Multimodal N-best List Rescoring with Weakly Supervised Pre-training in Hybrid Speech Recognition

Yuanfeng Song[*†], Xiaoling Huang[†], Xuefang Zhao[†], Di Jiang[†], Raymond Chi-Wing Wong[*]
[*]The Hong Kong University of Science and Technology, Hong Kong, China
[†]AI Group, Webank Co., Ltd., Shenzhen, China
[*]{songyf, raywong}@cse.ust.hk [†]{summerzhao, smallhuang, jiangdi}@webank.com

***Abstract*—N-best list rescoring, an essential step in hybrid automatic speech recognition (ASR), aims to re-evaluate the N-best hypothesis list decoded by the acoustic model (AM) and language model (LM), and selects the top-ranked hypotheses as the final ASR results. This paper explores the performance of neural rescoring models in scenarios where large-scale relevance training signals are not available. We propose a weakly supervised neural rescoring method, `WSNeuRescore`, where a listwise multimodal neural rescoring model is pre-trained using labels automatically obtained without human annotators. Specifically, we employ the output of an unsupervised rescoring model, the weighted linear combination of the AM score and the LM score, as a weak supervision signal to pre-train the neural rescoring model. Our experimental evaluations on a public dataset validate that the pre-trained rescoring model based on weakly supervised data leads to an impressive performance. In the extreme scenario without any high-quality labeled data, it achieves up to an 11.90% WER reduction and a 15.56% NDCG@10 improvement over the baseline method in Kaldi, a well-known open-source toolkit in the ASR community.**

***Index Terms*—Multimodal learning, weak supervision, N-best rescoring, automatic speech recognition, learning-to-rank**

I. Introduction

Nowadays, speech-driven applications, such as voice-based search engines, speech-driven querying systems [1], and voice-based chatbots and AI-powered virtual assistants (e.g., Siri, Xiaoice, and Cortana), have become mainstream in the market due to the popularity of mobile phones and smart devices. A reliable and accurate automatic speech recognition (ASR) system is the premise of the success of these speech-driven applications. Existing ASR technologies can be roughly divided into two categories, that is, the hybrid system and the end-to-end system. Despite the popularity of end-to-end ASR models in the research community, the hybrid ones still dominate the ASR industry due to their flexibility and modularization [2]. Hence, advancing hybrid ASR systems still draws great attention from both the research and the industrial communities [3].

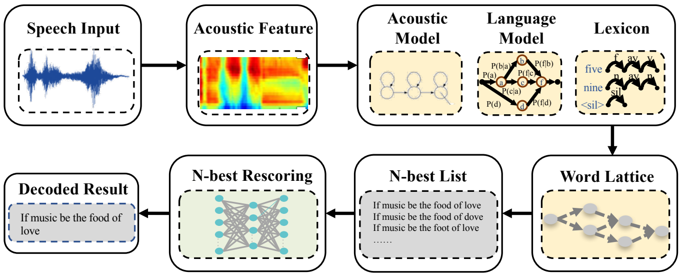

Fig. 1. The pipeline of a typical hybrid ASR system, including the N-best rescoring procedure, is essentially analogous to the working mechanism of modern IR systems.

This work views the hybrid ASR pipeline from the information retrieval (IR) perspective [4], [5] and aims to boost the performance of ASR systems with methods inspired by advanced Learning-to-Rank (LTR) techniques [6]. As shown in Fig. 1, a typical hybrid ASR system usually consists of two main components: an acoustic model (AM) and a basic language model (LM), the first of which converts the speech signals into phone sequences, and the latter of which evaluates the probability of generated word sequences. Finally, an N-best list rescoring step re-evaluates the N-best hypothesis list decoded by the AM and the basic LM, and selects the top-ranked hypotheses as the final decoding results. Since modern IR systems usually enjoy a similar working mechanism, namely, first retrieves a substantial amount of relevant candidates, then re-ranks them, and finally outputs the top-N list, we argue that it would be promising to integrate techniques across the IR and ASR communities.

Among all the steps described in the hybrid ASR pipeline, N-best list rescoring is the most important since it greatly affects the final recognition accuracy, because more sophisticated and effective LMs can only be used in this period. Besides ASR, the N-best list rescoring is also essential and fundamental in many other natural language processing (NLP) tasks such as machine translation and dialogue system [7]. As such, we mainly focus on exploring more effective rescoring mechanisms to advance the performance of the hybrid ASR systems in this paper. However, the proposed techniques would potentially benefit a broader range of fields besides ASR.

