L2RS: A Learning-to-Rescore Mechanism for Hybrid Speech Recognition

Yuanfeng Song[1,2], Di Jiang[2], Xuefang Zhao[2], Qian Xu[2]
Raymond Chi-Wing Wong[1], Lixin Fan[2], Qiang Yang[1,2]
[1]The Hong Kong University of Science and Technology [2]AI Group, WeBank Co., Ltd, China
{songyf,raywong,qyang}@cse.ust.hk

ABSTRACT

This paper aims to advance the performance of industrial ASR systems by exploring a more effective method for N-best rescoring, a critical step that greatly affects the final recognition accuracy. Existing rescoring approaches suffer the following issues: (i) limited performance since they optimize an unnecessarily harder problem, namely predicting accurate grammatical legitimacy scores of the N-best hypotheses rather than directly predicting their partial orders regarding a specific acoustic input; (ii) hard to incorporate various information by advanced natural language processing (NLP) models such as BERT to achieve a comprehensive evaluation of each N-best candidate. To relieve the above drawbacks, we propose a simple yet effective mechanism, *Learning-to-Rescore* (L2RS), to empower ASR systems with state-of-the-art information retrieval (IR) techniques. Specifically, L2RS utilizes a wide range of textual information from the state-of-the-art NLP models and automatically deciding their weights to directly learn the ranking order of each N-best hypothesis with respect to a specific acoustic input. We incorporate various features including BERT sentence embeddings, the topic vectors, and perplexity scores produced by an n-gram language model (LM), topic modeling LM, BERT, and RNNLM to train the rescoring model. Experimental results on a public dataset show that L2RS outperforms not only traditional rescoring methods but also its deep neural network counterparts by a substantial margin of 20.85% in terms of NDCG@10. The L2RS toolkit has been successfully deployed for many online commercial services in WeBank Co., Ltd, China's leading digital bank. The efficacy and applicability of L2RS are validated by real-life online customer datasets.

CCS CONCEPTS

• **Information systems** → **Learning to rank**; • **Artificial intelligence** → **Speech recognition**;

KEYWORDS

Learning-to-Rank, Automatic Speech Recognition, Information Retrieval, N-best Rescoring

MM '21, October 20–24, 2021, Virtual Event, China
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3481542

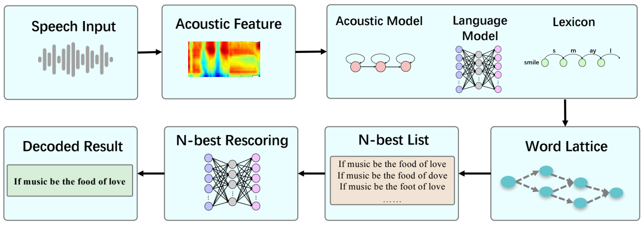

Figure 1: The typical pipeline of a hybrid ASR system, including a *Decoding* step and an N-best *Rescoring* Step

ACM Reference Format:
Yuanfeng Song, Di Jiang, Xuefang Zhao, Qian Xu, Raymond Chi-Wing Wong, Lixin Fan, Qiang Yang. 2021. L2RS: A Learning-to-Rescore Mechanism for Hybrid Speech Recognition. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21), October 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3474085.3481542

1 INTRODUCTION

Due to the ubiquity of mobile devices in daily life, speech communication has gained significant momentum in recent years. A reliable automatic speech recognition (ASR) system converts speech signals into their corresponding transcripts, and it is the premise of the success of existing speech-driven applications such as voice assistants, voice databases, speech retrieval systems, and so on. Even though end-to-end ASR models have engaged increasing attention in the research community [1, 10], in terms of the industrial applications of ASR systems, the *hybrid* ones[1] with several distinct components - an acoustic model (AM), a pronunciation model (PM), and a language model (LM), still dominate the area due to its modularization, robustness, and flexibility [29]. Hence, advancing the hybrid ASR systems is still a hot and valuable topic that attracts many researchers and engineers [28, 50, 54].

In a typical hybrid system, the AM focuses on translating the features (for example, MFCC [16]) extracted from speech signals into their phoneme representation and the LM evaluates the likelihood of the word sequences from the grammatical legitimacy perspective. Figure 1 describes the pipeline of such a hybrid system, which can roughly be divided into two periods: *decoding* and *rescoring*. The decoding period involves the decoding on a precompiled static graph based on a basic n-gram LM to get a set of hypotheses, while the rescoring period employs an N-best rescoring model to rank these

[1] We use the term "hybrid" to emphasize the fact that different from the end-to-end ASR systems, the hybrid systems preserve distinct AM, PM, and LM, however, each AM or LM component can be deep neural network (DNN)-based.

hypotheses, and selects the most promising one as the final recognition result. Since more sophisticated LMs such as the DNN-based ones (e.g., RNNLMs) theoretically encode infinite word history, they cannot be used in the decoding period as it is impossible to compile them into a static decoding graph [56]. The most common way to make use of these DNN-based LMs is by incorporating them into the *N*-best rescoring, and this makes the *N*-best rescoring a critical step that greatly affects the final recognition accuracy.

Although existing rescoring models have shown promising performance, they still suffer the following issues: (i) they rely on scores produced by an AM and LM to rescore and rank the *N*-best lists, even when the LM is a more sophisticated one such as RNNLM or neural speech-to-text LM (NS2TLM) [49]; (ii) the weight of formulating the AM and LM scores into the final ranking score is usually determined empirically and much effort is required to tune this parameter; (iii) existing rescoring methods such as the EC-Model [40] utilizes quite limited information, and the vast arsenal of the state-of-the-art models for gauging linguistic and semantic legitimacy is heavily under-utilized. For example, common word embeddings (from Word2Vec [38], Speech2Vec [12] to BERT [14]), despite their dominating performance in various NLP tasks, are hard to be utilized under existing rescoring frameworks since they have a limited ability in mapping embeddings into ranking scores.

To alleviate the aforementioned issues, we propose a simple yet effective *Learning-to-Rescore* (L2RS) mechanism, which, as a first, tackles the *N*-best list rescoring problem from a Information Retrival (IR) perspective. In particular, L2RS directly predicts the partial orders of the *N*-best list regarding a specific acoustic input. What is more, L2RS utilizes a wide range of features with automatically optimized weights to rank the *N*-best lists for ASR and selects the most promising one as the final recognition result. The efficacy of L2RS relies on the design of the features, as such, we construct a comprehensive set of features using BERT, topic modeling, DNN-based LMs such as RNNLM [36] and BERT LM [14], together with an AM. By combining all these features, L2RS learns a rescoring model using the ranking algorithms such as RankSVM [7] [33]. Since each feature reflects one perspective from the linguistic and semantic legitimacy of the *N*-best hypotheses, L2RS achieves superior performance by ensembling the information from all these evaluation metrics.

