Learning to rank with BERT-based confidence models in ASR rescoring

Ting-Wei Wu[1,*], *I-Fan Chen*[2], *Ankur Gandhe*[2]

[1]Georgia Institute of Technology
School of Electric Computer and Engineering
[2]Amazon, Seattle, Washington

waynewu@gatech.edu, {ifanchen, aggandhe}@amazon.com

Abstract

We propose a learning-to-rank (LTR) approach to the ASR rescoring problem. The proposed LTR framework has the flexibility of embracing wide varieties of linguistic, semantic, and implicit user feedback signals in rescoring process. BERT-based confidence models (CM) taking account of both acoustic and text information are also proposed to provide features better representing hypothesis quality to the LTR models. We show the knowledge of the entire N-best list is crucial for the confidence and LTR models to achieve best rescoring results. Experimental results on de-identified Alexa data show the proposed LTR framework provides an additional 5.16% relative word error rate reduction (WERR) on top of a neural language model rescored ASR system. On LibriSpeech, a 9.38 % WERR and a 13.63 % WERR are observed on the *test-clean* and *test-other* sets, respectively.

Index Terms: ASR 2nd-pass rescoring, learning to rank, confidence model, LambdaMART, BERT

1. Introduction

Rescoring is an approach allowing automatic speech recognition (ASR) systems [1, 2] to use large and powerful models for better accuracy while accommodating the latency constraint of the ASR services. In such setup, a small, efficient, but less accurate ASR model is used to generate a pruned hypothesis space, usually represented as lattices [3, 4] or N-best hypotheses [5, 6, 7, 8], for large models to rescore and select the best results from it. The large models are usually more powerful language models trained discriminatively [5] or with more complex model architectures [9, 10]. Recent research studies [6, 7, 11] demonstrate superior rescoring performance can be achieved by jointly considering text and audio information. For modern ASR systems, context signals such as the audio signal-to-noise ratio, device directedness scores for the input speech, etc., might also be available at rescoring time in addition to the audio and hypothesis texts. How to leverage those signals for better rescoring becomes a research question.

Despite using additional non-text information for rescoring, the above mentioned methods [6, 7, 11] rescore each hypothesis in the N-best list independently, without considering the information from the other hypotheses of the current utterance or the whole hypothesis space represented by the list. One research direction is to leverage the context sentence information to improve the prediction for the current utterance [12]; while the others attempt to adopt pairwise or listwise ranking models including pairwise RankSVM [13] and neural network based ranking [14, 15] for the rescoring process to leverage information from the other hypotheses of the current utterance and have shown promising results.

In this paper, we propose a learning-to-rank (LTR) framework able to easily include a variety of signals and consider global info of the whole N-best list in the rescoring process. Unlike [13, 14, 15], where pairwise ranking is used, we choose listwise approaches to implement our rescoring system. We adopt LambdaMART [16], a state-of-the-art listwise LTR model, in our research to better capture the global information in the N-best list and leverage a wide variety of signals that may help the rescoring process. Additional signals including non-device directiveness, ASR confidence, hypothesis rewrite, signal-to-noise ratio (SNR) are explored in this paper.

The choice of the listwise approach also enables us to build new BERT-based confidence models (CM) leveraging the listwise information from the ASR N-best for hypothesis confidence scores generation, which could not be easily achieved in the pairwise scenario. Unlike the other utterance level confidence models leveraging the features derived from the individual hypothesis [17, 18, 19], the proposed listwise BERT-based CMs consume the whole N-best list as the input and are trained with losses taking the whole N-best list into account. These BERT-based confidence models can be either used alone for N-best rescoring or provide additional features to the LambdaMART rescoring model. Our experimental result shows adopting listwise info at both feature and ranking (rescoring) level is important for achieving the best rescoring results.

