BEING GREEDY DOES NOT HURT: SAMPLING STRATEGIES FOR END-TO-END SPEECH RECOGNITION

Jahn Heymann[*], *Egor Lakomkin*[*], *Leif Rädel*[*]

Amazon.com
{jahheyma,egorlako,lraedel}@amazon.com

ABSTRACT

Maximum Likelihood Estimation (MLE) is currently the most common approach to train large scale speech recognition systems. While it has significant practical advantages, MLE exhibits several drawbacks known in literature: training and inference conditions are mismatched and a proxy objective is optimized instead of word error rate. Recently, the Optimal Completion Distillation (OCD) training method was proposed which attempts to address some of those issues. In this paper, we analyze if the method is competitive over a strong MLE baseline and investigate its scalability towards large speech data beyond read speech, which to our knowledge is the first attempt known in literature. In addition, we propose and analyze several sampling strategies trading off exploration and exploitation of unseen prefixes and their effect on ASR accuracy. We conduct several experiments on both public LibriSpeech data and in-house large scale far-field data and compare models trained with MLE and OCD. Our proposed greedy sampling with soft targets approach proves most effective and yields a 9% rel. word error rate improvement over the i.i.d sampling. Finally, we note that OCD method improves over the MLE without label smoothing by 12%, and underperform by 6% once label smoothing is introduced to MLE.

***Index Terms*—** Speech recognition, non-maximum likelihood training, training criteria

1. INTRODUCTION

Currently, MLE is the predominant approach to train end-to-end Automatic Speech Recognition (ASR) models. While this is a foundation of modern architectures including Connectionist Temporal Classification (CTC) [2], Listen-Attend-Spell (LAS) [3], and Recurrent Neural Network Transducer (RNN-T) [4], it has certain limitations. Likelihood is only a proxy metric towards Word Error Rate (WER) and does not capture improvements in Spoken Language Understanding (SLU) metrics like entity recognition. Additionally there is a dependency on teacher-forcing where the model is exposed only to ground-truth prefixes during training. This can negatively impact accuracy [5] and result in over-confident models [6].

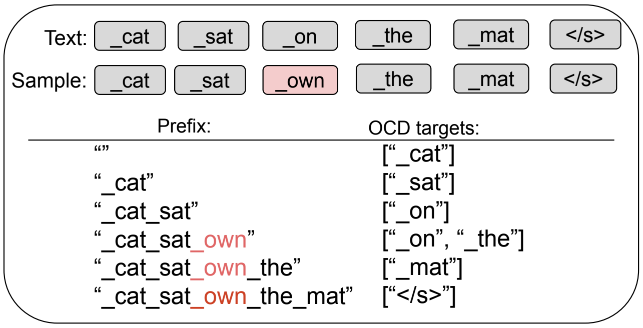

Fig. 1. An example of the way OCD [1] method works on imperfect prefixes. For every prefix OCD evaluates every word-piece as a candidate extensions and assigns score to it representing the edit distance between a hypothesis (composed from the prefix, candidate word-piece and optimal extension) and a ground truth.

Several approaches were proposed to address the issues related to maximum likelihood training for end-to-end ASR. These include non-maximum likelihood training criterion like Minimum Word Error Rate Training (MWER) [7, 8], MLE-guided parameter search [9], promising prefix boosting [10], and policy gradient [11, 12]. Another direction of research tackles the teacher forcing dependency like scheduled sampling [13], learning to search during training [14, 15, 16, 17]. Despite promising results, these approaches have certain limitations and require the MLE training criteria for pre-training or as an auxiliary loss, or incur additional complexity and require significantly more compute than the MLE method. Additionally, most make crude approximations such as using n-best lists for MWER computation and provide only sequence-level feedback making it difficult to emphasize contributions of individual words in the training criteria. MWER further requires pre-training, cannot be trained from scratch and needs to trade-off model recency and training speed in case of caching n-best decoding results.

