P(0|\mathbf{X}^u, \mathbf{X}^v)$ (see Section 2.1).

3. Modifications to DDM

We describe modifications applied to the features and the encoder of the DDM.

3.1. Backward LSTMLM

The auxiliary feature vector $\mathbf{a}_i^u$ for the word w_i^u is composed of the features provided by an ASR system, e.g. an acoustic model score, an n-gram LM score, and their weighted sum, i.e. an ASR score [25]. In addition to these basic auxiliary features, in our previous study [15], we greatly reduced the WER by using a forward LSTMLM score as an additional auxiliary feature. This WER reduction comes from the ability of the forward LSTMLM to capture the long preceding word sequence (context) information to predict the current word probability (the n-gram LM captures only a few words of context).

Various types of RNNLMs have been proposed in addition to the forward RNNLM, e.g. backward RNNLMs [5, 6, 10, 26], bidirectional RNNLMs [27–29], mixture-of-softmax (MoS) RNNLMs [30], and whole sentence RNNLMs [31], and their effectiveness has been reported. Among these candidates, in this study, we select the backward LSTMLM that exploits the succeeding word sequence to predict the current word. It is reasonable to use the backward LSTMLM, which captures the long succeeding context to predict the current word, in addition to the forward LSTMLM, as with previous studies [5, 6, 10], since we can expect these two LSTMLMs to work complementarily to predict the current word. We use the backward LSTMLM score as the second additional auxiliary feature in the DDM.

3.2. Ensemble encoders

As can be easily expected from Fig. 1 (left), the most important component of the DDM is the LSTM-based encoder. In preliminary experiments, we tried to modify the encoder by introducing, e.g. a bidirectional LSTM, a multiple-layer LSTM, or an LSTM with a large number of nodes (the word embedding size is equal to the number of LSTM nodes). However, we obtained almost no WER reduction with these modifications.

To steadily reduce the WER, we introduced an ensemble method [18, 19] to the encoder. Ensembling is a promising method to improve the performance of machine learning models and is also widely used in the NN modeling of ASR components, e.g. acoustic modeling [32–34] and language modeling [35–37]. Figure 2 shows a DDM structure with the proposed ensemble encoders. We first tried a simple parallelization of the encoders (the red path in Fig. 2). However, we obtained a limited WER reduction with this modification. We confirmed that, with this simple parallelization, only one encoder (`lstm1` or `lstm2` in Fig. 2) contributes to feature encoding. To force all the encoders to contribute to encoding, we employ an MTL framework [2, 20]. As shown in Fig. 2, we attach a classifier for each of the encoders and force all the sub DDMs to output the probability of the symbol 0. With this modification, all the encoders contribute to feature encoding and we can obtain a WER reduction using the output from the main DDM.

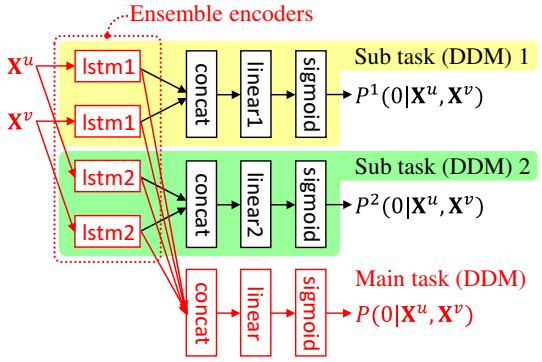

Figure 2: *Ensemble (two) encoder DDM with multitask learning framework.*

Note that, in this study, all the encoders have the same structure (they also share the word embedding layer) and their parameters are initialized randomly, i.e. the encoders only differ in terms of their initial parameters [19, 36]. When training an ensemble encoder DDM with the MTL framework, we do not introduce any weighting to any of the sub tasks or the main task (all the tasks have the same weight, i.e. 1.0).

4. Experiments

We conducted experiments using the corpus of spontaneous Japanese (CSJ) [38], which is a large-scale lecture speech corpus. We performed all the NN modeling using Chainer [39]. We performed 200-best rescoring in our previous study [15] but performed 100-best rescoring in this study (since many studies perform 100-best rescoring). Other than the N of the N-best lists, the experimental settings were the same as those in our previous study and [15] provides these settings in detail.