The existing N-best rescoring methods, such as the one used in open-source ASR platform Kaldi [8], typically work in an unsupervised style, specifically, a weighted linear addition of the AM score and the LM score. With the dominating performance of the deep neural networks (DNNs) in NLP tasks, researchers have also applied DNN-based rescoring methods in ASR. For example, recurrent neural network LMs (RNNLMs) have been used as the LM to rescore the N-best list and show promising results [9]. Neural Speech-to-Text LM (NS2TLM) [10] captures the information hidden in the speech signals as well as the historical word sequence to decode the next word. These DNN-based rescoring methods show much better performance compared with the traditional unsupervised approach. However, they usually tackle the rescoring problem in a two-step style, that is, first predicting a sophisticated LM score and AM score, and then formulating the ranking orders based on the weighted addition of these two scores. What is more, all the aforementioned rescoring methods assume the availability of very large-scale labeled training datasets, which is usually not feasible, especially for minority languages, dialects, and domain-dependent ASRs.

To alleviate the above-mentioned problems, this paper works towards a more effective neural rescoring model in the scenario where large-scale relevance training signals are not available. We propose a Weak Supervision Neural Rescoring framework, `WSNeuRescore`, which employs a two-step strategy: pre-training using weak supervision and fine-tuning with limited labeled instances. Under the `WSNeuRescore` framework, we further design a Multimodal (i.e., speech and text) neural rescoring network that adopts a listwise approach to directly optimize the order of the N-best list. In particular, the output of an unsupervised rescoring model, the weighted linear combination of the AM score and LM score, is used as a weak supervision signal to pre-train this Multimodal neural rescoring network. Then an optional fine-tuning step is incorporated by `WSNeuRescore` to further boost the performance of the pre-trained rescoring model if limited labeled data is available. Our experimental evaluations on a public dataset validate that the Multimodal neural rescoring network, together with the weakly supervised pre-training mechanism, leads to impressive performance improvement over the baseline model in Kaldi as well as other DNN-based counterparts. We hope this work inspires further work across the IR and ASR communities.

To summarize, the main contributions of the paper are as follows:

- To the best of our knowledge, this is the first work that studies the pre-training mechanism with weak supervision for the N-best rescoring problem in ASR. We hope this work will benefit ASR as well as other NLP tasks such as machine translation.
- We propose a novel weak supervision framework `WSNeuRescore`, together with a Multimodal neural rescoring network, dedicated to hybrid ASR systems. `WSNeuRescore` tackles the rescoring problem from an IR perspective and incorporates the first Multimodal neural rescoring network in the field of ASR.
- We conduct extensive experiments on a public dataset, and experimental results show that `WSNeuRescore` can significantly outperform not only traditional rescoring methods but also recent DNN-based counterparts, by up to an 11.90% word error rate (WER) reduction and a 20.85% normalized discount cumulative gain (NDCG)@10 improvement, respectively.

II. Related work

N-best List Rescoring in ASR The N-best list is normally ranked by the score combined by an AM and an LM. The n-gram LMs are the most commonly used ones [11], however, they suffer the problem of the incapability in modeling language context such as long-range dependencies. Discriminative LMs (DLMs) [12], [13] improve the n-gram LM by incorporating more features (e.g., the ASR errors during the model training period) to construct a discriminative model to classify the positive instances from the negative ones. LM adaptation (LMA) methods using cache [14] and topic modeling [15] have also been developed for N-best rescoring, which modify the LMs with the first-pass decoding result, and reduce the mismatch between the training domain and the prediction domain. Currently, the most popular models for N-best rescoring are DNN-based LMs such as RNNLMs [16], [17]. Compared with traditional approaches, they usually deliver better performance since deep neural models have the superior capability in capturing both short and long-range dependencies between words in human language. EC-Model [18] is another DNN-based classifier with the minimum necessary functionality to rescore the N-best lists. Neural Speech-to-Text LM (NS2TLM) [10] extends RNNLMs by making use of the encoded information from the acoustic feature sequence together with the historical textual information to decode the next word. L2RS [4] formalizes the N-best list rescoring problem as a Learning-to-Score problem, and then various features extracted using advanced NLP models (e.g., BERT [19] embeddings) are used to build an LTR model.

Weakly Supervised Learning in IR The data scarcity problem is a perpetual topic in machine learning, especially for DNN-based models since they usually require a large amount of labeled data. Weakly supervised learning aims to build predictive models by learning with weak supervision or weak signals where large-scale accurate labels are unavailable. According to the survey [20], there are three kinds of weak supervision methods: incomplete supervision, inexact supervision, and inaccurate supervision. Several attempts at weakly supervised training of neural ranking models have been made in the IR domain. A ranking model with weak supervision was proposed in [21], where the weak signals for training a more sophisticated LTR model are gathered from a basic BM25 model. A selective weak supervision strategy was proposed in [22], which employs a reinforcement learning-based weak supervision strategy to select the weak signals. Hamed *et al.* give theoretical analysis and show insight into weak supervision for LTR [23]. Dany *et al.* further design methods to reduce the amount of training data for weak supervision [24]. When it comes to the ASR area, studies such as [25], [26] also explore the possibility of training an ASR model (not a rescoring model) with a weak supervision mechanism.