2. Methodology

2.1. *N*-best Rescoring

ASR decoding is to search the word sequence w with the highest probability given the observed acoustic evidence a of the input audio. As shown in Eq. 1, the probability can be factorized into $P_{AM}(a|\overrightarrow{w})$ and $P_{LM}(\overrightarrow{w})$ modeled by acoustic and language models respectively.

$$\overrightarrow{w}^* = \arg\max_{\overrightarrow{w}} P(\overrightarrow{w}|a) = \arg\max_{\overrightarrow{w}} P_{AM}(a|\overrightarrow{w})P_{LM}(\overrightarrow{w})) \tag{1}$$

The search space of $\overrightarrow{w}$ in Eq. 1 can be large. In streaming ASR, a first pass beam search with smaller models are usually used to generate a N-best list, $\mathbf{H}$. More complex models, e.g., LM$'$, can then be used to recompute Eq. 1 within the search space $\mathbf{H}$:

$$\overrightarrow{w}^* = \arg\max_{\overrightarrow{w}_i \in \mathbf{H}} (\lambda\, logP_{AM}(a|\overrightarrow{w}_i) + \ logP_{LM'}(\overrightarrow{w}_i)), \tag{2}$$

where λ is an interpolation weight between the AM and LM$'$ *scores*. In Eq. 2, the computation of AM and LM scores only consider the current hypothesis $\overrightarrow{w}_i$ and ignore the hypotheses $\overrightarrow{w}_{j\setminus i}$ in the same list that may indicate the quality of $\overrightarrow{w}_i$ as well

*The first author performed the work during an internship at Amazon.

as other context signals that may help $\overrightarrow{w}^*$ selection. To address the issue, in this paper, we reformulate N-best rescoring as a learning-to-rank process, where we choose $\overrightarrow{w}^*$ directly based on its predicted ranks from $\phi(\cdot)$ within $\mathbf{H}$ in Eq. 3 to optimize WER.

$$\overrightarrow{w}^* = \arg\max_{\overrightarrow{w}_i \in \mathbf{H}} \phi(f(a, \overrightarrow{w}_i, \mathbf{s}_i, \mathbf{H})). \quad (3)$$

In Eq. 3, $\phi(\cdot)$ is the ranking function taking features generated by the feature function $f(\cdot)$ based on acoustic evidence a, current hypothesis $\overrightarrow{w}_i$, other auxiliary features $\mathbf{s}_i = g(a, \overrightarrow{w}_i)$ derived from the acoustic evidence and the text of the hypothesis based on a feature extraction function $g(.)$, and the entire N-best list $\mathbf{H}$ jointly.

2.2. Learning-to-Rank model

Learning-to-rank (LTR) is a class of techniques aiming to solve ranking problems in information retrieval [20]. In this paper, we choose LambdaMART [16], a state-of-the-art LTR model considering the whole candidate list in the ranking process, as our ranking model. LambdaMART is a boosted tree based model trained with lambda gradient λ_i from ranking loss [1]. The LambdaMART model generates a ranking score, r_i, for each N-best hypothesis. We interpolate r_i with AM and LM scores following [13] to create the final ranking score for N-best ranking.

2.3. Features for Learning-to-Rank models

The boosting-tree based LambdaMART model has the advantage of being easy to adopt heterogeneous features for the ranking process. In this paper, we explore up to fourteen features that may be available to a modern ASR system. The fourteen features are grouped in catalogs listed below with the number of features in the catalog shown in ():

- **ASR model scores** (4): The AM, LMs, and the rescoring NLM scores generated by the baseline ASR system during the decoding process.
- **Regression-based ASR confidence score** (2): Scores generated by a regression-based confidence model with input features including the posterior probability of the hypothesis, word prior probability, etc. as described in [17]. Both original and histogram normalized scores are used.
- **Device-Directedness scores** (2): Scores generated by device-directed speech detection models described in [22, 23, 24]. This is a useful signal in far-field ASR systems to reduce insertion errors caused by background speech. Both original and histogram normalized scores are used.
- **BERT MLM scores** (2): We follow [25] to generate BERT MLM scores for each hypothesis from a pre-trained BERT.
- **ASR 1-best hypothesis rewrites** (2): hypothesis rewrites reduce customer friction from a voice assistant service based on the usage history [26, 27]. By learning popular query rephrase from history, a rewrite system generates alternative transcription for a given ASR hypothesis. We assume the rewrites are closer to the ground truth of what the speaker says. We use word-level Levantine distance and sentence embedding cosine similarity as two numerical features for each hypothesis.
- **Other features** (2): hypothesis length and signal-noise ratio (SNR) are used as two additional features to LTR models.