In this paper we investigate the recently proposed OCD [1] method. The most notable advantages of OCD are that it has

[*] Author list is in alphabetical order.

978-1-6654-0540-9/22/$31.00 ©2022 IEEE

the same computational complexity as MLE, does not use teacher forcing, and attempts to directly optimize word-piece edit distance, which is closer to WER than likelihood. The original paper reported results on LibriSpeech and WSJ datasets and it has been an open question if the method offers similar gains on large-scale datasets.

We fill this gap and conduct several experiments evaluating OCD method and comparing it with a strong MLE baseline across two datasets: Librispeech and far-field in-house large-scale data. Our initial results showed a performance gap between our OCD re-implementation and MLE as opposed to significant improvements reported in the original paper. Our MLE baseline has a WER of 8.18% which is significantly better than the baseline reported in the original paper of 15.4%. Additionally, the OCD code was not released publicly. To address this gap we propose several strategies for obtaining samples from the model and conclude that the greedy sampling yields significantly better performance than the i.i.d method used in the original paper.

2. OPTIMAL COMPLETION DISTILLATION

OCD [1] is an edit distance based training algorithm for sequence-to-sequence models. In contrast to MLE, OCD does not require teacher forcing for training. Instead it is directly trained on samples drawn from the ASR model exposing the model to its own mistakes and learning to correct them.

The objective is to learn a conditional autoregressive model $p_\theta(\mathbf{y}^*|\mathbf{x})$ of the target sequence $\mathbf{y}^*$ given input audio $\mathbf{x}$ from a dataset $\mathcal{D} = \{(\mathbf{x}, \mathbf{y}^*)\}$. For a single sample OCD aims to minimize the per step (indexed by t) KL divergence between the model output distribution and an optimal (soft) policy π^*:

$$\mathcal{J}_{\text{OCD}}(\theta) = \mathbb{E}_{\widetilde{\mathbf{y}}\sim p_\theta(\cdot|\mathbf{x})} \sum_{t=1}^{|\widetilde{\mathbf{y}}|} \mathrm{KL}(\pi^*(\cdot|\widetilde{\mathbf{y}}_{<t})||p_{\theta,t}(\cdot|\widetilde{\mathbf{y}}_{<t}, \mathbf{x})).$$

The optimal policy for each possible extension y_t (e.g. a word-piece) is obtained from Q-values with temperature parameter τ

$$\pi^*(y_t|\mathbf{y}_{<t}) = \frac{\exp(Q^*(\mathbf{y}_{<t}, y_t)/\tau)}{\sum_{y'} \exp(Q^*(\mathbf{y}_{<t}, y')/\tau)}. \qquad (1)$$

The Q-values themselves are computed as

$$Q^*(\mathbf{y}_{<t}, y_k) = \begin{cases} -m_t & \text{if } k \in \mathcal{O} \\ -m_t - 1 & \text{otherwise} \end{cases}$$

where m_t is the minimal obtainable edit distance given the prefix $\mathbf{y}_{<t}$ and $\mathcal{O}$ is the set of tokens that allows to obtain this edit distance when extended to the prefix. For further details on an efficient implementation we refer to [1].

2.1. Hard and soft targets

A simple technique to improve overconfidence for models trained with MLE is Label Smoothing (LS) [18]. In *uniform* LS instead of using 1-hot targets the probability for the correct label is set to $1 - \beta$ and the remaining β probability is distributed uniformly across all other labels. By default we use $\beta = 0.1$ for *LS* in the *smoothing policy*.

For OCD a comparable approach to LS is to use $\tau \neq 0.0$ in Eq. 1. That means that instead of hard targets for the optimal extensions we distribute part of the probability to other labels resulting in *soft* targets. Note that τ and β do not have the same effect because of the softmax when computing the optimal policy and because for OCD there can be multiple optimal completions.