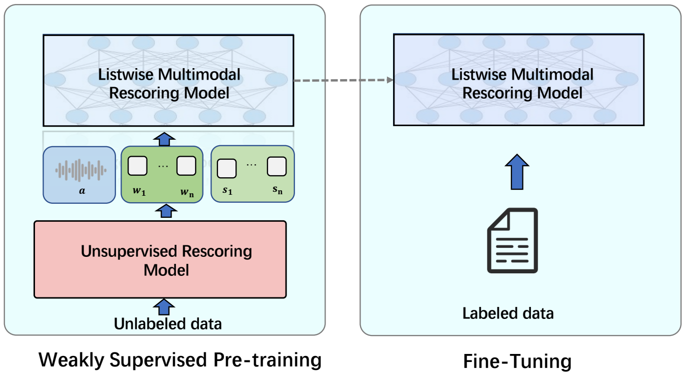
Listwise Multimodal Rescoring Model
a
w1
wn
s1
sn
Unsupervised Rescoring Model
Unlabeled data
Weakly Supervised Pre-training
Listwise Multimodal Rescoring Model
Labeled data
Fine-Tuning

Fig. 2. Weakly Supervised Pre-training and Fine-Tuning for `WSNeuRescore`

III. Methodology

While DNN-based rescoring models have taken off in the ASR area, these models are usually required to be trained with massive labeled training data. However, when it comes to minority languages and dialects, such as Cantonese[1] and Wu Chinese[2], large-scale annotations are usually costly to obtain or unavailable. Hence, unsupervised learning or weak supervision is believed to be a promising solution. Inspired by recent progress in weakly supervised learning [21], we make full use of existing unsupervised rescoring methods to create a weakly annotated set of training data and use these data with weak signals to pre-train the neural rescoring model. Then pre-trained neural rescoring is further fine-tuned with limited in-domain labeled data, as shown in Fig. 2. In the following discussion, we will go through the details of `WSNeuRescore` from three aspects, weak supervision signal generation, listwise multimodal rescoring model, and domain adaptation and fine-tuning.

A. Weak Supervision Signal Generation

For an ASR system, we use $\mathbf{a}$ to represent the acoustic input and $\mathbf{w}^*$ as the corresponding textual output. The system decodes the acoustic input $\mathbf{a}$ with an AM and LM, and then gets the N-best candidates, denoted as $\mathbf{W} = \{\mathbf{w}_1, \cdots, \mathbf{w}_N\}$. The N-best rescoring model ranks these candidates according to their correctness and selects the best candidate from the N-best list, which is represented as:

$$\mathbf{w}^* = \arg\max \mathcal{S}(\phi(\mathbf{a}, \mathbf{W}); \Theta), \tag{1}$$

where $\phi(\cdot)$ is the feature representation of pair $(\mathbf{a}, \mathbf{w})$ and $\mathcal{S}(\cdot; \Theta)$ is the neural rescoring model with network parameters Θ.

During the pre-training period, the training set $\mathcal{D}$ is composed of instances in forms of $(\mathbf{a}, \mathbf{w}, s_{\mathbf{a},\mathbf{w}})$, where $s_{\mathbf{a},\mathbf{w}}$ is a relevance score between $\mathbf{a}$ and $\mathbf{w}$. We use $\mathbf{S}_{\mathbf{a},\mathbf{W}} = (s_{\mathbf{a},\mathbf{w}_1}, \cdots, s_{\mathbf{a},\mathbf{w}_N})$ to represent all the relevance scores for a candidate list $\mathbf{W}$ with respect to $\mathbf{a}$. Given a large set of unlabeled speech data, we use the unsupervised rescoring