4.1. Experimental settings

We trained a convolutional NN (CNN)-based acoustic model, a trigram LM, and the forward and backward LSTMLMs listed in Table 2 (models 2 and 3) using 250 hours of speech data and their transcriptions. The vocabulary size of these LMs was 31k, and the word embedding size of the LSTMLMs was 650. The structure of the LSTMLMs is shown in Table 1. Using the CNN acoustic model and the trigram LM, we performed ASR with a weighted finite state transducer-based one-pass decoder [40] for 271/6/6 hours of the DDM training/development/evaluation speech data and obtained the 100-best ASR hypothesis lists for these data.

Using the obtained 100-best lists, we trained the eight DDMs listed in Table 2 (models 5 to 12) with the procedure described in Section 2.2. We selected 20 competitor hypotheses for each of the oracle hypotheses. The word embedding size was 100, and the ASR-based auxiliary feature vector size was 17 [15, 25] (models 5, 9, 10, and 11). We used the forward and/or backward LSTMLM scores as the 18th and 19th auxiliary features (models 6, 7, 8, and 12). The structures of the single-encoder DDM and the eight-encoder DDM (models 5 and 11) are shown in Table 1. The total numbers of parameters of the DDMs are very small compared with those of the LSTMLMs. This difference mainly comes from the differences in the word embedding size and the softmax/sigmoid output size. In contrast, those of the two DDMs are not very different. This is because all the encoders in the eight-encoder DDM share the word embedding layer (see Section 3.2).

In the evaluation phase, we performed 100-best rescoring using the LSTMLMs and the DDMs listed in Table 2 by taking the weighted sum of their scores and the ASR scores (Section 2.3). For each model, the weight λ value that provided the

Table 1: *Structures of the forward/backward LSTMLM, single-encoder DDM, and eight-encoder DDM. In these models, the number of nodes is the same for the word embedding layers, unidirectional LSTM units, and linear layers.*

	LSTMLM	DDM	8-enc. DDM
# nodes	650	100	100 (×8)
# LSTM layers	2	1	1 (×8)
# linear layers	1	1	1 (×9)
Output softmax/sigmoid size	31k	1	1 (×9)
# total parameters	47M	3.2M	4.0M

Table 2: *100-best rescoring results in WER [%] for the development and evaluation data. WERs of the ASR 1-best (trigram) and oracle hypotheses are also shown.*

No.	Model	Dev	Eval
1.	ASR 1-best (trigram)	18.1	14.8
2.	Forward LSTMLM	17.4	14.2
3.	Backward LSTMLM	17.3	14.1
4.	Forward and backward LSTMLMs	17.3	14.1
5.	Single-encoder DDM	16.4	13.7
6.	5 with forward LSTMLM score	15.6	13.0
7.	5 with backward LSTMLM score	15.7	13.0
8.	5 with fwd & bwd LSTMLM scores	15.4	12.8
9.	Two-encoder DDM	16.2	13.5
10.	Four-encoder DDM	16.1	13.5
11.	Eight-encoder DDM	16.1	13.4
12.	11 with fwd & bwd LSTMLM scores	**15.2**	**12.6**
13.	Oracle	11.6	9.7

lowest WER for the development data was selected as the optimal value, and this value was used for 100-best rescoring on the evaluation data. When we performed 100-best rescoring using both the forward and backward LSTMLMs (model 4), we merged their scores word-by-word in the log probability domain with the equal weight (i.e. 0.5) as with [5, 6, 10, 26].

4.2. Experimental results

Table 2 shows the 100-best rescoring results for the development and evaluation data. We can confirm that the LSTMLMs (models 2, 3, and 4) steadily reduce the WER from the ASR 1-best (trigram) baseline and the single-encoder DDM (model 5) provides additional WER reduction from the good LSTMLM results. This WER reduction comes from the simplification of N-best rescoring task with the DDM (Section 1). The DDM focuses on performing the duels of the given hypothesis pairs (exploiting their auxiliary features) while the LSTMLMs estimate the probabilities of all the words in the 31k vocabulary.

