On semi-supervised LF-MMI training of acoustic models with limited data

Imran Sheikh, Emmanuel Vincent, Irina Illina

On Semi-Supervised LF-MMI Training of Acoustic Models with Limited Data

Imran Sheikh, Emmanuel Vincent, Irina Illina

Université de Lorraine, CNRS, Inria, Loria, F-54000 Nancy, France

{imran.sheikh,emmanuel.vincent,irina.illina}@inria.fr

Abstract

This work investigates semi-supervised training of acoustic models (AM) with the lattice-free maximum mutual information (LF-MMI) objective in practically relevant scenarios with a limited amount of labeled in-domain data. An error detection driven semi-supervised AM training approach is proposed, in which an error detector controls the hypothesized transcriptions or lattices used as LF-MMI training targets on additional unlabeled data. Under this approach, our first method uses a single error-tagged hypothesis whereas our second method uses a modified supervision lattice. These methods are evaluated and compared with existing semi-supervised AM training methods in three different matched or mismatched, limited data setups. Word error recovery rates of 28 to 89% are reported.

Index Terms: lattice-free MMI, semi-supervised training, speech recognition, error detection

1. Introduction

Voice-based personal assistants and user interfaces have gained popularity in the recent years thanks to improved automatic speech recognition (ASR) and spoken dialog technologies. ASR is now available in the form of online commercial services [1] as well as deployable open-source solutions [2]. However, poor performance due to domain mismatch still prevents many enterprises from building application-specific ASR systems. On top of that, many of these enterprises must deal with a limited amount of labeled in-domain data and sometimes even limited unlabeled data. Performance improvements from existing semi-supervised training approaches reduce with amount of in-domain data. Efficiently exploiting this limited amount of data is especially vital in early development stages and for under-resourced languages or privacy-critical applications.

State-of-the-art ASR systems employ a deep neural network based acoustic model (AM) trained with a sequence objective such as connectionist temporal classification (CTC) [3] or lattice-free maximum mutual information (LF-MMI) [4]. The LF-MMI approach requires a smaller amount of labeled data [5] but its performance starts degrading on small ($<$ 100 h) amounts of conversational speech data [4]. This approach has recently been extended to the semi-supervised setting [6], where so-called *supervision lattices* extracted from a large unlabeled dataset are used as training targets to improve the performance of a seed ASR model trained on a small labeled dataset. LF-MMI trained AMs have also been effective in transfer learning based domain adaptation, wherein an ASR model trained on a large out-of-domain dataset is adapted in a supervised [7] or semi-supervised way [8] to a smaller in-domain dataset. Semi-supervised LF-MMI based training has also been used for low-resourced languages [9]. Yet, these works have typically relied on hundreds of hours of in-domain data.

In this paper, we investigate semi-supervised training of AMs with the LF-MMI objective in practically relevant scenarios involving a limited amount of labeled and unlabeled in-domain data. We study both domain-mismatched and matched scenarios, wherein labeled and unlabeled data may come from different or same domains. Domain mismatch and weak initial models are expected to affect the accuracy of the supervision lattices on unlabeled data. To address this issue, we propose an error detection driven semi-supervised AM training approach, in which an error detector is used to modify the best-path hypothesis or the supervision lattice.

Guided LF-MMI training on unlabeled data has been recently attempted in different contexts. Fainberg et al. [10] proposed to merge the supervision lattices with noisy transcriptions (video subtitles) in a lightly supervised training setting. Unfortunately, such noisy transcriptions are unavailable in most practical cases. Tong et al. [11] addressed the bias towards low probability hypotheses in the supervision lattice by sampling hypotheses using dropout, which was shown earlier to correlate with AM uncertainty [12]. Our objective to promote the correct ASR hypotheses for unlabeled speech data is similar to Tong et al.'s, however we focus on limited data scenarios and we use an error detector to bias the supervision towards correct words in both matched and mismatched scenarios. Modern ASR error detectors [13–15] are more powerful than the lattice posterior-based confidence scores used to weight per-frame gradients [6] or to discard erroneous words [16, 17] or utterances [18, 19] in most semi-supervised neural AM training studies. Yet, to the best of our knowledge, they have not been leveraged for semi-supervised LF-MMI training so far. Our error detector exploits a range of acoustic and linguistic features extracted from the ASR confusion network, similar to [13].