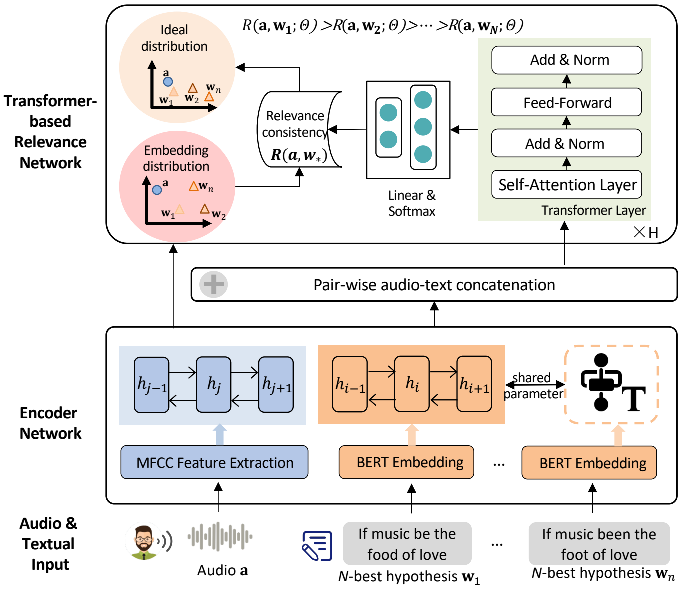

Fig. 3. The Network Architecture of Listwise Multimodal Rescoring Model used in `WSNeuRescore`

method to formulate the weak signals as the relevance. The unsupervised rescoring method can be denoted as

$$s_{\mathbf{a},\mathbf{w}} = \alpha \cdot s_{LM}(\mathbf{w}) + \beta \cdot s_{AM}(\mathbf{a}, \mathbf{w}), \tag{2}$$

where $s_{LM}(\mathbf{w})$ is the score given by the LM for word sequence $\mathbf{w}$, $s_{AM}(\mathbf{a}, \mathbf{w})$ is the score given by the AM, and α and β are pre-defined trade-off parameters. We can incoprate more sophisticated LM, such as a higher-order n-gram LM or RNNLM, as the LM in the unsupervised rescoring step, differing from the one used in the decoding period, where the LM is usually a basic n-gram LM due to the consideration of efficiency.

B. Multimodal Listwise Rescoring Model

1) Network Architecture: The architecture of the network is illustrated in Fig. 3. The network takes each training instance, in the form of $(\mathbf{a}, \mathbf{w}_1, \cdots, \mathbf{w}_N, s_1, \cdots, s_N)$, and aims to predict relevance $R(\mathbf{a}, \mathbf{w}; \Theta)$ which can accurately reflect the relative orders of $(\mathbf{w}_1, \cdots, \mathbf{w}_N)$ with respect to $\mathbf{a}$. Given the acoustic input $\mathbf{a}$ and N-best candidate $\mathbf{w}$, we extract the MFCC [27], [28] representation $(a_1, \cdots, a_g)$ and BERT embedding $(w_1, \cdots, w_l)$, where l and g are the length of the word sequence and acoustic sequence respectively. Then bi-directional LSTMs (BiLSTMs) are used to convert these sequences into hidden vectors, which is denoted as

$$\mathbf{h}^{\mathbf{a}}_k = \texttt{BiLSTM}(a_k, \mathbf{h}^{\mathbf{a}}_{k-1}), \tag{3}$$

$$\mathbf{h}^{\mathbf{w}}_j = \texttt{BiLSTM}(w_j, \mathbf{h}^{\mathbf{w}}_{j-1}), \tag{4}$$

where $j \in [1, l]$, $k \in [1, g]$. Then a multi-head self-attention (MHSA) [29] is used to calculate the similarity between $\mathbf{h}^{\mathbf{a}}$ and $\mathbf{h}^{\mathbf{w}}$. An attention module calculates scaled dot-product values of a query Q, a key K and a value V by

$$\texttt{Attention}(Q, K, V) = \texttt{Softmax}(\frac{QK^T}{\sqrt{d_k}})V, \tag{5}$$

[1] https://en.wikipedia.org/wiki/Cantonese
[2] https://en.wikipedia.org/wiki/Wu_Chinese

where d_k is the dimensionality of key K. In our scenario, the queries, keys, and values all equals to the acoustic and textual hidden state vectors, i.e., $Q = K = V = [\mathbf{h}^{\mathbf{a}}; \mathbf{h}^{\mathbf{w}}]$, where $[\cdot; \cdot]$ represents the concatenating operation. The MHSA gets h different representations and their attention values, and then concatenate these values together and projects through a feed-forward layer by

$$head_i = \texttt{Attention}(QW_i^Q, QW_i^K, VW_i^V), \tag{6}$$

$$\mathbf{M} = \texttt{MultiHead}(Q, K, V) = [head_1; \cdots; head_h]W^O, \tag{7}$$

where W_i^Q, W_i^K, W_i^V, and W^O are parameter matrices to be learned. The output of the MHSA is further fed into a max-pooling layer and then linearly combined into the final relevance score $R(\mathbf{a}, \mathbf{w}; \Theta)$ as

$$\mathbf{z} = \texttt{Max\_pooling}(\mathbf{M}), \tag{8}$$

$$R(\mathbf{a}, \mathbf{w}; \Theta) = \texttt{Softmax}(\mathbf{W}\mathbf{z} + \mathbf{b}), \tag{9}$$

where $\mathbf{W}$, $\mathbf{b}$ are training parameters, and Θ are the parameter set.