We can also confirm that the DDMs using the LSTMLM scores (models 6, 7, and 8) provide further large WER reductions. In contrast to the results reported in [5, 26], we cannot obtain a WER reduction by using both the forward and backward LSTMLMs (model 4) from when using only one of them i.e. the backward LSTMLM (model 3). In contrast, by using both their scores in the DDM (model 8), we can steadily reduce the WER from when using only one of their scores (model 6 or 7). This result confirms that the DDM can effectively exploit complementary features and encourages us to use other types of LSTMLM scores in addition [27–31].

We can confirm that the WER can be steadily reduced by introducing the ensemble encoders (models 9, 10, and 11). More encoders provide a larger WER reduction. Figure 3 shows the

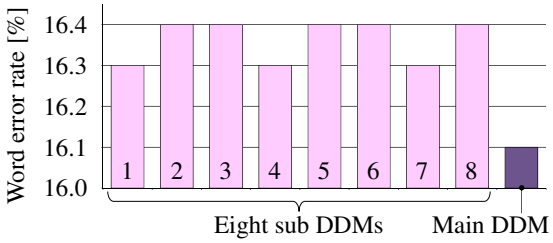

Figure 3: *WERs for the development data obtained with the eight sub DDMs and the main DDM in model 11.*

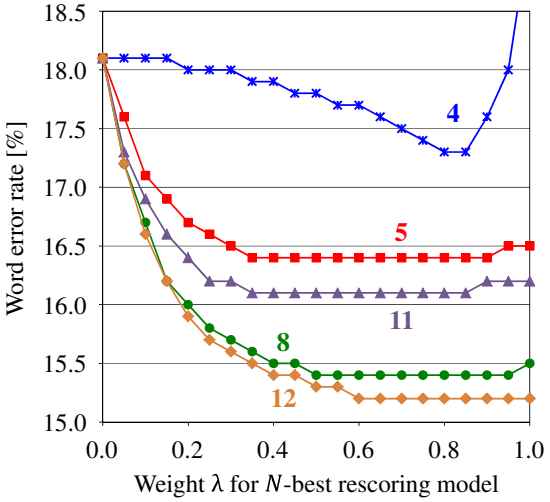

Figure 4: *WERs for the development data obtained with N-best rescoring models 4, 5, 8, 11, and 12 in Table 2 as a function of weight λ for the model scores.*

WERs for the development data obtained with the eight sub DDMs and the main DDM in model 11 (see Fig. 2). The WERs of the sub DDMs are the same as (or slightly lower than) that of the single-encoder DDM (model 5). We can clearly confirm the effectiveness of the ensemble encoders. Finally, using both the forward and backward LSTMLM scores in the eight-encoder DDM (model 12), we can obtain an over 10% relative WER reduction compared with the result obtained using both the forward and backward LSTMLMs (model 4).

Figure 4 shows the WERs for the development data obtained with models 4, 5, 8, 11, and 12 as a function of the weight λ for these models. We can confirm that the optimal value range of λ for the LSTMLM (model 4) is very narrow (around 0.8). In contrast, those for the DDMs (models 5, 8, 11, and 12) are very wide (around 0.5 to 1.0). This robustness against the weight λ comes from the fact that the DDMs use the ASR score as the auxiliary feature (Section 3.1) and this is also an advantage of the DDMs over the LSTMLMs (λ can be roughly adjusted). As with Table 2, from Fig. 4, we can also clearly confirm the effect of using both the forward and backward LSTMLM scores (model 5 $\rightarrow$ 8), that of introducing the ensemble encoders (model 5 $\rightarrow$ 11), and that of both the above two complementary modifications (model 5 $\rightarrow$ 12).

5. Conclusion and future work

We have improved our DDM for rescoring N-best speech recognition hypothesis lists by using the backward LSTMLM score and ensemble encoders. Future work will include the use of other types of LSTMLM scores [27–31] and the use of a mixture-of-experts framework [36, 37]. We also plan to compare our DDM with DLMs [16, 17] and apply it to rescoring N-best machine translation hypothesis lists [41, 42].