3. SAMPLING STRATEGIES

In [1] a full sequence $\widetilde{\mathbf{y}} \sim p_\theta(\cdot|\mathbf{x})$ is drawn *i.i.d.* from the model and then used to calculate the OCD objective. However, here we also explore alternative approaches to sampling a sequence from the model.

We evaluated the following *sampling strategies*. Firstly, *greedy* sampling from the model by taking the most likely output token at every step (e.g. *argmax*). The advantage of sampling greedily is that at the beginning of training the model converges quickly as it does not explore as much as when sampling from the entire distributions.

As a modification of sampling *i.i.d.* we introduce the temperature T of the softmax output during sampling. Temperature values $T \leq 1.0$ sharpen the probability distribution and can be seen as an intermediate step between *greedy* and i.i.d. sampling.

Further, by decoding the input audio using beam search with the current model one can obtain the n-best hypotheses. The n-best hypotheses can then be used for calculating the OCD loss by selecting a single sequence by either taking the best hypothesis or sampled based on their probability. Alternatively, we propose to use the entire n-best hypotheses and calculating the OCD loss given input audio using n samples drawn from the model. We denote n-best sampling with N and the type of sampling by *1-best*, *log-prob* or *all* when reporting our results, respectively.

4. EXPERIMENTAL RESULTS

Model Architecture: All our models are trained using 64 log-mel filterbanks, computed over 25ms window and 10ms stride. For our large-scale experiments the model consists of six-layer bi-directional LSTM [19] with 1,024 units encoder and a two-layer uni-directional LSTM decoder with a two head content-based attention mechanism [20]. Overall the model has 119 million trainable parameters. The model has been trained with Adam optimizer [21]. We use a SentencePiece [22] unigram word-piece model with 2,500 tokens. For

Table 1. Results with MLE- and OCD-trained LAS models on LibriSpeech *test-clean* and *test-other* splits. We highlight our best MLE and OCD models.

Loss	smoothing policy	sampling strategy	WER [%] test clean	 test other
MLE [23]	*LS*	-	*3.2*	*8.0*
MLE [1]	-	-	*8.4*	*15.4*
MLE	-	-	3.8	10.0
MLE	LS	-	**3.16**	**8.18**
OCD [1]	*hard*	*i.i.d*	*6.4*	*13.3*
OCD	hard	i.i.d	3.67	9.85
OCD	soft	i.i.d	3.71	9.71
OCD	soft	greedy	**3.23**	**8.80**
OCD	soft	i.i.d (T=0.1)	3.44	9.61
OCD	soft	N (1-best)	3.61	8.93
OCD	soft	N (all)	3.66	9.30
OCD	-	N (all)	3.67	9.22
OCD	soft	N (log-prob)	3.36	9.00

our LibriSpeech experiments, we use the same model architecture but only single-headed attention. SpecAugment data augmentation method with LD policy [23] was used throughout model training. The model was trained under the weight noise [24], which was added to all encoder trainable parameters by sampling noise from a normal distribution with a standard deviation of 0.075 starting from 15k training steps. For the MLE model we use uniform label smoothing [18] which distributes a probability mass of 0.1 to non-ground truth tokens. In addition, we add a weight decay factor of $1e^{-6}$ to the Adam optimizer. We use a SentencePiece [22] unigram word-piece model with 4,000 tokens. Overall the model has 96 million trainable parameters.

Datasets: We compare MLE and OCD-trained models on two datasets: an in-house collection of de-identified real recordings of natural human interactions with voice-controlled far-field devices and the publicly available LibriSpeech [25] corpus, where we used full 960 hours training set. For the far-field recognition task, we trained the models with approximately 60k hours of transcribed audio data. Models are evaluated on two different test sets: GENERAL - a general test set following the same distribution as the training data ($\sim$17k utterances), and RARE - a sample of utterances containing rare words ($\sim$50k utterances). LibriSpeech models are evaluated on two test sets: test-clean and test-other with the best checkpoint identified on the dev-other test set. We tuned several decoding hyperparameters before evaluating on the test data: temperature, length normalization, coverage penalty, and beam size on the dev-other set and far-field dev sets for LibriSpeech and far-field experiments respectively.