2) Listwise Model Training: There are three basic categories of LTR approaches for ranking in IR: *pointwise*, *pairwise*, and *listwise* [30]. The listwise approach is generally proven to be much better than the other two since it takes the whole ranking list concerning a given query as a training instance and closely models the nature of the ranking problem [31]. Hence, in `WSNeuRescore`, we also employ a listwise loss function defined as

$$\mathcal{L} = \sum_{(\mathbf{a}, \mathbf{W}) \in \mathcal{D}} \ell(\mathbf{S}_{\mathbf{a},\mathbf{W}}, \mathbf{S}'_{\mathbf{a},\mathbf{W}}), \tag{10}$$

where $\mathbf{S}_{\mathbf{a},\mathbf{W}}$ is the weak signals of hypothesis list $\mathbf{W}$ with respect to $\mathbf{a}$, and $\mathbf{S}'_{\mathbf{a},\mathbf{W}}$ is the relevance scores generated by the neural rescoring model, $\mathbf{S}'_{\mathbf{a},\mathbf{W}} = (R(\mathbf{a}, \mathbf{w_1}; \Theta), \ldots, R(\mathbf{a}, \mathbf{w_N}; \Theta))$. The loss function ℓ estimates the probability for different ranking orders, and in our setting, we use the top-1 probability [32] defined as

$$p_{(f)}(\mathbf{a}, \mathbf{w}) = \texttt{Softmax}(f(\mathbf{a}, \mathbf{w}; \Theta)), \tag{11}$$

where $f(\cdot)$ represents our rescoring method. Cross-entropy distance is used to measure the difference between these two probability distributions as

$$\ell(\mathbf{S}_{\mathbf{a},\mathbf{W}}, \mathbf{S}'_{\mathbf{a},\mathbf{W}}) = -\sum_{i=1}^{N} p_{(\mathbf{S})}(\mathbf{a}, \mathbf{w}_i) \log p_{(\mathbf{S}')}(\mathbf{a}, \mathbf{w}_i). \tag{12}$$

The whole network is trained in an end-to-end style with optimization methods such as Adam [33].

C. Domain Adaptation and Fine-Tuning

We apply small-scale annotated data to adapt the model pre-trained from weak supervision signals. Unlike pre-training models such as BERT [19] and Electra [34], we do not have to modify the network structure since the tasks of pre-training and fine-turning are the same. The fine-tuning is conducted relatively straightforwardly. However, the challenge of fine-tuning a weakly supervised pre-trained ASR rescoring model with a small amount of annotated data is how to avoid catastrophic forgetting, that is, preventing the parameters from being skewed by the small number of training samples.

Inspired by tricks from fine-tuning BERT [35], similar to ours in that it tunes a large model with a small amount of data, we propose a layer-wise tuning strategy, in which parameters are tuned in each layer with different learning rates. Denoting Θ_t^l as the parameters of the l-layer of our deep rescoring model at t-step, the updating rule is as follows:

$$\Theta_t^l = \Theta_{t-1}^l - \eta^l \cdot \nabla_{\Theta^l} J(\Theta), \tag{13}$$

where η^l is the learning rate for the l-layer. Setting the learning rate of the last layer η^L as η, which is the base learning rate, the learning rates decay with rate the ξ along with the layer becoming shallower, i.e.,

$$\eta^{l-1} = \xi \cdot \eta^l, \tag{14}$$

where ξ is a hyper-parameter from range (0,1]. Since each training instance $\mathbf{a}$ is labeled with the ground-truth transcript, after obtaining the N-best candidate list $\mathbf{W} = \{\mathbf{w}_1, \cdots, \mathbf{w}_N\}$, we compare each candidate with the ground-truth transcript and use the word error rate (WER) [36] as the relevance score. WER is formally defined as

$$WER = \frac{S + D + I}{T}, \tag{15}$$

where S, D, and I represent the number of word substitutions, deletions, and insertions, respectively, and T is the total number of words in the ground-truth transcript. From the definition, we can conclude that WER reflects the quality of the ASR decoding result, and the lower the value, the better the decoding result.

IV. Experiments

In this section, we describe the experimental settings and then go through the results of the evaluations.