[1] In this paper NDCG [21] is used. The target rank for each hypothesis in the N-best list is base on their WER.

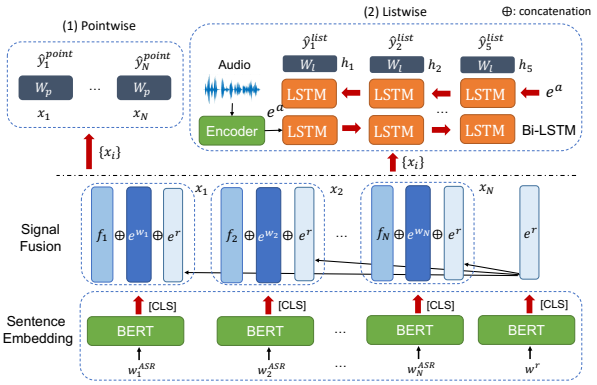

Figure 1: *The proposed BERT-based confidence model structure. The N-best hypothesis and rewrite are encoded via BERT. The fused signals are fed into a BiLSTM for listwise CMs (or directly fed into a feedforward network for pointwise CMs) to generate confidence scores.*

2.4. BERT-based Confidence Model

Although LambdaMART uses the entire list in the ranking process, the features of each N-best hypothesis are independent to each other. Further, our analysis indicates having a high-quality feature representing N-best hypothesis quality is crucial for reducing the performance gap between the 1-best and oracle hypothesis for LTR rescorers. This motivated us to propose new BERT[28]-based confidence models exploiting listwise information from the N-best list.

2.4.1. Model structure

Figure 1 illustrates the proposed BERT-based confidence models. In the proposed model structure, the sentence embedding of each N-best hypothesis text $\{\overrightarrow{w}_i^{ASR}\}$ are computed by a domain[2] fine-tuned BERT. We take the hidden states of the BERT [CLS] token output as the embedding vector $e^{\overrightarrow{w}_i}$ representing the given sentence text. A similar sentence embedding is also computed for the utterance rewrite text $\overrightarrow{w}^r$ to form the rewrite embedding vector, e^r. In the signal fusion layer, for the i-th hypothesis, its feature vector f_i, representing all the features described in section 2.3 and sentence embedding $e^{\overrightarrow{w}_i}$ are concatenated along with the global rewrite embedding e^r (shared among all hypotheses) to form the final feature vector x_i served as the input to the confidence score generation layer.

In this research, we consider two types of confidence score generation layers: pointwise and listwise. For the pointwise setup (as illustrated in the Figure 1 "(1) Pointwise" branch), the feature vector x_i for a hypothesis $\overrightarrow{w}_i^{ASR}$ is directly projected to a single score $\hat{y}_i^{point} = \sigma(W_p x_i)$ with a feedforward layer W_p.

The pointwise setup does not consider other competing hypotheses in the N-best list for confidence score generation. In the listwise setup (as illustrated in the Figure 1 "(2) Listwise" branch), we utilize a BiLSTM model to combine the input feature vectors x_i from the whole N-best list along with an acoustic embedding vector e^a extracted from the ASR encoder model state to jointly predict the confidence scores for each N-best hypothesis. More specifically, the acoustic embedding e^a for a given audio is extracted as the last hidden encoder state at

[2] For example, the Alexa task domain.

the last audio frame. We project the embedding e^a to the hidden state space of the BiLSTM and use the projected values to initialize the hidden states of the BiLSTM forward and backward paths. The BiLSTM consumes the sequence of x_i vectors, and the hidden layer output of the BiLSTM, h_i, at position i (i.e., the ith-best) is then projected into a single confidence score $\hat{y}_i^{list} = \sigma(W_l h_i)$ for $\overrightarrow{w}_i^{ASR}$. To the authors' knowledge, the proposed listwise CM is the first BERT-based CM with the flexibility to take the whole ASR N-best list disregarding the value of N. Using a BiLSTM layer is the key to achieve that.