To sum up, the main contributions of this paper are threefold:

- To the best of our knowledge, this is the first work in literature that reframes the ASR rescoring problem in new ways using information retrieval (IR) ranking approaches. We expect L2RS will inspire more research that marries IR with ASR, and lead to further work across the various communities.
- We propose a novel L2RS framework dedicated to ASR, which can easily incorporate various state-of-the-art NLP models to extract features. We systematically explore the effectiveness of these features and their combinations, and most of the features, such as BERT sentence embeddings, are also used in *N*-best list rescoring. They have proven to be quite promising compared to traditional features such as RNNLM, which is the mainstream one used in Kaldi toolkit [41].
- We conduct extensive experiments based on a public dataset, and experimental results show that L2RS can significantly outperform not only traditional rescoring methods but also improve its deep neural network counterparts, such as RNNLM and EC-Model by up to 20.85% in terms of NDCG@10. What is more, we implement an L2RS toolkit which can be integrated into the open-sourced Kaldi toolkit [41], and has currently the L2RS SDK has been successfully adopted for many online commercial services of WeBank Co., Ltd,[2] China's first digital bank, initiated by Tencent. The efficacy and applicability of L2RS is validated by real-life online customer datasets.

The rest of this paper is organized as follows: we first introduce the related work in Section 2. Then we introduce the details of our proposed L2RS models in Section 3. Experimental results are presented in Section 4, followed by a description of our industrial practice of L2RS Toolkit in Section 5. Finally we conclude the work in Section 6.

2 RELATED WORK

The present work is closely related to research fields of *N*-best hypothesis lists rescoring for ASR and Learn-to-Rank (LTR). In this sections, we survey the most related works from both fields.

2.1 *N*-best Hypothesis Lists Rescoring for ASR

The *N*-best hypothesis list rescoring is a fundamental problem in NLP applications such as ASR and machine translation. For many years, the back-off *n*-gram LMs have predominantly been used in ASR systems due to its reliability [3]. However, the *n*-gram LMs are rather simplistic and heavily limited in its ability to model language context, such as long-range dependencies. In order to alleviate this problem of the n-gram LMs, the mechanism of *N*-best list rescoring has been proposed and proven to be effective to significantly improve the ASR performance. For example, discriminative LMs (DLMs) [39, 42] and [43] are developed which can ensemble various features (e.g.ASR error during the model training period) to build discriminative models to distinguish positive instances from the negative ones. Language model adaptation (LMA) is another mechanism which modifies the original LMs with the first-pass or recent decoding results, either using caching [25] or topic modeling [11], and has been proved to be effective in relieving the mismatch between the training domain and the prediction domain.

With the superior performance of the DNN-based models in the NLP area, RNNLMs [36, 37] and LSTM-based RNNLMs [15] have also been heavily studied in ASR. However, RNNLMs cannot be compiled into a static decoding graph since they theoretically encodes infinite history [56], hence, the most common approach of taking advantage of RNNLMs is to apply them in the *N*-best rescoring step. As rescoring models, RNNLMs usually perform much better than the traditional *n*-gram LMs since DNNs have superior capabilities in capturing both short and long-range dependencies between words in human language. Other DNN-based rescoring models include the EC-Model which aims to train a classifier to find the best hypothesis from *N*-best list [40]. Neural Speech-to-Text LM (NS2TLM) [49] further extends a vanilla RNNLM by encoding the speech information from the acoustic feature sequence together with the original historical textual information to predict the probability of the next word.

[2] https://www.webank.com/

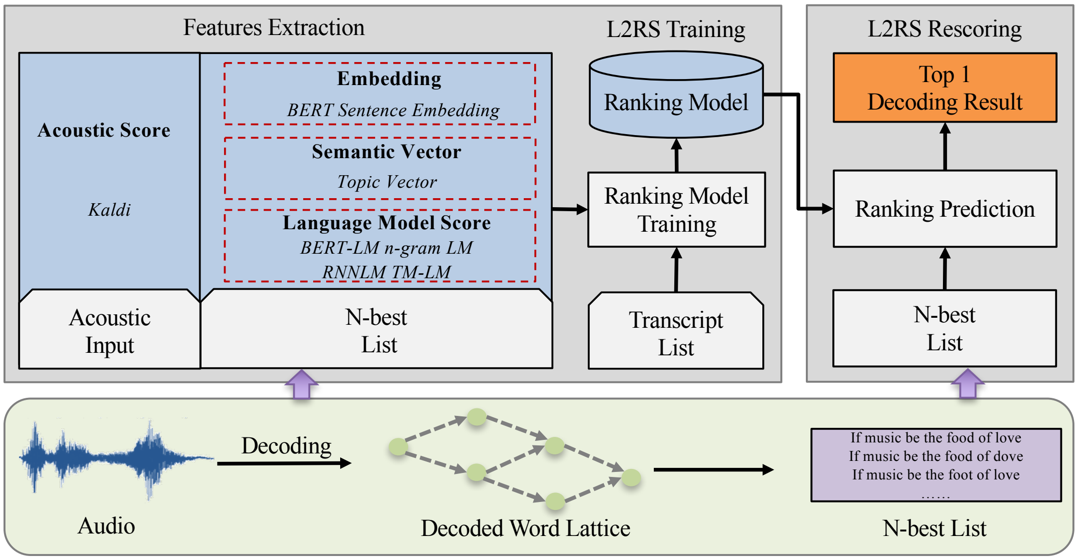

Figure 2: Flowchart of the proposed L2RS for ASR

Unfortunately, all above-mentioned approaches tackle the N-best rescoring problem in a two-step fashion, that is, predicting accurate grammatical legitimacy scores first and then adding these scores to formulate the final order. Our work differs from such approaches since L2RS formalizes the N-best list rescoring problem as a Learning-to-Score problem for ASR and directly predicts their partial orders for a specific acoustic input.

2.2 Learning To Rank Methods

Ranking is a central problem in database and IR applications such as document retrieval [33], recommendation [52], advertising [23] and so on. Traditional models such as boolean models [27], vector models [5] and probabilistic models [18] usually suffer the problem of high parameter tuning cost since they usually involve dozens of empirically-tuned features. Later machine learning techniques have been applied in this area for learning feasible models with automatically tuned parameters. This approach of learning a ranking model for IR is referred to as the Learning-to-Rank (LTR) approach. According to [33], there are three basic categories of LTR approaches: pointwise, pairwise, and listwise.

In the pointwise approaches, the algorithm considers each document of the ranking list in the training dataset as an independent instance, and the model tries to predict a relevant score indicating how relevant the document is with respect to the current query. The final ranking list is given by sorting the list according to these relevance scores. In this case, the ranking problem is actually transformed into a classification, regression or ordinal classification problem. The existing regression and classification algorithms can be readily used as pointwise LTR approaches, resulting in algorithms such as Subset Ranking [13], McRank [31], OC SVM [46] and so on.