A. Experimental Setup

1) Datasets: A commonly-used public dataset in the speech area – the TED-LIUM dataset [3] [37] – is adopted in our experiment for the rescoring model evaluation, with the detailed statistics shown in Table I. To ensure reproducibility, we build a Kaldi "Chain" model for the AM and a tri-gram LM as the "existing" ASR model for the decoding period with around 1000 hours of in-house speech data, and then fine-tune this ASR model with one-third of the TED-LIUM labeled training data. During the `WSNeuRescore` model pre-training period, we also use TED-LIUM 3 dataset (without labels) [38] to generate the weak supervision signals. Following [4], [39] and [5], we also set the N value to 50 to obtain the N-best list for each utterances.

[3] https://www.openslr.org/7/

TABLE I
The Statistics of the TED-LIUM Dataset

	Train	**Dev**	**Test**
No. of transcripts	774	8	11
No. of words	1.5M	17.8k	27.5k
No. of segments	56.8k	0.6k	1.5k
Length of waves	118 hours	1.72 hours	3.07 hours
Frequency	16kHz	16kHz	16kHz
Language	English	English	English

2) Baselines: We compare our proposed WSNeuRescore model with the following six baselines in N-best rescoring:

- Basic n-gram LM: This is a straightforward baseline, which is also the default one used in the Kaldi toolkit. We use the AM score together with the score given by a Trigram LM.
- Cache Model: This baseline belongs to the cache-based LMA category proposed in [40], and we name it Cache Model in the following comparison.
- Topic Model-based LMA [15]: We adopt an LDA model trained using LightLDA [41] with topic number K set to 50.
- RNNLM: This baseline uses RNNLM [40] to re-evaluate the LM score of the N-best candidates, which is also integrated by the Kaldi toolkit.
- EC-Model [18]: This baseline is also based on DNN to encode each N-best candidate into embeddings to do the classification.
- NS2TLM [10]: This baseline extends the RNNLM by incorporating speech signals into the hidden state.

We tune the parameters of these rescoring models to obtain their best performance. The RNNLM, EC-Model, and NS2TLM have all been proposed recently, and they are DNN-based baselines that achieve the state-of-the-art performance for the rescoring task.

3) Implementation Details: For WSNeuRescore, the dimension of BERT embedding is set to 1024, and the BiLSTM is a 2-layer one with a hidden size set to 256. The number of heads for MHSA is 8, with input and output dimensions set to 1024 and 256, respectively. During the training period, the learning rate, decay rate, dropout rate, and batch size of the network are set to 0.001, 0.8, 0.3, and 256 respectively. The experiments were conducted in a server with a 314 GB memory, 72 Intel Core Processor (Xeon), Tesla K80 GPU, and CentOS. Two measurements, NDCG@n [30] with n set to 10 and WER, are employed to evaluate the final performance. We use the default $\alpha = 1.0$ and $\beta = 0.1$ values in Kaldi in the pre-training period.

B. Experimental Results

1) NDCG@10 & WER: This set of experiments show the performance of these rescoring models where the training dataset is enough. We list the experimental results in terms of both NDCG@10 and WER in Table II. NDCG@10 reflects the quality of a ranking list, and a correctly ranked candidate list is vital for ASR in noisy environments or casual-style speech requiring multiple recognition hypotheses [18]. Different from NDCG@10, WER reflects the quality of an ASR system with only the top-1 ranked candidate. Following [4], [5], we also calculate the “Oracle Ceiling Perf.” by only including the decoding errors made in the decoding period and excluding the rescoring error. This acts as the theoretical ceiling performance for all rescoring methods. From the results, we can see that WSNeuRescore achieves the most significant improvement over other baselines. Specifically, adopting RNNLM as the LM for rescoring brings a relative NDCG@10 improvement of 3.97% over the “AM + Tri-gram” baseline. Among all these models, WSNeuRescore finally achieves a 19.84% relative improvement over the “AM + Tri-gram” baseline, and this validates the superiority of the WSNeuRescore model.

TABLE II
NDCG@10 & WER Comparison

Rescoring Methods	**Dev**		**Test**	
	NDCG@10	**WER**	**NDCG@10**	**WER**
Trigram LM	0.617	15.762	0.630	15.684
RNNLM	0.641	15.661	0.655	15.556
Cache Model	0.621	15.734	0.630	15.662
Topic Model LMA	0.621	15.802	0.632	15.695
EC-Model	0.618	15.447	0.618	15.483
NS2TLM	0.670	15.177	0.671	15.327
WSNeuRescore	0.759	13.609	0.755	12.858
Oracle Ceiling Perf.	1.0000	10.499	1.0000	8.800