Table 2. Large-scale far field results comparing MLE and OCD variants. Relative WER reduction w.r.t the MLE model on the GENERAL and RARE test sets are presented. We set our baseline to 0.0 as a reference. Positive values indicate the improvement over the MLE model.

Loss	smoothing policy	sampling strategy	GENERAL	RARE
MLE	LS	-		
OCD	soft	i.i.d		
OCD	soft	greedy		
OCD	soft	N (all)	-1.4%	-5.6%

Results: We present our LibriSpeech results in Table 1. The OCD model with i.i.d sampling underperform the greedy OCD with 9.7% WER and 8.8% on *test-other* using i.i.d and greedy sampling respectively. Both show better performance than our MLE baseline without label smoothing (10.0% WER on *test-other*) and underperform once we add label smoothing to MLE (8.18% WER on *test-other* and matching the reference performance reported in [23]). Furthermore using soft targets for constructing OCD policy is slightly better than using hard targets. This suggests that even when using a greedy policy which is better than the i.i.d policy used in the original OCD paper we could not improve over the MLE baseline with label smoothing. Furthermore, sampling from the n-best show a consistent improvement over the i.i.d sampling with using the top-1 hypothesis in a beam resulting in 8.93% WER, and using all hypotheses from n-best with and without smoothing results in 9.3% and 9.2% WER. Sampling from the model output distribution over n-best in 9.0% WER on *test-other* respectively. Overall these experiments suggest that exploitation during the sampling process results in better performance with the best models being trained either with greedy sampling or taking one-best from the beam.

Furthermore we present our results on a large scale far-field speech dataset in Table 2. We observe a similar trend as in our LibriSpeech experiments: greedy sampling yields the best results compared to i.i.d sampling and beam sampling. On *GENERAL* test set the OCD greedy model shows an improvement over the MLE baseline. However on the *RARE* test set all models trained with OCD training criteria are behind our MLE baseline. We can correlate these observations with the Librispeech results where the OCD model with greedy sampling had similar performance on *test-clean* as MLE with label smoothing but was lagging behind on *test-other*.

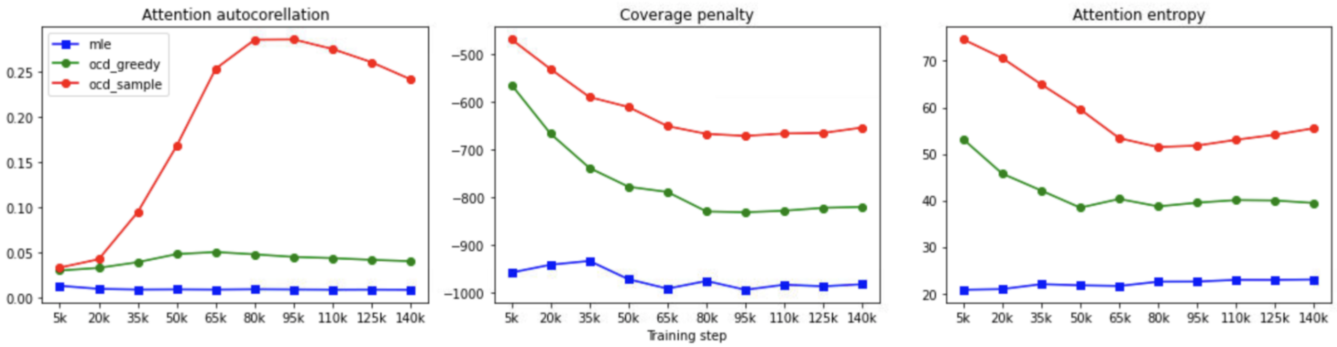

Fig. 2. Attention autocorrelation, coverage penalty, and attention entropy metrics captured at the different training stages for MLE, greedy OCD, and i.i.d OCD-trained models.