In the pairwise approaches, the algorithm looks at a pair of documents in the same ranking list each time for the loss function. The objective of the ranker is to construct the optimal ordering for each pair and to minimize the average number of inversions in the ranking. The ranking problem is actually transformed into a pairwise classification or pairwise regression problem. Some of the most popular pairwise LTR algorithms includes Ranking SVM [20], RankNet [6] and IR SVM [7]. In practice, the pairwise algorithms work much better than the pointwise algorithms since predicting the relative order of two documents in the same ranking list is much closer in nature to the ranking problem.

In the listwise approaches, the ranker directly considers the entire document list, instead of pairs, and tries to construct the optimal ordering for it. The algorithms either directly optimize the IR measures, such as NDCG, or indirectly optimize them by defining a loss function based on the properties of the ranking. Algorithms such as AdaRank [57] and SoftRank [51] belong to the first category, and ListNet [8] and ListMLE [26] belong to the second. Compared with the above-mentioned pointwise or pairwise approaches, listwise approaches can be fairly complex, but they usually produce the best results among all three categories of methods.

Besides database and IR, LTR has also been widely used in areas such as bioinformatics [32], software engineering [47], and image tagging [45]. However, to the best of our knowledge, this is the first work that tackles the N-best rescoring problem using the LTR approach, and our industrial practice in the public TED dataset and a in-house real-life dataset both validate the effectiveness of this approach. We expect L2RS would inspire further research to improve the ASR systems from the IR perspective.

3 LEARNING-TO-RESCORE

In this section, we first give the definition of the L2RS problem, followed by the description of the textual and acoustic features designed for L2RS. Finally, we describe the details of the rescoring model in L2RS.

3.1 Problem Formulation

The pipeline of L2RS is listed in Figure. 2. Formally, the ASR system aims to find the optimal textual string $\mathbf{w}^*$ for a given acoustic input, denoted as $\mathbf{a}$, by the following equation:

$$\mathbf{W} \propto \arg\max(\log P_{LM}(\mathbf{w}) + \log P_{AM}(\mathbf{a}|\mathbf{w})), \tag{1}$$

$$\mathbf{w}^* = \arg \max_{j \in [1,N] \& \mathbf{w}_j \in \mathbf{W}} f(\phi(\mathbf{a}, \mathbf{w_j})), \tag{2}$$

where P_{LM} represents a back-off n-gram LM, P_{AM} is an AM, $\phi(\mathbf{a}, \mathbf{w})$ is the feature-vector representation of pair $(\mathbf{a}, \mathbf{w})$ that includes textual features as well as acoustic features, $\mathbf{W}$ is the N-best list, and $f(\cdot)$ is the rescoring function learned by L2RS approaches. The third component, $f(\phi(\cdot))$, is our contribution in this paper. It reframes ASR rescoring problem in new ways using IR ranking approaches opening many research opportunities.

L2RS learns $f(\phi(\mathbf{a}, \mathbf{w_j}))$ through a Learning-to-Rescore approach, which involves three steps: feature extraction, model training and rescoring. During the L2RS training period, the ASR system generates the N-best lists, denoted as $\mathbf{W} = (\mathbf{w_1}, \cdots, \mathbf{w_j}, \cdots, \mathbf{w_N})$, $j \in [1, N]$, and their order list $(r_1, \cdots, r_N)$ is decided based on the word error rate (WER) [24] of each hypothesis with the ground truth transcript. This composes the training dataset $\mathcal{D} = (\mathbf{X}, \mathbf{Y}) = \{(\mathbf{x_i}, \mathbf{y_i}), i \in [1, M]\}$ used for L2RS, where $\mathbf{x_i} = (\phi(\mathbf{a_i}, \mathbf{w_{i,1}}), \cdots, \phi(\mathbf{a_i}, \mathbf{w_{i,N}}))$, $\mathbf{y_i} = (r_{i,1}, \cdots, r_{i,N})$ and M is the size of the training dataset. The feature extraction is conducted on this training dataset and then the the LTR model is trained on it.

3.2 Textual Features

The textual features used in L2RS are from the lexical level to the semantic level and fall in six categories: n-gram LM, BERT sentence embedding, BERT LM, Probabilistic Topic Model LM, topic vector and RNNLM.

***n*-gram LM** The n-gram LMs are also used in our framework since they are predominantly used in ASR systems due to their simplicity and reliability. In L2RS, we use trigram LM trained using the transcript corpus with the SRILM[3] toolkit.

BERT Sentence Embedding BERT, or Bidirectional Encoder Representations from Transformers [14], is a powerful new language representation model proposed by Google which obtains the state-of-the-art results on various NLP tasks. The goal of BERT sentence embedding is to represent a variable length N-best hypothesis into a fixed-length vector, e.g. "*hello, nice to meet you*" to $[0.1, 0.3, 0.5, ..., 0.3]$ as shown in Figure 3. Each element of this vector represents the semantics of the original sentence, and the vector is further used in L2RS as a representation for each N-best hypothesis.

BERT LM BERT can also be used as an LM [53] to evaluate the quality of the N-best hypotheses from the linguistic perspective. In L2RS, we use the perplexity given by a fine-tuned BERT model as a feature of the N-best hypotheses. In L2RS, we first set up a pretrained BERT model[4], and then conduct fine-tuning using the transcript corpus. Finally, we use the fine-tuned BERT model to evaluate the perplexity of the N-best candidate.

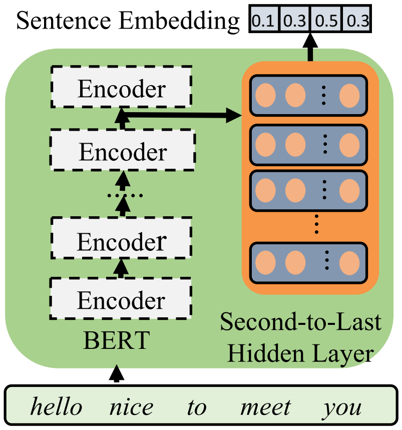

Figure 3: BERT Sentence Embedding

Probabilistic Topic Model LM Topic modeling, such as LDA [4] and SentenceLDA [2], has the ability to capture the semantic coherence of the N-best hypotheses. These topic models are previously used in the LM adaptation mechanism [11] to custom the n-gram LM with recently decoded words, and have proven to be effective in improving the final ASR performance. In L2RS, we choose the LDA model since it is widely used by the industrial community and has many open-sourced toolkits such as LightLDA [58] and Familia [22]. We first train an LDA topic model based on the transcript corpus, which produces the topic-word distribution φ_{kw} ($k \in [1, K]$ is the topic index and $w \in [1, W]$ is the word index). Next, we use the trained model to obtain the topic mixing proportion vector $\theta_{\mathbf{w_j}}$ of each hypothesis $\mathbf{w_j}$, which represents the semantic meaning of this hypothesis. Based on these two parameters, we compute a transcript-specific unigram LM by.

$$p(\mathbf{w_j}|\theta_{\mathbf{w_j}}) = \sum_{k \in K} \theta_{\mathbf{w_j}k} \varphi_{kw}. \tag{3}$$

Topic Vector Similar to Topic Model LM, L2RS directly uses the trained topic model to infer the N-best hypotheses' topic mixing proportion vector θ, and this vector is used as a topic representation for each N-best hypothesis.