2) Effectiveness of Weak Supervision for Rescoring: This set of experiments investigate the potential effect of weak supervision as a pre-training step for N-best rescoring. We list the performance of our model with 1) only pre-trained with weak supervision (only weak supervision), 2) fully trained with all the labeled data (fully supervised), and 3) pre-trained using weakly labeled data and then fine-tuned using only 1/3 of the labeled data (pre-training + fine-tuning), as well as the unsupervised Trigram LM baseline (unsupervised baseline). From Table III, we can see that the pre-training step with weak supervision is critical for obtaining a high-quality N-best model. The weakly supervision requires no labeled data and only trained with the weak labels given by the unsupervised baseline in Kaldi, and shown an 11.9% relative reduction in WER in the Test set. This validates the effectiveness of the pre-training mechanism when there is no labeled data at all for rescoring. For the scenario where we have an adequate labeled dataset to fully train a rescoring model, the pre-training mechanism with weak supervision can still boost the performance, showing a 2.94% relative WER reduction with only 1/3 of the labeled data.

In industrial practice, we usually lack domain-dependent data, and training a neural rescoring model from scratch would be infeasible. This proposed pre-training with weak supervision and fine-turning strategies can significantly alleviate the data scarcity problem and further boost performance.

V. Conclusion

The last decade has witnessed a flourish of speech-driven applications, like AI assistants, due to the popularity of smart

TABLE III
THE EFFECTIVENESS OF WEAK SUPERVISION FOR `WSNeuRescore`

Method	Dev		Test	
	NDCG@10	WER	NDCG@10	WER
Unsupervised	0.617	15.762	0.630	15.684
Only Weak Supervision	0.731	13.985	0.728	13.869
Relative Improvement	18.47%	10.45%	15.56%	11.9%
Fully Supervised	0.759	13.609	0.755	12.858
Pre-training + Fine-turning	0.768	13.282	0.764	12.480
Relative Improvement	1.19 %	2.4%	1.19%	2.94%

devices. The importance of reliable and practical ASR systems is evident. This work explores the possibility of advancing the industrial ASR system from the IR perspective. A weakly-supervised mechanism with a Multimodal neural rescoring network is proposed for pre-training neural N-best rescoring models in ASR. Experimental results have indicated that our proposed `WSNeuRescore` is effective for N-best list rescoring and opens a new door for ASR. We hope this work will inspire more research on exploring advanced unsupervised or weakly supervised machine learning and information retrieval techniques to promote the performance of ASR systems and other NLP applications.

Acknowledgements. We thank anonymous reviewers for their helpful comments. The research of Raymond Chi-Wing Wong is supported by GZSTI16EG24.

REFERENCES

[1] V. Shah, S. Li, A. Kumar, and L. Saul, "Speakql: Towards speech-driven multimodal querying of structured data," in *SIGMOD*, 2020.

[2] J. Li, R. Zhao, E. Sun, J. H. Wong, A. Das, Z. Meng, and Y. Gong, "High-accuracy and low-latency speech recognition with two-head contextual layer trajectory lstm model," in *ICASSP*, 2020.

[3] Y. Wang, A. Mohamed, D. Le, C. Liu, A. Xiao, J. Mahadeokar, H. Huang, A. Tjandra, X. Zhang, F. Zhang *et al.*, "Transformer-based acoustic modeling for hybrid speech recognition," in *ICASSP*, 2020.

[4] Y. Song, D. Jiang, X. Zhao, Q. Xu, R. C.-W. Wong, L. Fan, and Q. Yang, "L2rs: A learning-to-rescore mechanism for automatic speech recognition," in *MM*, 2021.

[5] Y. Song, D. Jiang, X. Huang, Y. Li, Q. Xu, R. C.-W. Wong, and Q. Yang, "Goldenretriever: A speech recognition system powered by modern information retrieval," in *MM*, 2020.

[6] T.-Y. Liu *et al.*, "Learning to rank for information retrieval," *Foundations and Trends® in Information Retrieval*, vol. 3, no. 3, pp. 225–331, 2009.

[7] M. Yu and D. Litman, "Leveraging linguistic coordination in reranking n-best candidates for end-to-end response selection using bert," in *FLAIRS*, 2021.

[8] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011.

[9] H. Xu, T. Chen, D. Gao, Y. Wang, K. Li, N. Goel, Y. Carmiel, D. Povey, and S. Khudanpur, "A pruned rnnlm lattice-rescoring algorithm for automatic speech recognition," in *ICASSP*, 2018.

[10] T. Tanaka, R. Masumura, T. Moriya, and Y. Aono, "Neural speech-to-text language models for rescoring hypotheses of dnn-hmm hybrid automatic speech recognition systems," in *APSIPA ASC*, 2018.