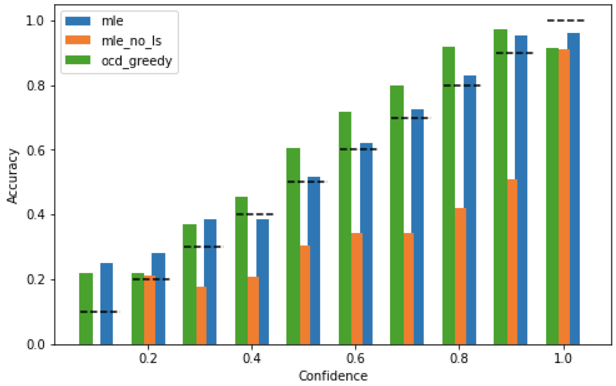

Fig. 3. Calibration plot with model accuracy per confidence bin of the MLE (blue), MLE without Label Smoothing (orange), and greedy OCD (green) models.

5. ANALYSIS

Decoding hyperparameters sensitivity. Firstly we note the difference in the preference of decoding hyperparameters by MLE and OCD models after tuning. We observed that MLE model attains the best performance with decoding temperature values of 1.3 and with a length penalty of 1.0 after tuning on the dev set. OCD did not benefit from any decoding parameters tuning and default settings (temperature 1.0 and no length penalty) were the best. We evaluated the performance of the MLE model with default decoding hyperparameters and the gap between the OCD and MLE is reduced from 6.8% relative to 4.6% relative. This insensitivity of OCD trained model to decoder parameters tuning can suggest that different decoder hyperparameters may be needed to be considered and tuned for the models trained with OCD. Further we looked at the oracle WER and Character Error Rate (CER) metrics and attempt to verify a hypothesis that a OCD model does more meaningful mistakes as it is always trained via sampling and optimizing word-piece edit-distance distance. However, we could not find any evidence of that as CER, Oracle CER, and Oracle WER metrics are better for the MLE model.

Attention metrics. To get a better understanding of the nature of nearly duplicate words appearing in hypotheses in the n-best we take a closer look at the attention behaviour. We computed several attention-based metrics on the *dev-other* utterances: 1) encoder states coverage penalty [26], attention entropy per utterance, and 3) dot product of attention vectors consecutive in time. We present the dynamics of those metrics throughout model training in Figure 2. We note a significant difference between the MLE and OCD models, where attention for OCD model with i.i.d sampling is on average significantly less peaked (as attention entropy and coverage terms are both relatively higher) and more autocorrelated compared to MLE. We hypothesise that training with OCD criteria is a significantly harder task in the case when multiple optimal completions are available

Calibration metrics. Further we compare how MLE and OCD models are calibrated by plotting accuracy of the model predictions for every confidence bin (see Fig. 3) and computing Expected Calibration Error (ECE) metric. We note that MLE without label smoothing (ECE of 11.4%) tend to be overconfident and has the worst calibration compared to OCD (ECE of 8.4%) and MLE with label smoothing (ECE of 4.3%). Overall the OCD model tends to be under-confident across almost all confidence bin, which could be a side effect of having multiple optimal targets once the model is making a mistake during sampling.

6. CONCLUSIONS

In this paper we presented our experiments on the non-maximum likelihood training for LAS architecture. To our knowledge, this is the first attempt to reproduce the original OCD paper results and verify if the same reported improvements hold when using a stronger MLE baseline. Furthermore, we presented experimental results on a larger scale dataset beyond read speech. Our conclusions suggest that *greedy* sampling with soft targets method performs significantly better than the *i.i.d* sampling with hard targets. OCD shows significantly better convergence and improves the WER over MLE without label smoothing and at the same time lags behind the MLE model with label smoothing. Our conclusion is that OCD has the potential to replace the MLE method eventually. We leave for future work extending OCD to support metrics beyond word-piece edit distance and designing a better trade-off between exploration and exploitation of model prefixes improving further sampling policy.