Neural Network-based LM Neural network-based LMs are proven to be effective for N-best list rescoring in ASR systems. We train a RNNLM [36] with the transcript corpus, and the perplexity of each hypothesis given by the RNNLM acts as a feature reflecting the quality of the hypothesis. Compared with the n-gram LMs, RNNLMs have the ability to process a word sequence with arbitrary length and learn to store past information in the hidden states.

3.3 Acoustic Feature

The acoustic feature used in L2RS is the acoustic score given by the acoustic model. Specifically, in L2RS, we train a "chain" model based on the training data using the Kaldi[5] toolkit. It should be noted that other features such as speech embedding produced by Speech2Vec [12] can also be used.

[3] http://www.speech.sri.com/projects/srilm/

[4] https://github.com/google-research/bert

[5] https://github.com/kaldi-asr/kaldi

3.4 Rescoring Model

In this section, we give a brief introduction of the LTR models we used for L2RS.

3.4.1 Pointwise approaches. Since the pointwise approach converts the ranking problem into a classification or regression problem, existing regression and classification algorithms can be readily used in our framework. In L2RS, we adopt Multiple Additive Regression Trees (MART) [17, 31, 48] as the rescoring model. A decision tree can be interpreted as the parameter space partitioned into disjoint regions R_u, where U is the number of leaves and $u \in \{1, \ldots, U\}$. Then the regression tree gives the prediction value for each region by.

$$T(\mathbf{x}; \Phi) = \sum_{u=1}^{U} \gamma_u \mathbb{I}(\mathbf{x} \in R_u), \tag{4}$$

where $\Phi = \{R_u, r_h\}_1^U$ is the model parameters, $\mathbb{I}$ is an indicator function having the value 1 if the argument is true and 0 otherwise, and $\gamma_u = avg(y_i | x_i \in R_u)$. The empirical risk function for optimization is:

$$\widehat{\Theta} = \arg\min_{\Phi} \sum_{u=1}^{U} \sum_{\mathbf{x_i} \in R_u} L(y_i, \gamma_u). \tag{5}$$

MART tries to learn a ranking function $F(\mathbf{x})$ by minimizing the loss function:

$$F^*(\mathbf{x}) = \arg\min_{F(\mathbf{x})} E_{\mathbf{x},y}[L(y, F(\mathbf{x}))], \tag{6}$$

where $F(\mathbf{x})$ is a weighted combination of a series of decision trees.

3.4.2 Pairwise approaches. We choose Ranking SVM to train a rescoring model, and the learning of RankSVM is formalized as the following quadratic programming problem:

$$\begin{aligned} &\min_{\alpha,\varepsilon} \frac{1}{2} \|\alpha\|^2 + \frac{C}{M} \sum_i \varepsilon_i \\ &\text{s.t. } y_i(\alpha, \phi(\mathbf{a}, \mathbf{w_p}) - \phi(\mathbf{a}, \mathbf{w_q})) \geq 1 - \varepsilon_i \\ &\qquad \varepsilon_i \geq 0, i = 1, \cdots, M, \end{aligned} \tag{7}$$

where $\mathbf{w}_p$ and $\mathbf{w}_q$ are two hypotheses from the same N-best list, $\|\cdot\|$ denotes L_2 norm, m denotes the number of training instances, α is the model parameters in the feature space, and $C > 0$ is a coefficient.

3.4.3 Listwise approaches. We use ListNet [8] for the listwise approach, and the total loss with respect to the training data to be minimized is defined as:

$$\sum_{i=1}^{M} L(\mathbf{y_i}, \mathbf{z_i}), \tag{8}$$

where $\mathbf{y_i}$ is the ground truth ranking list of hypotheses with respect to ith input, and $\mathbf{z_i}$ is the ranking list generated by current rescoring algorithm. Suppose $\mathbf{y_i} = (r_{i,1}, \ldots, r_{i,N})$ and $\mathbf{z_i} = (f(x_{i,1}), \ldots, f(x_{i,N}))$, Listwise uses probabilistic methods to represent the difference between this two lists. The first one is to calculate a permutation probability by:

$$p_s(\pi) = \prod_{j=1}^{N} \frac{\varphi(s_{\pi(j)})}{\sum_{k=j}^{N} \varphi(s_{\pi(k)})}, \tag{9}$$

where $s_{\pi(j)})$ is the score of the hypothesis at position j of the permutation and φ is an increasing and strictly positive function such as the exponential function $\exp(\cdot)$. Since the total number of permuation is $n!$, ListNet propose to use top one probability to approximate it by,

$$p_s(j) = \frac{\varphi(s_{\pi(j)})}{\sum_{k=1}^{N} \varphi(s_{\pi(k)})}, \tag{10}$$

where s_j is the score of the hypothesis j in the list. The ListNet trains a neural network and use the gradient descendant to optimize the parameters w. Suppose we use exponential function as the φ function and the top one probability can be written as:

$$p_{\mathbf{z_i}(f_w)}(x_{i,j}) = \frac{\exp(f_w(x_{i,j}))}{\sum_{k=1}^{N} \exp(f_w(x_{i,k}))}. \tag{11}$$

In this case, the gradient of the loss function $L(\mathbf{y_i}, \mathbf{z_i})$ with respect to the parameter w is as follow,

$$\begin{aligned} \Delta w = &- \sum_{j=1}^{N} p_{\mathbf{y_i}}(x_{i,j}) \frac{\partial f_w(x_{i,j})}{\partial w} \\ &+ \frac{1}{\sum_{j=1}^{N} \exp(f_w(x_{i,j}))} \sum_{j=1}^{N} \exp(f_w(x_{i,j})) \frac{\partial f_w(x_{i,j})}{\partial w}. \end{aligned} \tag{12}$$

4 EXPERIMENTS

In this section, we conduct experiments on a public dataset to verify the effectiveness of the proposed model. In Section 4.1, we describe the experimental setup. Then we introduce the comparison baselines in Section 4.2 and the two evaluation measurements, Normalized Discount Cumulative Gain (NDCG) and Word Error Rate (WER) in Section 4.3. Finally, we present the experimental results in terms of these two measurements and also give a detailed analysis of the quality of the features.