6. References

[1] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[2] D. Yu and L. Deng, *Automatic speech recognition: A deep learning approach*. Springer-Verlag London, 2015.

[3] G. Saon, H.-K. J. Kuo, S. Rennie, and M. Picheny, "The IBM 2015 English conversational telephone speech recognition system," in *Proc. Interspeech*, 2015, pp. 3140–3144.

[4] G. Saon, T. Sercu, S. Rennie, and H.-K. J. Kuo, "The IBM 2016 English conversational telephone speech recognition system," in *Proc. Interspeech*, 2016, pp. 7–11.

[5] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "Achieving human parity in conversational speech recognition," *arXiv:1610.05256v2 [cs.CL]*.

[6] W. Xiong, L. Wu, F. Alleva, J. Droppo, X. Huang, and A. Stolcke, "The Microsoft 2017 conversational speech recognition system," in *Proc. ICASSP*, 2018, pp. 5934–5938.

[7] J. Barker, S. Watanabe, E. Vincent, and J. Trmal, "The fifth 'CHiME' speech separation and recognition challenge: Dataset, task and baselines," in *Proc. Interspeech*, 2018, pp. 1561–1565.

[8] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: Word error minimization and other applications of confusion networks," *Computer Speech and Language*, vol. 14, no. 4, pp. 373–400, Oct. 2000.

[9] J. Du, T. Gao, L. Sun, F. Ma, Y. Fang, D.-Y. Liu, Q. Zhang, X. Zhang, H.-K. Wang, J. Pan, J.-Q. Gao, C.-H. Lee, and J.-D. Chen, "The USTC-iFlytek systems for CHiME-5 challenge," in *Proc. of The 5th Intl. Workshop on Speech Processing in Everyday Environments (CHiME 2018)*, 2018.

[10] N. Kanda, R. Ikeshita, S. Horiguchi, Y. Fujita, K. Nagamatsu, X. Wang, V. Manohar, N. E. Y. Soplin, M. Maciejewski, S.-J. Chen, A. S. Subramanian, R. Li, Z. Wang, J. Naradowsky, L. P. Garcia-Perera, and G. Sell, "The Hitachi/JHU CHiME-5 system: Advances in speech recognition for everyday home environments using multiple microphone arrays," in *Proc. of The 5th Intl. Workshop on Speech Processing in Everyday Environments (CHiME 2018)*, 2018.

[11] Z. Zhao, J. Wu, and L. Xie, "The NWPU system for CHiME-5 challenge," in *Proc. of The 5th Intl. Workshop on Speech Processing in Everyday Environments (CHiME 2018)*, 2018.

[12] C. Li and T. Wang, "The ZTSpeech system for CHiME-5 challenge: A far-field speech recognition system with front-end and robust back-end," in *Proc. of The 5th Intl. Workshop on Speech Processing in Everyday Environments (CHiME 2018)*, 2018.

[13] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. Interspeech*, 2010, pp. 1045–1048.

[14] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM neural networks for language modeling," in *Proc. Interspeech*, 2012.

[15] A. Ogawa, M. Delcroix, S. Karita, and T. Nakatani, "Rescoring N-best speech recognition list based on one-on-one hypothesis comparison using encoder-classifier model," in *Proc. ICASSP*, 2018, pp. 6099–6103.

[16] B. Roark, M. Saraclar, and M. Collins, "Discriminative n-gram language modeling," *Computer Speech and Language*, vol. 21, no. 2, pp. 373–392, Apr. 2007.

[17] T. Oba, T. Hori, A. Nakamura, and A. Ito, "Round-robin duel discriminative language models," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 4, pp. 1244–1255, May 2012.

[18] T. G. Dietterich, "Ensemble methods in machine learning," in *Proc. MCS*, 2000, pp. 1–15.

[19] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*, 2016.

[20] R. Caruana, "Multitask learning: a knowledge-based source of inductive bias," in *Proc. ICML*. IMLS, 1993, pp. 41–48.