[11] J. R. Bellegarda, "Statistical language model adaptation: review and perspectives," *Speech communication*, 2004.

[12] 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.

[13] B. Roark, M. Saraclar, M. Collins, and M. Johnson, "Discriminative language modeling with conditional random fields and the perceptron algorithm," in *ACL*, 2004.

[14] R. Kuhn and R. De Mori, "A cache-based natural language model for speech recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 12, no. 6, pp. 570–583, 1990.

[15] K.-Y. Chen, H.-S. Chiu, and B. Chen, "Latent topic modeling of word vicinity information for speech recognition," in *ICASSP*, 2010.

[16] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Eleventh annual conference of the international speech communication association*, 2010.

[17] T. Mikolov, S. Kombrink, L. Burget, J. Černockỳ, and S. Khudanpur, "Extensions of recurrent neural network language model," in *ICASSP*, 2011.

[18] 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 *ICASSP*, 2018.

[19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *NAACL*, 2019.

[20] Z.-H. Zhou, "A brief introduction to weakly supervised learning," *National Science Review*, vol. 5, no. 1, pp. 44–53, 2018.

[21] M. Dehghani, H. Zamani, A. Severyn, J. Kamps, and W. B. Croft, "Neural ranking models with weak supervision," in *SIGIR*, 2017.

[22] K. Zhang, C. Xiong, Z. Liu, and Z. Liu, "Selective weak supervision for neural information retrieval," in *WWW*, 2020.

[23] H. Zamani and W. B. Croft, "On the theory of weak supervision for information retrieval," in *SIGIR*, 2018.

[24] D. Haddad and J. Ghosh, "Learning more from less: Towards strengthening weak supervision for ad-hoc retrieval," in *SIGIR*, 2019.

[25] K. Singh, D. Okhonko, J. Liu, Y. Wang, F. Zhang, R. Girshick, S. Edunov, F. Peng, Y. Saraf, G. Zweig *et al.*, "Training asr models by generation of contextual information," in *ICASSP*, 2020.

[26] K. Singh, V. Manohar, A. Xiao, S. Edunov, R. Girshick, V. Liptchinsky, C. Fuegen, Y. Saraf, G. Zweig, and A. Mohamed, "Large scale weakly and semi-supervised learning for low-resource video asr," *Interspeech*, 2020.

[27] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE transactions on acoustics, speech, and signal processing*, vol. 28, no. 4, pp. 357–366, 1980.

[28] B. Logan *et al.*, "Mel frequency cepstral coefficients for music modeling." in *ISMIR*, vol. 270, 2000, pp. 1–11.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017.

[30] C. Manning, P. Raghavan, and H. Schütze, "Introduction to information retrieval," *Natural Language Engineering*, vol. 16, no. 1, pp. 100–103, 2010.

[31] J. Sánchez, F. Luque, and L. Lichtensztein, "A structured listwise approach to learning to rank for image tagging," in *ECCV*, 2018.

[32] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li, "Learning to rank: from pairwise approach to listwise approach," in *ICML*, 2007.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," Tech. Rep., 2014.

[34] K. Clark, M. Luong, Q. V. Le, and C. D. Manning, "ELECTRA: pre-training text encoders as discriminators rather than generators," in *ICLR*, 2020.

[35] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to fine-tune bert for text classification?" in *China National Conference on Chinese Computational Linguistics*. Springer, 2019, pp. 194–206.

[36] D. Klakow and J. Peters, "Testing the correlation of word error rate and perplexity," *Speech Communication*, vol. 38, no. 1-2, pp. 19–28, 2002.

[37] A. Rousseau, P. Deléglise, and Y. Estève, "Ted-lium: an automatic speech recognition dedicated corpus," in *LREC*, 2012.

[38] F. Hernandez, V. Nguyen, S. Ghannay, N. Tomashenko, and Y. Estève, "Ted-lium 3: twice as much data and corpus repartition for experiments on speaker adaptation," in *International Conference on Speech and Computer*. Springer, 2018, pp. 198–208.

[39] X. Liu, Y. Wang, X. Chen, M. J. F. Gales, and P. Woodland, "Efficient lattice rescoring using recurrent neural network language models," in *IEEE International Conference on Acoustics*, 2014.

[40] K. Li, H. Xu, Y. Wang, D. Povey, and S. Khudanpur, "Recurrent neural network language model adaptation for conversational speech recognition," *Interspeech*, 2018.

[41] J. Yuan, F. Gao, Q. Ho, W. Dai, J. Wei, X. Zheng, E. P. Xing, T.-Y. Liu, and W.-Y. Ma, "Lightlda: Big topic models on modest computer clusters," in *WWW*, 2015.