4.1 Experimental Setup

We use the public TED-LIUM dataset[6] [44] in our experiment with the statistics listed in Table 1. For RankSVM,[7] the parameters C is set to 10. For BERT, we first set up a pretrained BERT model,[8] and then conduct fine-tuning using the transcript corpus of the training dataset. The dimension of the BERT sentence embedding is set to 1024 using a method similar to the BERT-as-service toolkit [55]. For topic modeling, we use LightLDA [58] and the number of topics is set to 50. Following [34], we obtain the 50-best for each utterances in the dataset. All experiments were conducted on a server with a 314 GB memory, 72 Intel Core Processor (Xeon), Tesla K80 GPU and CentOS.

4.2 Baselines

Several strong baselines for N-best rescoring have been implemented for comparison with the proposed L2RS model.

- **Basic *n*-gram LM**: this baseline is the classical one used in the Kaldi toolkit. In our experiment, the N-best list is rescored using a weighted addition of the AM score and a Tri-gram LM score with default weights 1.0 and 0.1.

[6] http://www.openslr.org/7/

[7] https://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

[8] https://github.com/google-research/bert

Table 1: 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		
Language	English		

- **Cache Model**: the Cache model baseline saves the recent decoded words in a cache and boosts the probability of these words [30] in the following rescoring period.
- **Trigger-based DLM**: This is a classical DLM [42] based on statistical machine learning, which trains a classier to distinguish the best candidates with the others.
- **Topic Modeling-based LMA**: this baseline uses topic modeling to infer the topic distribution of the first-pass decoding results, and then boosts the probability of the words belongs to these topics [11]. In this baseline, we train a LDA model using LightLDA [58] with the number of topics, K, set to 50.
- **RNNLM**: this baseline integrates a RNNLM to rescore the N-best list [30], and it is also supported by the Kaldi toolkit.
- **EC-Model**: this baseline trains a classifier to distinguish the best candidate from the others [40], and it includes an DNN-based encoder to encode each N-best candidate.
- **NS2TLM**: this baseline extends the RNNLM by incorporating speech signals into the hidden state to decode the next words [49].
- **L2RS**: This is our proposed learning-to-rescore approach, which includes pointwise, pairwise and listwise methods.

To ensure fairness in the performance comparison, we construct all the models and methods on the same training set, and then evaluated and reported the results on the same development and testing set. Methods such as EC-Model, NS2TLM have all been recently proposed based on the state-of-the-art neural networks, and they are very strong baselines for performance comparison.

4.3 Evaluation Measurements

We adopt two widely-used measurements to evaluate the performance of the proposed L2RS framework.

- NDCG@n: NDCG@n reflects the quality of a ranking list by measuring its top-n elements. In some extreme scenarios such as ASR in noisy environments or casual-style speech, the ASR systems usually need to produce multiple recognition hypotheses [35, 40], rather than only the 1-best from the N-best list. For a N-best list $\mathbf{W} = (\mathbf{w_1}, \cdots, \mathbf{w_j}, \cdots, \mathbf{w_N})$ concerning a given acoustic input $\mathbf{a}$, we use y_j to represent the ground-truth relevance score of each hypothesis $\mathbf{w_j}$, and then the ground-truth ranking list $\mathbf{W}^g$ is obtained by sorting the hypotheses by their ground truth relevance scores. The $DCG@n$ score of the N-best list $\mathbf{W}$ is defined as:

$$DCG@n(\mathbf{W}) = \sum_{i=1}^{n} c(i)(2^{y_i} - 1), \tag{13}$$

where $c(i)$ is a rank-decaying function. In L2RS, we use a widely-used one which is defined as:

$$c(i) = \begin{cases} \frac{1}{\log(1+i)} & i < L \\ 0 & i \geq L \end{cases} \tag{14}$$

where L is the "truncation level" and it reflects the fact that that the quality of a list is mainly decided by the order of the top results. And the $NDCG@n$ score of the N-best list $\mathbf{W}$ is calculated by normalizing by the $DCG@n$ score of $\mathbf{W}^g$, that is,

$$NDCG@n(\mathbf{W}) = \frac{DCG@n(\mathbf{W})}{DCG@n(\mathbf{W}^{\mathrm{g}})}. \tag{15}$$

From above definition, we can see that the higher the $NDCG@n$ value the better the ranking list and the ASR system.
- WER: The word error rate is a standard metric to evaluate the performance of an ASR. It compares a reference to a hypothesis and is defined as follows,

$$WER = \frac{S + D + I}{Z}, \tag{16}$$

where S is the number of word substitutions, D is the number of word deletions, I is the number of word insertions and Z is the number of words in the reference. We can see from the definition that the lower the WER the better the performance of the ASR system.

4.4 Experimental Results

4.4.1 NDCG@10. Table 2 lists the rescoring performance of L2RS in terms of NDCG@10. In most cases, the ASR system finally delivers the 1-best result from the rescored N-best list. However, if an algorithm can produce a better ranking list (Top N), it will probably improve the final ASR results (Top 1). However, some tasks such as ASR in noisy environments or casual-style speech require multiple recognition hypotheses [35, 40], and in these cases, the correctly ordered ranking list becomes quite vital to the system's performance. From the results, we can see that compared with other methods, L2RS can produce a much better ranking list, which means not only is the top-1 result improved but the whole ranking list is also correctly ordered. By incorporating all the features, L2RS (Listwise) achieves up to 20.85% relative improvement over the AM+n-gram baseline. Since all other algorithms mainly optimize on the top-1 result instead of the whole ranking list, it is expected that their NDCG@10 score will be much worse than that of L2RS. We also notice that comparing all other methods besides RNNLM, NS2TLM performs a little bit better than LMA and the EC-Model, which is consistent with previous studies [49].

From the results, we can see that L2RS (Listwise) achieves the best performance among all three methods in the L2RS framework, which is also consistent with previous studies [8], where the listwise model was shown to be more effective for ranking than the pairwise model [9]. L2RS provides a framework where various LTR models can be incorporated to explore their effectiveness for the N-best rescoring problem in ASR.

4.4.2 WER. Since our ultimate goal is to improve ASR, we finally examine the effectiveness of the L2RS method in terms of WER, with the results listed in Table 3. The "Oracle" WER is computed using the best result from the N-best list by comparing it with the

Table 2: NDCG@10 of different N-best rescoring methods

Model	**Dev**	**Test**
Trigram LM	0.5931	0.5859
RNNLM	0.6455	0.5938
Trigger-based DLM	N/A	N/A
Cache Model	0.6186	0.5940
Topic Model LMA	0.6136	0.5944
EC-Model	0.6107	0.5943
NS2TLM	0.6495	0.6097
L2RS(Pointwise)	0.7263	0.6953
L2RS(Pairwise)	0.7430	0.7070
L2RS(Listwise)	**0.7498**	**0.7081**

ground truth transcript, and it is the theoretical ceiling performance of all the rescoring methods. Among all these methods, RNNLM, BERT-LM, Trigger-based DLM, Cache Model, EC-Model, NS2TLM, L2RS (Pointwise), L2RS (Pairwise) and L2RS (Listwise) have 0.509%, -0.083%, -1.036%, 0.026%, 0.204%, 0.506%, 2.084%, 2.448% and 2.502% improvement respectively over the baseline n-gram LM method in the test dataset. L2RS shows performance improvement over the state-of-the-art rescoring methods by a significant margin. The experimental results validate that by incorporating more valuable features from the state-of-the-art NLP models, L2RS can benefit current ASR systems. From the result, we can see that L2RS (Listwise) achieves the lowest WER among all three methods in the L2RS framework.