2.4.2. Model Training

Four training objectives for the proposed BERT-based CM are explored in this paper:

- **bce_gt**: binary cross entropy loss with target=1 if the hypothesis matches ground truth transcription, else 0.
- **bce_mwer**: binary cross entropy loss with target=1 if the hypothesis has the lowest WER among the N-best list, else 0
- **ce_ht_mwer**: cross entropy loss over the N CM outputs with hard targets where the target=1 if the output corresponding to the hypothesis with minimal WER, else 0
- **ce_st**: cross entropy loss over the N CM outputs with soft targets defined as softmax smoothed hypothesis accuracy, .i.e., $y_i = e^{-WER_i} / \sum_{j=1}^{N} e^{-WER_j}$

The first two training criteria can be applied to both pointwise and listwise CMs since they compare the ground truth independently; while the rest two can only be applied to listwise CMs as they treat the prediction for the entire list as a single best hypothesis classification problem.

3. Experiments

3.1. Experimental Setup

Experiments are conducted on de-identified en-US Alexa data and the LibriSpeech dataset [29]. For Alexa experiments, the baseline ASR system is a hybrid ASR system with AM and LM in the first pass decoding plus a second pass rescoring with NLM. We use the baseline system to generate the N-best lists for train, dev, and eval sets, which consist of 280 hours, 13 hours, and 12 hours of data. The training data is used for LTR models and BERT-based CMs training. The maximal N for ASR N-best is set to 5 for N-best generation. We leverage all features listed in section 2.3 for our LTR framework rescoring.

For LibriSpeech experiments[3], we adopt the ESPnet2 [30, 31] pretrained Conformer ASR model with HuBERT self-supervised learning features and a rescoring Transformer-LM as the baseline system. The baseline system without LM rescoring provides WER of 1.92% and 4.11% on the LibriSpeech *test-clean* and *test-other* datasets, respectively. When the Transformer-LM is used for N-best rescoring, the WER can be further reduced to 1.81% and 3.71%. We decoded the LibriSpeech *train-clean-100*, *dev-clean*, and *dev-other* datasets with the baseline system to generate 30-best lists for LambdaMART and BERT-based confidence model training. Due to availability, we only use ASR model scores (i.e., ctc, decoder, and Transformer-LM scores) from the baseline system and the hypothesis length as features for the BERT-based CMs and the LambdaMART model.

[3]The code and data for the LibriSpeech experiments can be found at `https://github.com/waynewu6250/ExampleCode-LTRwithBERTCM-ASR_Rescoring`.)

Table 1: *Alexa Experiments: Word Error Rate Reduction (WERR) (%) of different rescoring systems.*

System	WERR
Baseline	-
Baseline (Weight Tuned)	0.20%
Transformer LM	1.69%
LambdaRank	3.08%
LamdaMART	**3.37**%
Oracle	20.24%

Our LTR framework is built on top of LightGBM [32]. The BERT-based CMs are based on $\text{BERT}_{\text{BASE}}$ model[28], with the attached 12 layers and 12 self-attention heads. The $\text{BERT}_{\text{BASE}}$ model parameters are updated in the CM training process together with the attached layers. We use an Adam optimizer with the learning rate of 1e-6 and the batch size of 32 utterances for training. All the models are trained for 20 epochs, saving the best epoch on the dev set for evaluation. For the Alexa experiments, we show the relative word error reduction (WERR) of each rescoring system over the baseline system[4]. For the LibriSpeech experiments, absolute word error rate (WER) numbers are provided. To verify the significance of the observed WER differences between different systems, for each LibriSpeech experiment, we repeat the experiment 10 times and perform the two-tailed paired student's *t*-test when comparing the system performance. We calculate the *p*-values and examine the significance based on 95% confidence with the threshold α=0.05.

3.2. Results

3.2.1. Alexa Experiments

Four types of rescoring systems in addition to the baseline are implemented for comparison:

- **Baseline(Weight Tuned)**: We rescore the baseline N-best list with the optimized AM, LM, and NLM interpolation weights. The performance of the system can be considered as the upper bound of the baseline system.
- **Transformer LM**: We follow [6] to implement a transformer-based rescorer model taking both audio embeddings and hypothesis text into account for rescoring score generation. The training data is the same 280 hours used for LTR models and BERT-based CMs training.
- **LambdaRank**: A neural-based LTR model to be compared with LambdaMART based system.
- **LambdaMART**: The boosting-tree-based LTR model proposed for N-best rescoring in this paper.