The rest of the paper is organized as follows. Section 2 briefly introduces supervised and semi-supervised LF-MMI. Section 3 describes the proposed approach, including our error detection model and our proposed methods to obtain supervision for the unlabeled speech data. Section 4 discusses the experimental setup and evaluation of different semi-supervised training approaches, followed by a conclusion in Section 5.

2. Semi-supervised LF-MMI

Given a labeled dataset consisting of acoustic feature sequences O with reference transcripts W_{ref} and a language model (LM) L, the MMI objective for supervised training of an acoustic model A can be expressed as the sum over all utterances of

$$f_{\text{MMI}} \propto \log \frac{P_A(O|W_{\text{ref}})P_L(W_{\text{ref}})}{\sum_W P_A(O|W)P_L(W)} \quad (1)$$

$$= \log \frac{\sum_{\pi \in G_{\text{Num}}(W_{\text{ref}})} P(\pi)}{\sum_{\pi \in G_{\text{Den}}} P(\pi)} \quad (2)$$

where W in (1) spans all possible transcripts, which are typically approximated by a lattice decoded with a weak LM. LF-MMI training replaces the denominator lattice with a phone-level LM [4]. Expression (2) represents an equivalent finite state

transducer (FST) version of (1), wherein the sequence of states in the reference transcript forms the numerator graph G_{Num} and all possible state sequences of the phone-level LM form the denominator graph G_{Den}.

In the case of semi-supervised AM training, the reference transcripts W_{ref} for utterances in the unlabeled dataset are not available. Instead, training relies on a set of hypotheses H obtained using a seed ASR model trained on the labeled dataset. The objective function for these utterances becomes

$$f_{\text{MMI}} \propto \log \frac{\sum_{W \in H} P_A(O|W) P_L(W)}{\sum_W P_A(O|W) P_L(W)} \tag{3}$$

$$= \log \frac{\sum_{\pi \in G_{\text{Num}}(H)} P(\pi)}{\sum_{\pi \in G_{\text{Den}}} P(\pi)}. \tag{4}$$

It can be optimized using the same algorithm as (2), with the only difference that G_{Num} is bigger. See [6] for more details.

3. Proposed Approach

Semi-supervised LF-MMI has been shown to be effective with limited amounts of labeled training data [6]. However, domain mismatch and limited amounts of unlabeled speech data are expected to affect its performance. Mismatched vocabularies and biased LMs can affect the presence and the scores of the correct word sequences in the decoded lattices obtained from unlabeled data. We propose to use an explicit error detection model to control the supervision on the unlabeled speech data.

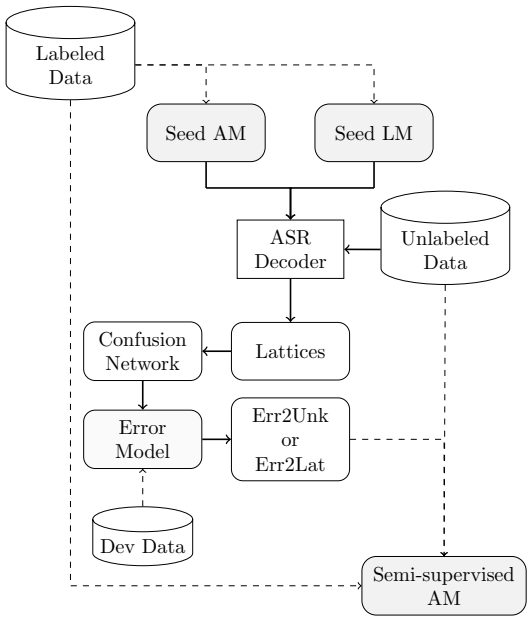

Figure 1: *Error detection driven semi-supervised AM training. Dashed arrows indicate 'use for training'.*