7. REFERENCES

[1] Sara Sabour, William Chan, and Mohammad Norouzi, "Optimal completion distillation for sequence learning," *CoRR*, vol. abs/1810.01398, 2018.

[2] A. Graves, Santiago Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," *Proceedings of the 23rd international conference on Machine learning*, 2006.

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

[4] A. Graves, "Sequence transduction with recurrent neural networks," *ArXiv*, vol. abs/1211.3711, 2012.

[5] Chung-Cheng Chiu, Tara N. Sainath, Yonghui Wu, Rohit Prabhavalkar, et al., "State-of-the-art speech recognition with sequence-to-sequence models," *CoRR*, vol. abs/1712.01769, 2017.

[6] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger, "On calibration of modern neural networks," *ArXiv*, vol. abs/1706.04599, 2017.

[7] Rohit Prabhavalkar, Tara N. Sainath, Yonghui Wu, Patrick Nguyen, et al., "Minimum word error rate training for attention-based sequence-to-sequence models," *CoRR*, vol. abs/1712.01818, 2017.

[8] Matt Shannon, "Optimizing expected word error rate via sampling for speech recognition," *CoRR*, vol. abs/1706.02776, 2017.

[9] Sean Welleck and Kyunghyun Cho, "Mle-guided parameter search for task loss minimization in neural sequence modeling," *CoRR*, vol. abs/2006.03158, 2020.

[10] Murali Karthick Baskar, Lukáš Burget, Shinji Watanabe, Martin Karafiát, et al., "Promising accurate prefix boosting for sequence-to-sequence asr," 2018.

[11] Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura, "Sequence-to-sequence ASR optimization via reinforcement learning," *CoRR*, vol. abs/1710.10774, 2017.

[12] Yingbo Zhou, Caiming Xiong, and Richard Socher, "Improving end-to-end speech recognition with policy learning," *CoRR*, vol. abs/1712.07101, 2017.

[13] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," *CoRR*, vol. abs/1506.03099, 2015.

[14] R. Leblond, Jean-Baptiste Alayrac, Anton Osokin, and Simon Lacoste-Julien, "SEARNN: training rnns with global-local losses," *CoRR*, vol. abs/1706.04499, 2017.

[15] Kai-Wei Chang, Akshay Krishnamurthy, Alekh Agarwal, Hal Daumé III, et al., "Learning to search better than your teacher," *CoRR*, vol. abs/1502.02206, 2015.

[16] Ronan Collobert, Awni Hannun, and Gabriel Synnaeve, "A fully differentiable beam search decoder," *CoRR*, vol. abs/1902.06022, 2019.

[17] Sam Wiseman and Alexander M. Rush, "Sequence-to-sequence learning as beam-search optimization," *CoRR*, vol. abs/1606.02960, 2016.

[18] J. Chorowski and Navdeep Jaitly, "Towards better decoding and language model integration in sequence to sequence models," in *INTERSPEECH*, 2017.

[19] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[20] J. Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, et al., "Attention-based models for speech recognition," in *NIPS*, 2015.

[21] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2015.

[22] Taku Kudo and John Richardson, "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," 2018.

[23] Daniel S. Park, William Chan, Y. Zhang, C. Chiu, et al., "Specaugment: A simple data augmentation method for automatic speech recognition," in *INTERSPEECH*, 2019.

[24] A. Graves, "Practical variational inference for neural networks," in *NIPS*, 2011.

[25] Vassil Panayotov, Guoguo Chen, Daniel Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210, 2015.

[26] Yonghui Wu, M. Schuster, Z. Chen, Quoc V. Le, Mohammad Norouzi, et al., "Google's neural machine translation system: Bridging the gap between human and machine translation," *ArXiv*, vol. abs/1609.08144, 2016.