Table 1 shows the word error rate reduction (WERR) of different rescoring system over the baseline system. The LambdaRank[5] and LambdaMART systems are using the same feature set listed in section 2.3. Both LTR-based rescoring systems show larger WERR than the other rescoring approaches with WERR. The LambdaMART based system has best WERR at 3.37%, where listwise information through lambda gradients is verified to benefit the rescoring.

Table 2 shows the experimental results with the proposed BERT-based CMs. The row *BERT-based CM* in the table shows the rescoring result using the confidence scores directly

[4]All WERs are below 15% in this paper.

[5]Implemented with LightGBM[32]

Table 2: *Alexa Experiments: WERR (%) of the systems with different BERT-based Confidence Model configurations.*

System	BERT CM Config	WERR
Baseline		-
LambdaMART	-	3.37%
BERT-based CM	listwise, bce_gt	2.18%
LambdaMART	pointwise, bce_gt	3.57%
LambdaMART	listwise, bce_mwer	4.37%
LambdaMART	listwise, bce_gt	**4.46%**
LambdaMART	CM ensemble	5.16%

Table 3: *Alexa Experiments: Ablation study for feature importance analysis. We compared the percentage drop of the WERR when removing each feature from the LamdaMART CM ensemble system. "lw" means listwise. The BERT CM based features are highlighted in* ***bold****.*

System	WERR (%)	Percentage drop (%)
Oracle WER	20.24%	-
Baseline	-	-
LambdaMART CM ensemble	5.16%	-
– BERT CM lw bce_gt	4.38%	**15.00%**
– BERT CM lw ce_ht_mwer	4.42%	**14.23%**
– device directedness	4.56%	11.54%
– BERT CM lw ce_st	4.56%	**11.54%**
– rewrite/hyp cosine similarity	4.65%	9.81%
– regression-based ASR CM	4.67%	9.42%
– hyp_length	4.70%	8.85%
– rewrite/hyp Levenshtein dist	4.70%	8.85%
– BERT CM pointwise bce_gt	4.74%	**8.08%**
– LM scores	4.76%	7.69%
– SNR	4.84%	6.15%
– NLM score	4.98%	3.46%
– AM score	5.07%	1.73%

(without LambdaMART). The listwise bce_gt based CM alone (2.18%) outperforms the Transformer rescorer with a similar model architecture trained on the same training set (1.69%) in Table 1, showing the importance of listwise information. When including its output as an additional feature to the LambdaMART model, we further achieve 4.46% WERR. We achieve the best WERR at 5.16% by adopting all BERT CMs, showing the CMs score are to some extent complementary to each other, and the LambdaMART model is able to leverage that complementary information for better rescoring.

3.2.2. Ablation Study: Feature Importance Analysis

Table 3 shows the feature importance analysis of the LambdaMART BERT CM ensemble model. The BERT-based CM scores are the most important features, followed by the device directedness score, which is important for reducing insertion caused by background speech. For the BERT CMs, the result that "lw bce_gt" and "lw ce_ht_mwer" being the top two most important features shows having listwise information for CM is crucial for rescoring performance.

3.2.3. LibriSpeech Experiments

Table 4 compares the WER performance on LibriSpeech *test-clean* and *test-other* sets. The baseline system is the ESPnet2 conformer model without Transformer LM rescoring. Rescoring the baseline system with the LamdaMART model provides about 1% WERR on both *test-clean* and *test-other* sets. When further adding the BERT-based CM score as an additional feature to the LambdaMART model, the WERs are reduced to 1.78% and 3.70% (i.e., 7.29% and 9.98% WERR) on *test-clean* and *test-other* sets, respectively.

Table 4: *LibriSpeech Experiments: WER (%) of the systems on test-clean and test-other.*

System	test-clean		test-other	
(%)	WER	WERR	WER	WERR
Baseline	1.92	-	4.11	-
LambdaMART	1.90	1.04	4.07	0.97
LambdaMART w/ BERT CM	**1.78**	**7.29**	**3.70**	**9.98**
Baseline w/ Transformer LM	1.81	5.73	3.71	9.73
LambdaMART w/ Transformer LM + BERT CM	**1.74**	**9.38**	**3.55**	**13.63**
Oracle	0.67	65.10	1.83	55.47