Table 3: WER of different N-best rescoring methods

Model	**Dev**	**Test**
Trigram LM	21.999%	27.084%
RNNLM	21.490%	27.040%
Trigger-based DLM	23.303%	28.120%
Cache Model	21.925%	27.058%
Topic Model LMA	22.044%	27.091%
EC-Model	21.706%	26.880%
NS2TLM	21.200%	26.578%
L2RS(Pointwise)	20.328%	25.000%
L2RS(Pairwise)	19.924%	24.636%
L2RS(Listwise)	**19.659%**	**24.582%**
Oracle	16.538%	19.196%

4.4.3 Quantitative Analysis of Features. We use each dimension of all the features described in Section 3.2 to train an L2RS (Pairwise) model and take the NDCG@10 on the test set as a measure to reflect the quality of these features, which is a common method in the IR area to evaluate the quality of a feature for ranking [19]. The results are listed in Figure 4, with the x-axis representing the feature category and the y-axis representing their NDCG values. We can see that besides traditional AM and LM scores, other features also provide valuable information from different linguistic and semantic perspectives. Since the traditional rescoring pipeline usually combines the LM score and the AM score with an empirically tuned weight to formulate the final ranking score, features such as BERT sentence embedding are hard to use for the rescoring pipeline. However, L2RS provides a flexible mechanism to make full use of these *embedding* representations and their combinations, and the experimental results validate that BERT sentence embedding is even more effective than the RNNLM score.

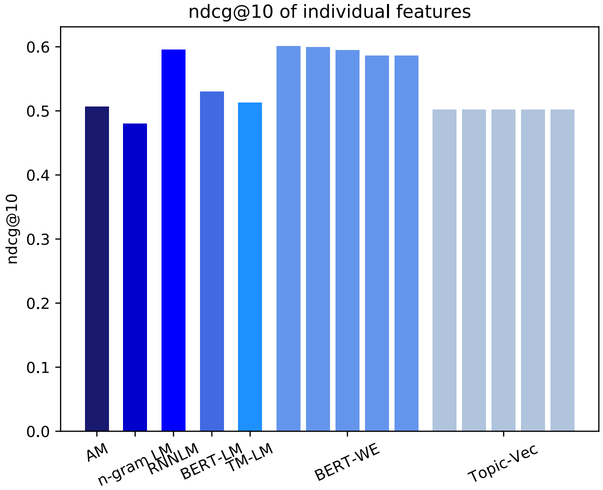

Figure 4: Quality (NDCG@10) of Individual Features

5 THE L2RS TOOLKIT AND INDUSTRIAL PRACTICE

We have successfully deployed the L2RS mechanism for various online commercial services in WeBank Co., Ltd. In this section, we briefly introduce the L2RS toolkit developed for our online commercial services. We start by describing the structure of the toolkit, and then provide some benchmark results on a real-world customer service dataset.

5.1 Toolkit Architecture

As shown in Figure 5, our ASR SDK is based on the Kaldi toolkit. Kaldi is an open-source toolkit for speech recognition written in C++ and and is licensed under the Apache License v2.0. Our ASR SDK is also implemented in C++ and includes various modules: *Online ASR*, *Offline ASR*, *Language Model Adaptation (LMA) Toolkit* and the *L2RS* Toolkit. Dozens of users have been supported using the this ASR SDK with L2RS integrated as the rescoring method for our online services.

5.1.1 LMA Toolkit. The LMA toolkit supports common LMA methods, including Cache, PLSA [21], LDA [4], WVM [11] and so on. For each kind of topic model, the LMA toolkit supports the basic functions: topic model training, topic model-based unigram construction and LM interpolation.

The LMA toolkit uses the topics discovered by topic models to interpolate the basic n-gram LM. After the ASR system generates the first-pass decoding result d, the LMA toolkit will infer its topic distribution θ_d. Together with the topic-word distribution φ_{kw} from a trained topic model, the LMA will get a topic model based unigram model by $P_{TM}(w|\theta_d) = \sum_{k \in K} \varphi_{kw}\theta_{dk}$ and adapt the basic n-gram LM as follows:

$$P_d(w|C) = \lambda P_{TM}(w|\theta_d) + (1-\lambda)P_{LM}(w|C), \quad (17)$$

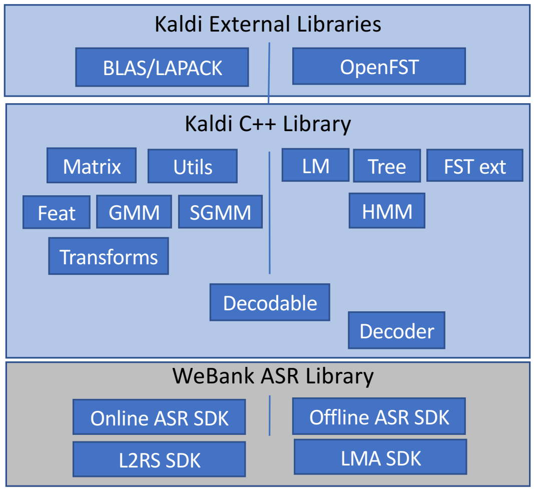

Figure 5: A simplified view of the WeBank ASR Toolkit. The WeBank ASR Toolkit is based on the core libraries of the Kaldi toolkit and supports various LMA methods as well as the L2RS method for N-best Rescoring

where λ is a trade-off parameter, C is the context information on sounding words, and $P_{LM}(w|C)$ is the probability given by the basic n-gram LM. The adapted LM is further utilized for rescoring the N-best lists.

5.1.2 L2RS Toolkit. The L2RS Toolkit supports several modules, mostly for the basic rescoring operations: *feature extraction, training the rescoring model and rescoring*. For evaluation, both the NDCG@n as well as the WER can be computed on some test data. For feature extraction, besides the features mentioned above, we also include the reverse LM score, which is also a common metric for linguistic evaluation. The Chinese version BERT [9] is incorporated to extract embeddings for sentence. For the LTR model, we implement the Ranking SVM algorithm for the pairwise approach due to the consideration of balance between effectiveness and model complexity.