[21] I. Sutskever, O. Vinyals, and Q. Le, "Sequence to sequence learning with neural networks," in *Proc. NIPS*, 2014, pp. 3104–3112.

[22] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, "End-to-end continuous speech recognition using attention-based recurrent NN: First results," in *Proc. Deep Learning and Representation Learning Workshop: NIPS 2014*, 2014.

[23] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. ICASSP*, 2016, pp. 4960–4964.

[24] D. Knuth, *The Art of Computer Programming, Volume 3: Sorting and Searching, Second Edition*. Addison-Wesley, 1998.

[25] A. Ogawa and T. Hori, "Error detection and accuracy estimation in automatic speech recognition using deep bidirectional recurrent neural networks," *Speech Communication*, vol. 89, pp. 70–83, May 2017.

[26] Y. Shi, M. Larson, P. Wiggers, and C. M. Jonker, "Exploiting the succeeding words in recurrent neural network language models," in *Proc. Interspeech*, 2013, pp. 632–636.

[27] E. Arisoy, A. Sethy, B. Ramabhadran, and S. Chen, "Bidirectional recurrent neural network language model for automatic speech recognition," in *Proc. ICASSP*, 2015, pp. 5421–5425.

[28] T. He, Y. Zhang, J. Droppo, and K. Yu, "On training bi-directional neural network language model with noise contrastive estimation," in *Proc. ISCSLP*, 2016.

[29] X. Chen, A. Ragni, X. Liu, and M. J. F. Gales, "Investigating bidirectional recurrent neural network language models for speech recognition," in *Proc. Interspeech*, 2017, pp. 269–273.

[30] Z. Yang, Z. Dai, R. Salakhutdinov, and W. W. Cohen, "Breaking the softmax bottleneck: A high-rank RNN language model," in *Proc. ICLR*, 2018.

[31] Y. Huang, A. Sethy, K. Audhkhasi, and B. Ramabhadran, "Whole sentence neural language models," in *Proc. ICASSP*, 2018, pp. 6089–6093.

[32] X. Chen and Y. Zhao, "Building acoustic model ensembles by data sampling with enhanced trainings and features," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 3, pp. 498–507, Mar. 2013.

[33] L. Deng and J. C. Platt, "Ensemble deep learning for speech recognition," in *Proc. Interspeech*, 2014, pp. 1915–1919.

[34] J. Du, Q. Wang, Y.-H. Tu, X. Bao, L.-R. Dai, and C.-H. Lee, "An information fusion approach to recognizing microphone array speech in the CHiME-3 challenge based on a deep learning framework," in *Proc. ASRU*, 2015, pp. 430–435.

[35] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *arXiv:1409.2329 [cs.NE]*.

[36] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," in *Proc. ICLR*, 2017.

[37] W. Wang, Z. Gan, W. Wang, D. Shen, J. Huang, W. Ping, S. Satheesh, and L. Carin, "Topic compositional neural language model," in *Proc. AISTATS, PMLR*, vol. 84.

[38] K. Maekawa, "Corpus of spontaneous Japanese: its design and evaluation," in *Proc. Workshop on Spontaneous Speech Processing and Recognition (SSPR)*, 2003, pp. 7–12.

[39] S. Tokui, K. Oono, S. Hido, and J. Clayton, "Chainer: a next-generation open source framework for deep learning," in *Proc. Workshop on Machine Learning Systems at NIPS*, 2015.

[40] T. Hori, C. Hori, Y. Minami, and A. Nakamura, "Efficient WFST-based one-pass decoding with on-the-fly hypothesis rescoring in extremely large vocabulary continuous speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1352–1365, May 2007.

[41] G. Neubig, M. Morishita, and S. Nakamura, "Neural reranking improves subjective quality of machine translation: NAIST at WAT2015," in *Proc. WAT*, 2015, pp. 35–41.

[42] J. Niehues, E. Cho, and A. W. T.-L. Ha, "Analyzing neural MT search and model performance," in *Proc. The 1st Workshop on Neural Machine Translation*, 2017, pp. 11–17.