Table 5: *LibriSpeech Experiments: The derived p-values when comparing four pairs of the systems in Table 4. All the WER differences for the selected pairs are* $\alpha = 0.05$ *significant.*

System1	System2	test-clean	test-other
Baseline	LambdaMART	1.4e-6	7.7e-5
LambdaMART	LambdaMART w/BERT CM	7.1e-19	1.7e-17
Baseline w/ Transformer LM	LambdaMART w/ BERT CM	2.7e-3	3.2e-2
Baseline w/ Transformer LM	LambdaMART w/ Transformer LM + BERT CM	1.7e-4	1.5e-8

Even when comparing with a stronger baseline (i.e., Baseline w/ Transformer LM, which achieves 1.81% and 3.71% WER on *test-clean* and *test-other* with N-best rescoring), the proposed LambdaMART w/ BERT CM rescoring framework still significantly outperforms the baseline system with the p-values of 2.7e-3 and 3.2e-2, as shown in the third row of Table 5. Finally, when providing all the available features including the Transformer LM and BERT-based CM scores to the LambdaMART model, the proposed system achieves the best WER at 1.74% and 3.55% on *test-clean* and *test-other* sets. The results confirm the importance of listwise information in the rescoring process.

4. Conclusion

In this paper, we propose a LambdaMART based LTR framework with BERT-based confidence models (CMs) leveraging listwise information for N-best rescoring. We show the framework can easily incorporate a variety of features to assist the rescoring task. We also demonstrate that it is crucial to leverage group information at both feature and ranking model levels to achieve better results. The proposed LambdaMART system with BERT-based CMs outperforms other rescoring baselines and achieves a 5.16% WERR on de-identified Amazon Alexa data, a 9.38 % WERR on the LibriSpeech *test-clean* and a 13.63 % on *test-other*.

5. References

[1] N. Moritz, T. Hori, and J. L. Roux, "Streaming end-to-end speech recognition with joint ctc-attention based models," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 936–943.

[2] T. Tanaka, R. Masumura, T. Moriya, T. Oba, and Y. Aono, "A Joint End-to-End and DNN-HMM Hybrid Automatic Speech Recognition System with Transferring Sharable Knowledge," in *Proc. Interspeech 2019*, 2019, pp. 2210–2214.

[3] E. Beck, W. Zhou, R. Schlüter, and H. Ney, "Lstm language models for lvcsr in first-pass decoding and lattice-rescoring," *ArXiv*, vol. abs/1907.01030, 2019.

[4] K. Li, D. Povey, and S. Khudanpur, "A parallelizable lattice rescoring strategy with neural language models," *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6518–6522, 2021.

[5] 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, 2012.

[6] A. Li, C. Zheng, C. Fan, R. Peng, and X. Li, "A Recursive Network with Dynamic Attention for Monaural Speech Enhancement," pp. 1–5, 2020. [Online]. Available: http://arxiv.org/abs/2003.12973

[7] A. Gandhe and A. Rastrow, "Audio-Attention Discriminative Language Model for ASR Rescoring," in *Proc. of ICASSP*, vol. 2020-May, 2020, pp. 7944–7948.

[8] 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 *Proc. of APSIPA ASC*, 2018, pp. 196–200.

[9] T. Mikolov, S. Kombrink, L. Burget, J. Černocký, and S. Khudanpur, "Extensions of recurrent neural network language model," in *Proc. of ICASSP*, 2011, pp. 5528–5531.

[10] K. Irie, A. Zeyer, R. Schlüter, and H. Ney, "Language modeling with deep transformers," *Proc. of Interspeech*, Sep 2019. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2225

[11] T. N. Sainath, R. Pang, D. Rybach, Y. He, R. Prabhavalkar, W. Li, M. Visontai, Q. Liang, T. Strohman, Y. Wu, I. McGraw, and C. C. Chiu, "Two-pass end-to-end speech recognition," in *Proc. of INTERSPEECH*, vol. 2019-Septe, 2019, pp. 2773–2777.

[12] K. Irie, A. Zeyer, R. Schlüter, and H. Ney, "Training language models for long-span cross-sentence evaluation," in *Proc. of ASRU*, 2019, pp. 419–426.