5.2 Performance Analysis

We also evaluate the system performance on a real-world customer service dataset we collected. The dataset contains around 8000 hours of conversational speech collected from real-life online customer service scenarios in Mandarin Chinese. Compared with the public standard datasets in ASR, this real-life dataset contains *diverse* and *noisy* speech with *different background noise*, which raises more challenges to our ASR systems. We remove all personally-identifiable information (such as name) due to privacy concerns, and then reserve 80% of the data for the training set and split the rest for development (10%) and testing (10%). The training set is used to train the AM, LMs, and Ranking SVM model, while the development set is reserved to tune the parameters. We utilize a Kaldi "chain" model [41] for the AM, and a trigram LM using the SRILM toolkit for the back-off n-gram LM. For the N-best list in each utterance, the parameter N is also set to 50 according to [34].

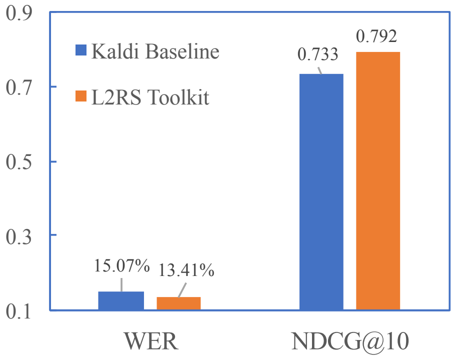

Figure 6: Performance of the L2RS Toolkit on the Real-life Custom Service Dataset

5.2.1 NDCG@10. We show the NDCG@10 value in Figure. 6. From the result, we can see again that the L2RS rescoring approach improves the NDCG@10 from the AM+n-gram baseline 0.733 to 0.792. By incorporating all these features, L2RS finally achieves up to 7.45% relative improvement over the AM+n-gram baseline.

5.2.2 WER. We also examine the effectiveness of L2RS SDK in terms of WER, and the result is also listed in Figure 6. From the results, we can see that the L2RS SDK reduces the final ASR error over the baseline method by a significant margin (1.657% absolute WER reduction). The experiment results prove that by incorporating valuable features from the state-of-the-art NLP models, L2RS can boost the performance of current ASR systems. L2RS provides a new rescoring framework under which various acoustic and grammatical features can easily be incorporated into the traditional ASR pipeline.

6 CONCLUSION

In this paper, we propose a simple yet effective Learning-to-Rescore mechanism for hybrid ASR, which reframes the ASR rescoring problem in new ways using IR ranking approaches opening many research opportunities. Under this mechanism, comprehensive features from advanced NLP models, with automatically optimized weights can be used to form a rescoring model. Experimental results on a public dataset as well as an in-house real-life dataset have both indicated that L2RS is quite effective for N-best list rescoring.

Following the perspective of treating the rescoring problem as an IR ranking problem, there are dozens of further works to be done. In particular, it would be interesting to explore the performance of neural ranking models designed for L2RS. Our work shows that directly optimizing the ranking order of the N-best list is quite helpful for reducing the final ASR decoding error. Following this line, a more comprehensive study that involves end-to-end neural ranking methods can be conducted to confirm the generality of the framework. It would also be helpful to carry out the work of developing a multi-modal Siamese network that directly captures the semantic similarity between the acoustic signal and the N-best hypotheses for N-best rescoring.

Acknowledgement. We are grateful to the anonymous reviewers for their constructive comments on this paper. The research is supported by GZSTI16EG24.

[9] https://github.com/google-research/bert

REFERENCES

[1] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. 2016. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*. 173–182.

[2] Georgios Balikas, Massih-Reza Amini, and Marianne Clausel. 2016. On a topic model for sentences. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 921–924.

[3] Jerome R Bellegarda. 2004. Statistical language model adaptation: review and perspectives. *Speech communication* 42, 1 (2004), 93–108.

[4] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[5] Abraham Bookstein. 1982. Explanation and generalization of vector models in information retrieval. In *International Conference on Research and Development in Information Retrieval*. Springer, 118–132.

[6] Christopher Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine learning (ICML-05)*. 89–96.

[7] Yunbo Cao, Jun Xu, Tie-Yan Liu, Hang Li, Yalou Huang, and Hsiao-Wuen Hon. 2006. Adapting ranking SVM to document retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 186–193.

[8] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*. ACM, 129–136.

[9] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to Rank: From Pairwise Approach to Listwise Approach. In *Proceedings of the 24th International Conference on Machine Learning (ICML '07)*. Association for Computing Machinery, New York, NY, USA, 129–136.

[10] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 4960–4964.

[11] Kuan-Yu Chen, Hsuan-Sheng Chiu, and Berlin Chen. 2010. Latent topic modeling of word vicinity information for speech recognition. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 5394–5397.

[12] Yu-An Chung and James Glass. 2018. Speech2vec: A sequence-to-sequence framework for learning word embeddings from speech. *arXiv preprint arXiv:1803.08976* (2018).

[13] David Cossock and Tong Zhang. 2006. Subset ranking using regression. In *International Conference on Computational Learning Theory*. Springer, 605–619.

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 4171–4186.

[15] Hakan Erdogan, Tomoki Hayashi, John R Hershey, Takaaki Hori, Chiori Hori, Wei-Ning Hsu, Suyoun Kim, Jonathan Le Roux, Zhong Meng, and Shinji Watanabe. 2016. Multi-channel speech recognition: LSTMs all the way through. In *CHiME-4 workshop*. 1–4.

[16] Jonathan T Foote. 1997. Content-based retrieval of music and audio. In *Multimedia Storage and Archiving Systems II*, Vol. 3229. International Society for Optics and Photonics, 138–147.

[17] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001), 1189–1232.

[18] Norbert Fuhr. 1992. Probabilistic models in information retrieval. *The computer journal* 35, 3 (1992), 243–255.

[19] Xiubo Geng, Tie-Yan Liu, Tao Qin, and Hang Li. 2007. Feature selection for ranking. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 407–414.

[20] Ralf Herbrich. 2000. Large margin rank boundaries for ordinal regression. *Advances in large margin classifiers* (2000), 115–132.

[21] Thomas Hofmann. 1999. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 289–296.

[22] Di Jiang, Yuanfeng Song, Rongzhong Lian, Siqi Bao, Jinhua Peng, Huang He, Hua Wu, Chen Zhang, and Lei Chen. 2021. Familia: A Configurable Topic Modeling Framework for Industrial Text Engineering. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2021), 516.

[23] Maryam Karimzadehgan, Wei Li, Ruofei Zhang, and Jianchang Mao. 2011. A stochastic learning-to-rank algorithm and its application to contextual advertising. In *Proceedings of the 20th international conference on World wide web*. ACM, 377–386.

[24] Dietrich Klakow and Jochen Peters. 2002. Testing the correlation of word error rate and perplexity. *Speech Communication* 38, 1-2 (2002), 19–28.

[25] Roland Kuhn and Renato De Mori. 1990. A cache-based natural language model for speech recognition. *IEEE transactions on pattern analysis and machine intelligence* 12, 6 (1990), 570–583.

[26] Yanyan Lan, Yadong Zhu, Jiafeng Guo, Shuzi Niu, and Xueqi Cheng. 2014. Position-Aware ListMLE: A Sequential Learning Process for Ranking.. In *UAI*. 449–458.

[27] Arash Habibi Lashkari, Fereshteh Mahdavi, and Vahid Ghomi. 2009. A boolean model in information retrieval for search engines. In *2009 International Conference on Information Management and Engineering*. IEEE, 385–389.

[28] Duc Le, Xiaohui Zhang, Weiyi Zheng, Christian Fügen, Geoffrey Zweig, and Michael L Seltzer. 2019. From senones to chenones: Tied context-dependent graphemes for hybrid speech recognition. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 457–464.

[29] Jinyu Li, Rui Zhao, Eric Sun, Jeremy HM Wong, Amit Das, Zhong Meng, and Yifan Gong. 2020. High-Accuracy and Low-Latency Speech Recognition with Two-Head Contextual Layer Trajectory LSTM Model. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 7699–7703.

[30] Ke Li, Hainan Xu, Yiming Wang, Daniel Povey, and Sanjeev Khudanpur. 2018. Recurrent neural network language model adaptation for conversational speech recognition. *INTERSPEECH, Hyderabad* (2018), 1–5.

[31] Ping Li, Qiang Wu, and Christopher J Burges. 2008. Mcrank: Learning to rank using multiple classification and gradient boosting. In *Advances in neural information processing systems*. 897–904.

[32] Bin Liu, Junjie Chen, and Xiaolong Wang. 2015. Application of learning to rank to protein remote homology detection. *Bioinformatics* 31, 21 (2015), 3492–3498.

[33] Tie-Yan Liu et al. 2009. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval* 3, 3 (2009), 225–331.

[34] Xunying Liu, Yongqiang Wang, Xie Chen, Mark J. F. Gales, and Phil Woodland. 2014. Efficient Lattice Rescoring Using Recurrent Neural Network Language Models. In *IEEE International Conference on Acoustics*.

[35] Lidia Mangu, Eric Brill, and Andreas Stolcke. 2000. Finding consensus in speech recognition: word error minimization and other applications of confusion networks. *Computer Speech & Language* 14, 4 (2000), 373–400.

[36] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*.

[37] Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 5528–5531.

[38] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[39] T. Oba, T. Hori, A. Nakamura, and A. Ito. 2012. Round-Robin Duel Discriminative Language Models. *IEEE Transactions on Audio, Speech, and Language Processing* 20, 4 (May 2012), 1244–1255. https://doi.org/10.1109/TASL.2011.2174225

[40] Atsunori Ogawa, Marc Delcroix, Shigeki Karita, and Tomohiro Nakatani. 2018. Rescoring N-Best Speech Recognition List Based on One-on-One Hypothesis Comparison Using Encoder-Classifier Model. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 6099–6103.

[41] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. 2011. The Kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society.

[42] Brian Roark, Murat Saraclar, and Michael Collins. 2007. Discriminative N-gram Language Modeling. *Comput. Speech Lang.* 21, 2 (April 2007), 373–392. https://doi.org/10.1016/j.csl.2006.06.006

[43] Brian Roark, Murat Saraclar, Michael Collins, and Mark Johnson. 2004. Discriminative language modeling with conditional random fields and the perceptron algorithm. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 47.

[44] Anthony Rousseau, Paul Deléglise, and Yannick Esteve. 2012. TED-LIUM: an Automatic Speech Recognition dedicated corpus.. In *LREC*. 125–129.

[45] Jorge Sánchez, Franco Luque, and Leandro Lichtensztein. 2018. A Structured Listwise Approach to Learning to Rank for Image Tagging. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 0–0.

[46] Amnon Shashua and Anat Levin. 2003. Ranking with large margin principle: Two approaches. In *Advances in neural information processing systems*. 961–968.

[47] Zhendong Shi, Jacky Keung, Kwabena Ebo Bennin, and Xingjun Zhang. 2018. Comparing learning to rank techniques in hybrid bug localization. *Applied Soft Computing* 62 (2018), 636–648.

[48] Krysta M Svore and CJ Burges. 2011. Large-scale learning to rank using boosted decision trees. *Scaling Up Machine Learning: Parallel and Distributed Approaches* 2 (2011), 2011.

[49] Tomohiro Tanaka, Ryo Masumura, Takafumi Moriya, and Yushi Aono. 2018. Neural Speech-to-Text Language Models for Rescoring Hypotheses of DNN-HMM Hybrid Automatic Speech Recognition Systems. In *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 196–200.

[50] Tomohiro Tanaka, Ryo Masumura, Takafumi Moriya, Takanobu Oba, and Yushi Aono. 2019. A Joint End-to-End and DNN-HMM Hybrid Automatic Speech Recognition System with Transferring Sharable Knowledge.. In *INTERSPEECH*. 2210–2214.

[51] Michael Taylor, John Guiver, Stephen Robertson, and Tom Minka. 2008. Softrank: optimizing non-smooth rank metrics. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*. 77–86.

[52] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. 2013. Deep content-based music recommendation. In *Advances in neural information processing systems*. 2643–2651.

[53] Alex Wang and Kyunghyun Cho. 2019. BERT has a mouth, and it must speak: BERT as a markov random field language model. *arXiv preprint arXiv:1902.04094* (2019).

[54] Yongqiang Wang, Abdelrahman Mohamed, Due Le, Chunxi Liu, Alex Xiao, Jay Mahadeokar, Hongzhao Huang, Andros Tjandra, Xiaohui Zhang, Frank Zhang, et al. 2020. Transformer-based acoustic modeling for hybrid speech recognition. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 6874–6878.

[55] Han Xiao. 2018. bert-as-service. https://github.com/hanxiao/bert-as-service.

[56] Hainan Xu, Tongfei Chen, Dongji Gao, Yiming Wang, Ke Li, Nagendra Goel, Yishay Carmiel, Daniel Povey, and Sanjeev Khudanpur. 2018. A pruned rnnlm lattice-rescoring algorithm for automatic speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 5929–5933.

[57] Jun Xu and Hang Li. 2007. Adarank: a boosting algorithm for information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 391–398.

[58] Jinhui Yuan, Fei Gao, Qirong Ho, Wei Dai, Jinliang Wei, Xun Zheng, Eric Po Xing, Tie-Yan Liu, and Wei-Ying Ma. 2015. Lightlda: Big topic models on modest computer clusters. In *Proceedings of the 24th International Conference on World Wide Web*. 1351–1361.