3.1. Error Detection Driven Semi-supervised AM Training

Figure 1 shows a block diagram of the proposed error detection driven semi-supervised AM training approach. In the first step, labeled training data containing accurate speech-transcript pairs are used to train the seed AM and LM. These seed models are used to decode the unlabeled speech data into ASR lattices. ASR confusion networks (a.k.a. sausages) are obtained from these lattices by minimum Bayes risk (MBR) decoding [20]. A pre-trained error detector, as will be discussed in Section 3.2, tags the confusion network bins into 3 classes: <error>, <no-error> and <eps>. These tags correspond to the error detector's hypotheses whether the best arc of a confusion bin is erroneous, correct, or a null arc, respectively. To exploit these tags for semi-supervised AM training, we propose two methods.

3.1.1. Err2Unk Semi-Supervised AM Training

The MBR estimate of the word sequence corresponding to a test utterance can be obtained by taking the most probable arc at each confusion bin [20]. A similar MBR hypothesis can be obtained for each unlabeled training utterance. However, we propose to use the tags from the error detector to: (a) retain the most probable arc of a confusion bin tagged as <no-error>, (b) remove the most probable arc of a confusion bin tagged as <eps>, and (c) replace the most probable arc of a confusion bin tagged as <error> with <unk> (which typically corresponds to a garbage phone or spoken noise). Hence, we refer to this method as *Err2Unk* semi-supervised training. Utterances which are left only with <unk> symbols are excluded from the training process. The set of Err2Unk transcriptions for the unlabeled speech data is merged with the labeled data and a new AM is trained using the standard supervised LF-MMI approach.

3.1.2. Err2Lat Semi-Supervised AM Training

Our Err2Unk method resorts to the <unk> symbol in place of words hypothesized to be erroneous by the error detector. Our second proposal, the *Err2Lat* method, falls back to the decoded lattice in the error regions. The main motivation is to exploit the uncertainty in the decoded lattices, similar to standard semi-supervised LF-MMI [6], but only in the hypothesized error regions. To do so, we combine the Err2Unk transcript with the original decoded lattice using an approach similar to Fainberg et al.'s [10]. However, we propose a crucial modification to their lattice combination method.

The lattice combination method in [10] takes a linear transducer R representing a noisy transcription and an ASR decoded lattice H projected on words with all weights scaled to zero. An edit transducer E that allows for insertions, deletions, and substitutions is used to compose them as:

$$T = R \circ E \circ H. \tag{5}$$

This is followed by (a) pruning, which retains the paths below a given cost, (b) projecting the pruned transducer on the output, and (c) epsilon removal, determinization, and minimization:

$$T = \min(\det(\text{rmeps}(\text{proj}(\text{prune}(R \circ E \circ H))))). \tag{6}$$

See [10] for further details on the edit transducer, pruning, and a visual example. It must be noted that this lattice combination method collapses to the word sequence in R when it is also present in the corresponding region of H. Hence, it cannot be used directly with Err2Unk transcripts, since it is likely to collapse back to <unk> in error regions.

To avoid this issue, we replace all <unk> arcs A_{UNK} introduced by the Err2Unk method in the Err2Unk transcript R_{E2U} by arcs A_{OOS} whose output symbol is not part of the symbol table of the LM graph G:

$$R = \text{replace}(R_{\text{E2U}}, A_{\text{UNK}}, A_{\text{OOS}}). \tag{7}$$

The modified transcript R is combined with H as in (6). The combined transducer T is composed with G to add LM costs,

and the resulting FST is used to compile a new training graph which is aligned to the speech utterance to add acoustic costs. The result of these operations is the supervision lattice for a given unlabeled speech utterance in our Err2Lat method.

3.2. Error Detection Model

The error detector is a neural network based classifier trained on features extracted from the ASR confusion network. We use the same feature set as the baseline features proposed in [13], except that we use a 3-gram LM to extract LM-related features due to limited LM training data. As compared to the feedforward neural network classifier in [13], we use a bidirectional long short-term memory network which can make use of the context in the sequence of confusion bins in an utterance.

The training data for this classifier must be disjoint from the training set used to train the ASR AM and LM. Indeed, the confusion statistics and errors made on that training set do not generalize to unseen utterances, especially in domain-mismatched scenarios. In the following, we demonstrate that it is feasible to effectively train the error detection model on the development set, i.e., no additional labeled data is required on top of the development set which is always needed.

4. Experiments and Evaluation

We evaluate the two proposed approaches for the task of AM training for conversational speech. In the following, we introduce the considered evaluation setups, describe the baselines and the other semi-supervised AM training approaches they are compared with, and discuss the results.

4.1. Small-Data and/or Domain-Mismatched Setups

We consider three limited data setups. The first one is a domain-mismatched scenario involving read vs. conversational speech. The two others are matched-domain scenarios involving human conversations and human-machine dialogs, respectively. Table 1 summarizes the datasets and splits used.

4.1.1. LS100-VM20: From Read to Conversational Speech

Recent speech data collection efforts by the open data community have lead to significant amounts of read speech in different languages [21, 22]. Yet, AMs trained on such read speech corpora show degraded performance on conversational speech encountered in most real life applications. Previous works have studied such scenarios [7, 8], albeit with several hundred hours of read and conversational speech data. For most languages, the conversational speech data available in the initial development stages is much smaller and it is mostly unlabeled. Not even all languages have such a large amount of read speech data to begin with, as evident from the statistics available from [22].

To assess this scenario, we select the standard *train-clean-100* subset of the English read speech corpus Librispeech [21] as our labeled out-of-domain training dataset, and 20 h of English conversational speech from the Verbmobil corpus [23] as our unlabeled in-domain training dataset. The development and test sets, which consist of 2 and 3 h of speech, respectively, are extracted from Verbmobil. Speakers and conversations do not overlap across the three subsets of the Verbmobil corpus. Conversations corresponding to speakers with non-US English accent are kept in the development and test sets to resemble real application scenarios. We refer to this setup as *LS100-VM20*.

4.1.2. VM5-VM20: Human-Human Conversations

Our second setup tackles the matched-domain scenario where both labeled and unlabeled training data belong to the Verbmobil conversational speech corpus. By contrast with [6], we consider a limited data scenario. Specifically, we consider 5 additional hours of the Verbmobil corpus as a labeled training dataset. The unlabeled training set, the development set, and the test set are same as for LS100-VM20. We refer to this setup as *VM5-VM20*. Again, speakers and conversations do not overlap across the four subsets of Verbmobil.

4.1.3. LG4-LG19:Human-Machine Dialog Utterances

We also analyze the matched-domain scenario wherein both labeled and unlabeled data are human utterances extracted from a human-machine dialog system. To do so, we use subsets of the Let's Go bus information system dataset [24], 1 year of which has been annotated and made available [25]. We use data collected in the first 15 days of October 2008 (4 h) as the labeled training dataset and data collected in the next two and a half months (19 h) as the unlabeled training dataset. The development and test sets consist of data collected in the first and the last 15 days of September 2009, respectively (6 h each). This setup features a larger variety of speakers within each subset compared to VM5-VM20. We refer to it as *LG4-LG19*.

Table 1: *Datasets and splits used in the three evaluation setups. LS = Librispeech, VM = Verbmobil, LG = Let's Go.*

	LS100-VM20	VM5-VM20	LG4-LG19
Train labeled	LS 100 h	VM 5 h	LG 4 h
Train unlabeled	VM 20 h	VM 20 h	LG 19 h
Development	VM 2 h	VM 2 h	LG 6 h
Test	VM 3 h	VM 3 h	LG 6 h

4.2. Baseline, Topline, and Compared AMs

We evaluate the proposed Err2Unk and Err2Lat semi-supervised AM training methods against the *seed AM* trained with LF-MMI on the labeled dataset only (baseline) and the *oracle AM* trained on both labeled and unlabeled data assuming the availaility of reference transcripts for the latter (topline). We also compare with the original semi-supervised LF-MMI approach in [6] and with semi-supervised training based on the *best path* obtained by the seed model. All approaches use the 3-gram *seed LM* trained on the labeled dataset, i.e., the LM is not retrained on unlabeled data.

We report the Word Error Rates (WERs) achieved on the development and test sets, which have been averaged over 3 trials for each approach in each setup. We also report the Relative WER Improvement (RWI) and the WER Recovery Rate (WRR) [6] on the test set, which are calculated as

$$\text{RWI} = \frac{\text{Seed AM WER} - \text{Semi-sup AM WER}}{\text{Seed AM WER}} \tag{8}$$

$$\text{WRR} = \frac{\text{Seed AM WER} - \text{Semi-sup AM WER}}{\text{Seed AM WER} - \text{Oracle AM WER}}. \tag{9}$$

The AMs use a time delay neural network (TDNN) architecture with splices {-2,-1,0,1,2} {-1,0,1} {-1,0,-1} {-3,0,3} {-3,0,3} {-6,-3,0} at each successive layer with 512 dimensions.

Inputs are 40 Mel-frequency cepstral coefficients and 100 dimensional online i-vectors [4, 26]. The i-vector extractor is trained on the combined labeled and unlabeled datasets.

ASR error detector's F1 scores, with learning entirely on the development set and evaluation on the unlabeled train set, for classes (<no-error>, <eps>, <error>) are (0.84, 0.80, 0.69), (0.87, 0.88, 0.51), and (0.88, 0.85, 0.65) for setups LS100-VM20, VM5-VM20, and LG4-LG19, respectively.

For the semi-supervised LF-MMI and Err2Lat methods, phone-levels LMs for creating the denominator FST are estimated from both labeled and unlabeled data. A weight of 3:2 is chosen for phone sequences from labeled vs. unlabeled data, without any tuning. Additionally, graph costs from the lattice are added with an LM scale of 0.5 [6]. Because the denominator FST resulting from the Err2Unk method may include too many <unk> phones and lead to increased deletion error, we remove <unk> arcs from the final HCLG decoding graph [27]. Removing <unk> from the HCLG decoding graph did not improve the WER for other compared approaches.

4.3. Results and Discussion

Table 2 reports the results in the LS100-VM20 setup, wherein labeled data is read speech (Librispeech 100 h) and unlabeled data is conversational speech (Verbmobil 20 h). In this domain-mismatched, limited data scenario, semi-supervised LF-MMI is outperformed by the simpler, best-path semi-supervised training approach. The proposed Err2Unk semi-supervised training approach achieves the lowest WER, while the proposed Err2Lat approach gives a (statistically significant) higher WER than Err2Unk on the test set. Err2Lat is statistically equivalent to best-path semi-supervision, further suggesting that alternative paths from lattice supervision may not be helping in this setup. More experiments are required to confirm this.

Table 2: *WER (%) achieved in the domain-mismatched read vs. conversational speech setup (LS100-VM20). (Best result, and ones statistically equivalent to it at p=0.05, are in bold font.)*

Type of Supervision	Dev WER	Test WER	Test RWI	Test WRR
seed (LS100)	41.07	40.95	-	-
Semi-sup best path	38.37	38.02	7.1%	27.3%
Semi-sup LF-MMI	39.13	39.34	3.9%	15.0%
Semi-sup Err2Unk	**38.08**	**37.62**	**8.1%**	**31.0%**
Semi-sup Err2Lat	**38.21**	37.97	7.3%	27.7%
oracle (LS100+VM20)	31.59	30.22	-	-

Table 3 presents the results in the domain-matched VM5-VM20 setup, wherein labeled and unlabeled data are both from the Verbmobil dataset. The baseline trained with just 5 h of labeled in-domain data outperforms the one trained with 100 h of out-of-domain data in Table 2. Furthermore, the relative improvements over the baseline achieved by all semi-supervised training methods, except best-path, are about twice bigger than above. In this setup, the proposed Err2Lat semi-supervised training method gives the lowest WER on par with classical semi-supervised LF-MMI. Note that the unlabeled training set and the development and test sets are identical in both tables, hence these results are directly comparable.

Finally, Table 4 shows the performance in the domain-matched LG4-LG19 setup, wherein labeled and unlabeled data both come from Let's Go. Err2Unk semi-supervised training gives the lowest WER and performs almost as well as the fully supervised topline, with an impressive WRR of 89.1%. Err2Lat gives a similar but slightly higher WER than Err2Unk, with the difference being statistically significant on the test set.

Table 3: *WER (%) achieved in the matched-domain human conversation setup (VM5-VM20). (Best result, and ones statistically equivalent to it at p=0.05, are highlighted in bold.)*

Type of Supervision	Dev WER	Test WER	Test RWI	Test WRR
seed (VM5)	37.95	37.84	-	-
Semi-sup best path	33.45	33.66	11.1%	36.4%
Semi-sup LF-MMI	30.90	**31.67**	**16.3%**	**53.7%**
Semi-sup Err2Unk	**30.65**	32.00	15.4%	50.8%
Semi-sup Err2Lat	**30.43**	**31.42**	**17.0%**	**55.9%**
oracle (VM5+VM20)	26.26	26.35	-	-

Overall, Err2Unk performs better than Err2Lat on LS100-VM20 and LG4-LG19 but not on VM5-VM20 where the error detector achieves a lower F1 score for the <error> class, as mentioned in Section 4.2. Analysis of the error detection trade-off should give finer control on these proposed methods.

Table 4: *WER (%) achieved in the matched-domain human-machine dialog setup (LG4-LG19). (Best result, and ones statistically equivalent to it at p=0.05, are highlighted in bold.)*

Type of Supervision	Dev WER	Test WER	Test RWI	Test WRR
seed (LG4)	37.90	37.10	-	-
Semi-sup best path	34.29	33.04	10.9%	61.8%
Semi-sup LF-MMI	33.86	32.67	11.9%	67.4%
Semi-sup Err2Unk	**32.71**	**31.25**	**15.8%**	**89.1%**
Semi-sup Err2Lat	**32.90**	31.65	14.7%	83.0%
oracle (LG4+LG19)	32.09	30.53	-	-

5. Conclusion

Existing methods for semi-supervised training of acoustic model using LF-MMI exhibit a different behavior in small-data and/or domain-mismatched scenarios. We proposed two new methods which use an error detector to control the supervision provided for learning from unlabeled speech data. An evaluation on three different limited data setups, emulating domain-mismatched and real application scenarios, confirms that error detection driven supervision performs better than classical best-path or lattice-based based semi-supervised LF-MMI training. WRRs of 28 to 89% are reported. Our future work will explore finer error control and methods to bias the lattice supervision.

6. Acknowledgments

This work was supported by the European Union's Horizon 2020 Research and Innovation Program under Grant Agreement No. 825081 COMPRISE (https://www.compriseh2020.eu/). Experiments were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000.fr).

7. References

[1] J. Y. Kim, C. Liu, R. A. Calvo, K. McCabe, S. C. R. Taylor, B. W. Schuller, and K. Wu, "A comparison of online automatic speech recognition systems and the nonverbal responses to unintelligible speech," *arXiv preprint arXiv:1904.12403*, 2019.

[2] B. Rizk, "Evaluation of state of art open-source ASR engines with local inferencing," Bachelor's Thesis, Institute of Information Systems, Hof University, 2019.

[3] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *23rd International Conference on Machine Learning (ICML)*, 2006, p. 369–376.

[4] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for ASR based on lattice-free MMI," in *Interspeech*, 2016, pp. 2751–2755.

[5] H. Hadian, H. Sameti, D. Povey, and S. Khudanpur, "End-to-end speech recognition using lattice-free MMI," in *Interspeech*, 2018, pp. 12–16.

[6] V. Manohar, H. Hadian, D. Povey, and S. Khudanpur, "Semi-supervised training of acoustic models using lattice-free MMI," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4844–4848.

[7] P. Ghahremani, V. Manohar, H. Hadian, D. Povey, and S. Khudanpur, "Investigation of transfer learning for ASR using LF-MMI trained neural networks," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2017, pp. 279–286.

[8] T. Lo and B. Chen, "Semi-supervised training of acoustic models leveraging knowledge transferred from out-of-domain data," in *2019 APSIPA Annual Summit and Conference*, 2019, pp. 1400–1404.

[9] A. Carmantini, P. Bell, and S. Renals, "Untranscribed web audio for low resource speech recognition," in *Interspeech*, 2019, pp. 226–230.

[10] J. Fainberg, O. Klejch, S. Renals, and P. Bell, "Lattice-based lightly-supervised acoustic model training," in *Interspeech*, 2019, pp. 1596–1600.

[11] S. Tong, A. Vyas, P. N. Garner, and H. Bourlard, "Unbiased semi-supervised LF-MMI training using dropout," in *Interspeech*, 2019, pp. 1576–1580.

[12] A. Vyas, P. Dighe, S. Tong, and H. Bourlard, "Analyzing uncertainties in speech recognition using dropout," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6730–6734.

[13] Y. Tam, Y. Lei, J. Zheng, and W. Wang, "ASR error detection using recurrent neural network language model and complementary ASR," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 2312–2316.

[14] R. Errattahi, A. E. Hannani, F. Z. Salmam, and H. Ouahmane, "Incorporating label dependency for ASR error detection via RNN," *Procedia Computer Science*, vol. 148, pp. 266 – 272, 2019.

[15] K. Lybarger, M. Ostendorf, and M. Yetisgen, "Automatically detecting likely edits in clinical notes created using automatic speech recognition," in *AMIA Annual Symposium*, 2017.

[16] S. Thomas, M. L. Seltzer, K. Church, and H. Hermansky, "Deep neural network features and semi-supervised training for low resource speech recognition," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 6704–6708.

[17] K. Veselý, L. Burget, and J. Černocký, "Semi-supervised DNN training with word selection for ASR," in *Interspeech*, 2017, pp. 3687–3691.

[18] F. Grezl and M. Karafiat, "Semi-supervised bootstrapping approach for neural network feature extractor training," in *2013 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2013, pp. 470–475.

[19] P. Zhang, Y. Liu, and T. Hain, "Semi-supervised DNN training in meeting recognition," in *2014 IEEE Spoken Language Technology Workshop (SLT 2014)*, 2014, pp. 141–146.

[20] H. Xu, D. Povey, L. Mangu, and J. Zhu, "Minimum Bayes risk decoding and system combination based on a recursion for edit distance," *Computer Speech and Language*, vol. 25, no. 4, pp. 802–828, 2011.

[21] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[22] Mozilla common voice. [Online]. Available: https://voice.mozilla.org/

[23] S. Burger, K. Weilhammer, F. Schiel, and H. G. Tillmann, "Verbmobil data collection and annotation," in *Verbmobil: Foundations of Speech-to-Speech Translation*, W. Wahlster, Ed., 2000, pp. 537–549.

[24] G. Parent and M. Eskenazi, "Toward better crowdsourced transcription: Transcription of a year of the Let's Go bus information system data," in *2010 IEEE Spoken Language Technology Workshop (SLT)*, 2010, pp. 312–317.

[25] DailRC. The integral LET'S GO! dataset. Last accessed April 1, 2020. [Online]. Available: https://dialrc.github.io/LetsGoDataset/

[26] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2013, pp. 55–59.

[27] Kaldi-help: train_more2 in chain / nnet3 scenario. Last accessed April 1, 2020. [Online]. Available: https://groups.google.com/d/msg/kaldi-help/K6fXrt0vMtM/zXvI9ytyAAAJ