[13] 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," 2019. [Online]. Available: http://arxiv.org/abs/1910.11496

[14] D. Fohr and I. Illina, "Dnn-based semantic model for rescoring n-best speech recognition list," *CoRR*, vol. abs/2011.00975, 2020. [Online]. Available: https://arxiv.org/abs/2011.00975

[15] Z. Zhou, X. Song, R. Botros, and L. Zhao, "A Neural Network Based Ranking Framework to Improve ASR with NLU Related Knowledge Deployed," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2019-May, pp. 6450–6454, 2019.

[16] C. J. Burges, "From ranknet to lambdarank to lambdamart: An overview," Tech. Rep. MSR-TR-2010-82, June 2010. [Online]. Available: https://www.microsoft.com/en-us/research/publication/from-ranknet-to-lambdarank-to-lambdamart-an-overview/

[17] P. Swarup, R. Maas, S. Garimella, S. H. Mallidi, and B. Hoffmeister, "Improving ASR Confidence Scores for Alexa Using Acoustic and Hypothesis Embeddings," in *Proc. INTERSPEECH*, 2019, pp. 2175–2179. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-1241

[18] Q. Li, D. Qiu, Y. Zhang, B. Li, Y. He, P. C. Woodland, L. Cao, and T. Strohman, "Confidence estimation for attention-based sequence-to-sequence models for speech recognition," *Proc. of ICASSP*, pp. 6388–6392, 2021.

[19] A. Afshan, K. Kumar, and J. Wu, "Sequence-Level Confidence Classifier for ASR Utterance Accuracy and Application to Acoustic Models," in *Proc. INTERSPEECH*, 2021, pp. 4084–4088.

[20] T. Y. Liu, "Learning to rank for Information Retrieval," *Foundations and Trends in Information Retrieval*, vol. 3, no. 3, pp. 225–231, 2009.

[21] Y. Wang, L. Wang, Y. Li, D. He, W. Chen, and T. Y. Liu, "A theoretical analysis of NDCG ranking measures," in *Journal of Machine Learning Research*, vol. 30, 2013, pp. 25–54.

[22] X. Tong, C. W. Huang, S. H. Mallidi, S. Joseph, S. Pareek, C. Chandak, A. Rastrow, and R. Maas, "Streaming ResLSTM with Causal Mean Aggregation for Device-Directed Utterance Detection," *Proc. IEEE SLT Workshop*, pp. 659–664, 2021.

[23] S. H. Mallidi, R. Maas, K. Goehner, A. Rastrow, S. Matsoukas, and B. Hoffmeister, "Device-directed utterance detection," *Proc. Interspeech*, vol. 2018-Septe, pp. 1225–1228, 2018.

[24] H. Huang and F. Peng, "An Empirical Study of Efficient ASR Rescoring with Transformers," 2019. [Online]. Available: http://arxiv.org/abs/1910.11450

[25] J. Salazar, D. Liang, T. Q. Nguyen, and K. Kirchhoff, "Masked language model scoring," in *Proc. of ACL*. Association for Computational Linguistics, Jul. 2020, pp. 2699–2712. [Online]. Available: https://aclanthology.org/2020.acl-main.240

[26] P. Ponnusamy, A. R. Ghias, C. Guo, and R. Sarikaya, "Feedback-based self-learning in large-scale conversational AI agents," in *AAAI*, 2020. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/7022

[27] Z. Chen, X. Fan, Y. Ling, L. Mathias, and C. Guo, "Pre-training for query rewriting in A spoken language understanding system," *CoRR*, vol. abs/2002.05607, 2020. [Online]. Available: https://arxiv.org/abs/2002.05607

[28] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. of NAACL HLT*, vol. 1, 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423.pdf

[29] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *Proc. of ICASSP*, 2015, pp. 5206–5210.

[30] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-end speech processing toolkit," in *Proc. of INTERSPEECH*, 2018, pp. 2207–2211. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1456

[31] Espnet2 asr model: Self-supervised learning features hubert_large_ll60k, conformer, utt_mvn with transformer-lm. [Online]. Available: https://github.com/espnet/espnet/blob/master/egs2/librispeech/asr1/README.md#self-supervised-learning-features-hubert_large_ll60k-conformer-utt_mvn-with-transformer-lm

[32] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," *Proc. of NeurIPS*, vol. 2017-Decem, pp. 3147–3155, 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf