



Dealing with Unknowns in Continual Learning for End-to-end Automatic Speech Recognition

Martin Sustek^{1,2,*}, Samik Sadhu^{2,*}, Hynek Hermansky^{1,2,3}

¹Brno University of Technology, Czechia

²Center for Language and Speech Processing, Johns Hopkins University, USA

³Human Language Technology Center of Excellence, Johns Hopkins University, USA

isustek@fit.vut.cz, samiksadhu@jhu.edu, hynek@jhu.edu

Abstract

Learning continually from data is a task executed effortlessly by humans but remains to be of significant challenge for machines. Moreover, when encountering unknown test scenarios machines fail to generalize. We propose a mathematically motivated dynamically expanding end-to-end model of independent sequence-to-sequence components trained on different data sets that avoid catastrophically forgetting knowledge acquired from previously seen data while seamlessly integrating knowledge from new data. During inference, the likelihoods of the unknown test scenario are computed using internal model activation distributions. The inference made by each independent component is weighted by the normalized likelihood values to obtain the final decision.

Index Terms: speech recognition, continual learning, multi-stream speech recognition

1. Introduction

Automatic Speech Recognition (ASR) systems work best when they are used on speech with the same characteristics as that of its own training data. Due to this lack of generalization, individualized ASRs are trained on different environmental conditions (eg. clean-read speech, reverberated speech, noisy speech from particular environments) each supplied with its own training data and matched test data for evaluation. Aggregating all available data from all tasks and training one multi-condition ASR model has two fundamental problems

- All old data that has been used to train the current multi-condition ASR needs to be preserved to train another multi-condition ASR in the event of a new future data set
- Multi-condition training can negatively affect ASR performance if data from widely different data domains are aggregated together, a case in point being multi-lingual ASR [1]. Even with English speech data, multi-condition training with clean-read as well as noisy speech impacts the performance on clean speech [2, 3, 4].

Motivated by these drawbacks, we move to a Continual Learning paradigm to build a general-purpose ASR that can operate under an unknown test condition that might or might not be included under the training conditions without the need to aggregate all past training data or carry out multi-condition training.

*Equal contribution

2. Continual Learning in ASR

Continual Learning (also called Lifelong Learning) has been of significant interest [5, 6, 7], with the ultimate goal of learning from a sequence of data potentially coming from different *domains* or *tasks* without suffering a catastrophic reduction in performance on old domains.

When dealing with different speech environmental conditions for ASR, in recent literature, mainly two types of Continual Learning techniques stand out - a) Memory Replay based methods that keep a portion of the old data to relearn past knowledge episodically [8, 9] b) Expanding a model dynamically to preserve knowledge about old domains in frozen network parameters [10, 6]. Our approach is an extension of category (b) that can even be easily used to address a scenario in which data is collected simultaneously at multiple locations or “nodes” coming from either the same or different environment with the constraint that data cannot be manipulated outside of its source location due to data privacy, a typical condition for federated learning.

Our proposition is to gradually add tuples of models trained on data gathered in each node independently of other nodes. Each tuple consists of an ASR and a generative model (in our case a Variational Auto-encoder or VAE) used to provide a performance measure, somewhat similar to [11]. During inference, the decisions from each independently trained model are combined using their performance measures.

3. Our Approach

Our approach is to use a mathematically-motivated combination of models. For the sake of clarity, let’s assume we want to compute posterior distribution over characters Y for only one input feature frame X of the speech signal using multiple ASR models M^1 . We can write the joint distribution of X and Y as

$$p(X, Y) = \sum_{i=1}^{|M|} p(X, M_i) p(Y|X, M_i) \quad (1)$$

from which follows

$$p(Y|X) = \frac{\sum_{i=1}^{|M|} p(X, M_i) p(Y|X, M_i)}{p(X)} \quad (2)$$

This conditional distribution can be interpreted as

$$p(Y|X) = \sum_{i=1}^{|M|} w_i p(Y|X, M_i) \quad (3)$$

¹We use notation $p(X|M_i)$ to simplify $p(X|M = i)$ and $|M|$ to denote the the number of ASR models

with

$$w_i = p(M_i|X) = \frac{p(X, M_i)}{\sum_{j=1}^{|M|} p(X, M_j)}, \quad (4)$$

where w_i is the weight used to combine posterior distribution $p(Y|X, M_i)$ from the M_i^{th} ASR model. In this formulation, prior distribution $p(M)$ corresponds to the probability of picking the particular model before observing X . It can reflect the amount of data being used in each model, the performance of each model on its own train/dev dataset or it can simply be set to a uninformative uniform distribution as in the case of our experiments.

3.1. Combination Strategy Based on Input Feature Distribution

Similar to our earlier work [11], we assumed uniform prior $p(M)$ in Equation 4 and applied Bayes rule to get $w_i = \frac{p(X|M_i)}{\sum_{j=1}^{|M|} p(X|M_j)}$. We model $\log p(X|M_i)$ with the evidence lower bound (ELBO) from a Variational Autoencoder [12] (VAE) trained on the same input data that were used to train the corresponding ASR. In order to evaluate w_i , we apply a softmax function to a vector - of which the i^{th} element is $\log(p(X|M_i))$, the *score* for the i^{th} ASR from the i^{th} VAE². Note that the scores (and therefore combination weights) are based on statistical proximity of an unknown feature vector X to different input data distributions and is completely independent of the ASR model itself.

3.2. Combination Strategy Based on ASR Internal Activation Distribution

Considering entirely ASR independent scores $\log(p(X|M_i))$ (under the assumption of a flat prior) is problematic especially because it has no direct correlation to how well the classifier behaves on unknown test conditions. Logically, the *confidence* of the ASR model has to feature as a part of the score and consequently in the combination strategy.

We use joint CTC/attention end-to-end ASRs that have a core encoder-decoder [13] architecture with a CTC module added onto the encoded features. The likelihood of an encoded vector under test conditions w.r.t its distribution under training conditions is a measure of how close the model is to the familiar operational territory. Therefore an alternative and more reasonable way of estimating the weights w_i would be to score the i^{th} ASR with $\log(p(H_i|M_i))$ with $H_i = ASR_{ENC_i}(X)$, $ASR_{ENC_i}(\cdot)$ being the encoder of the i^{th} end-to-end ASR. As in our old proposition, we model $\log(p(H_i|M_i))$ with the ELBO from VAE models trained *after* the i^{th} ASR has already been trained. To distinguish between these two approaches, we will refer to the original combination of weights as w_i^{INP} and the proposed weights as w_i^{ASR} .

To compare the two strategies in simple terms, our old strategy assigns scores to different ASR models based on the similarity of the unknown test data to the training data of the ASR models. In contrast, the proposed strategy defines a confidence score by measuring how closely the encoded representations under test conditions match to those during training for each ASR.

3.3. Dealing With A Sequence of Frames

Previously, we described the process of obtaining posterior distribution $p(Y|X)$ using multiple models M for one input frame

²The same can be done even when we don't assume flat prior $p(M)$, in this case the input to softmax is $\log(p(X|M)) + \log(p(M))$

X , which however needs to be extended to a sequence of frames for end-to-end ASR models. Using different weights w_i at different time steps is not straightforward³. However, since at every time step, an attention-based decoder can use the whole input sequence for predicting the next character in the sequence, it provides justification to simply use the same weights w_i for all time steps.

We chose two different approaches for combing per frame scores from VAE into time-independent weights for our two combination strategies as shown in Equation 5 and 6 respectively.

$$w_i^{INP} = \text{softmax}\left(\frac{1}{T} \sum_{t=1}^T \log(p(X_t|M))\right)_i \quad (5)$$

$$w_i^{ASR} = \frac{1}{T} \sum_{t=1}^T \text{softmax}(\log(p(H_t|M)))_i \quad (6)$$

For VAEs modelling the distribution of input variable X , we compute the average log-likelihoods across time before computing the softmax to obtain w_i^{INP} . However, the encoded representations H have different characteristics compared to X . In our experiments, we observed that ASR encoders use very similar representations for all silence frames which leads to the VAE assigning very high likelihoods to silences. Besides, outlier frames were observed to occur more often for VAE modelling of H which easily biases the average log-likelihood across time. Therefore for w_i^{ASR} , we compute the average of the per-frame weight across time.

Given a speech utterance X_1, X_2, \dots, X_T from an unknown test condition, we forward pass it through each individual ASR encoders $ASR_{ENC_i}(\cdot)$ to obtain the encoded representations. Thereafter, either Equation 5 or 6 is used (depending on the combination strategy) to compute the combination weights. We follow the autoregressive decoding pipeline in ESPnet modified for our setup:

1. CTC being non-autoregressive, we first combine the CTC posterior from all ASRs using Equation 3.
2. The attention-based part of each ASR decoder is used to generate the posterior distribution over the characters at the t^{th} time step using input X_1, X_2, \dots, X_T and already predicted sequence $Y_{1:t-1}$. Distributions from all ASR decoders are weighted and summed according to Equation 3 to hypothesize the next character Y_t .
3. We re-weight the probability of hypotheses $Y_{1:t-1}$ using CTC decoder prefix scores and language model scores and keep only n -best hypothesis, n being the beam size.
4. Steps 2 and 3 are repeated until the whole output sequence is generated.

In our experiment, we used a single language model, but the same approach could be extended to combine different language models as well.

4. Experimental Details

4.1. Data sets

For ASR training, we use a clean-read speech from Wall Street Journal (WSJ) and artificially reverberated speech from RE-VERB [14] data sets, each split into two parts with a unique

³We also performed experiments while using only CTC-based decoding. The performance for shared weight and using the different weights for each time step in this model was similar.

set of speakers. The main motivation behind this choice is to allow us to simulate conditions where training data can come both from the same and different environments.

Our in-domain test sets come from WSJ and REVERB which have clean-read and *real* reverberated single-channel speech test sets respectively. Apart from that, we included two additional test sets from conditions unseen during training, namely Aurora-4 [15] and CHiME-4 [16] test sets.

4.2. ASR and VAE Models

We use end-to-end ASR models trained with ESPnet [17] ASR toolkit with FDLP-spectrogram input features [18]. In particular, we use a hybrid CTC/attention-based encoder-decoder architecture with 0.3 CTC weight⁴. In addition, we use a RNN based language model and we execute decoding with a beam size of 10.

Our VAE encoder and decoder are 3-layer LSTMs and the encoder, as well as decoder output, are assumed to be Laplace distributed. Even though our encoder and decoder consist of LSTM layers, we do not model the likelihood of sequences and train the VAE to maximize per-frame likelihoods⁵.

5. Results

We report a single ASR word error rate (WER) on all used test data sets in Table 1. The difference between the performance of ASR models trained on the first and the second part of REVERB dataset (referenced as R1 and R2) is small, but R2 usually performs slightly better. On the other hand, the performance of models W1 and W2 that were trained on the first and the second part of WSJ datasets differ more. W1 and W2 have almost the same performance on WSJ test set, but on all other datasets, W1 performs better than W2. These differences might be caused by the fact that both data sets were divided based on mutually exclusive speaker sets. R1 and R2 are performing worse than W1 and W2 on all but their own test sets.

Table 1: WER of single ASR models are shown [%]. **W1**, **W2**, **R1** and **R2** represent ASRs models on 2 splits of the WSJ and REVERB data set respectively. The columns represent different test sets.

	WSJ	REVERB	AURORA-4	CHiME-4
W1	6.3	69.2	21.9	39.8
W2	6.2	77.7	26.1	49.9
R1	41.1	43.2	52.4	71.0
R2	38.5	41.6	51.0	71.0

Next, we evaluate the performance of various collections of ASR models using our proposed combination strategies. Apart from the results of combination weights defined in Equation 5 (VAE INP) and Equation 6 (VAE ASR), we also report the performance of using the same weight $\frac{1}{|M|}$ for all ASR models (SAME WEIGHT) as well as the performance of the best single model (BEST SINGLE) among a particular set of ASRs (assuming there is an oracle system that could do the selection).

⁴Further details of the acoustic model can be found in https://github.com/espnet/espnet/blob/master/egs/wsj/asr1/conf/tuning/train_pytorch_transformer.yaml.

⁵Our code can be found in <https://github.com/sadhusamik/multistream>

We aim for our setup to satisfy the following conditions during evaluation:

1. When combining two ASRs that were trained on data from a similar domain, the performance of the VAE-based combination should approximate the performance of the SAME WEIGHT approach.
2. When combining two ASRs that were trained on data from different domains, the performance of our system should be close to picking the best possible single model.
3. Under the Continual Learning condition, using multiple classifiers should not degrade the performance to below the BEST SINGLE performance. In addition, adding a new ASR to a set of ASRs should not drastically reduce performance.

Condition 1 is tested by evaluating our combination strategies on collections of two models trained on data from the same domain (W1+W2 and R1+R2). Condition 2 can be tested using any combinations of WSJ and REVERB ASRs while performance evaluations for 2 (W1+R1), 3 (W1+W2+R1), and all 4 models allow us to check for any contradictions to condition 3.

5.1. Decoding on in-domain test conditions

We report the performance on WSJ and REVERB test sets in Table 2 and Table 3. In this setting, if we are aware of the test domain, we expect that the best performance should be achieved by averaging *only* the two in-domain ASR models (SAME WEIGHT). The first row in Table 2 and the second row in Table 3 show that using both VAE INP and VAE ASR approach, we get similar results to SAME WEIGHT satisfying evaluation condition 1.

However, practically we won't be aware of the test condition and we won't have the luxury of cherry-picking the two in-domain ASRs. In that case, both VAE INP and VAE ASR perform better than SAME WEIGHT and BEST SINGLE approaches. The performance of both VAE INP and VAE ASR approaches is comparable on WSJ, while VAE INP works better on REVERB test set. Note that on REVERB data, using VAE INP approach with all 4 models is even better than taking an average of two in-domain ASRs.

To conclude, for neat in-domain test conditions, VAE INP comes out triumphant overall, although VAE ASR achieves comparable performance under most conditions.

Table 2: WER of combination of multiple ASR models on WSJ test set [%].

WSJ				
Approach Models	VAE	VAE	SAME	BEST
	INP	ASR	WEIGHT	SINGLE
W1+W2	5.5	5.5	5.5	6.2
R1+R2	31.5	31.8	31.3	38.5
W1+R1	6.3	6.4	7.9	6.3
W1+W2+R1	5.7	5.7	6.3	6.2
W1+W2+R1+R2	5.7	5.6	7.0	6.2

5.2. Decoding on out-of-domain datasets

The real test of our approach is when the test conditions are unknown to the trained classifiers. Decoding result for out-of-

Table 3: WER of combination of multiple ASR models on REVERB test set [%].

REVERB				
Approach	VAE INP	VAE ASR	SAME WEIGHT	BEST SINGLE
W1+W2	67.5	66.9	66.8	69.2
R1+R2	33.8	34.1	33.7	41.6
W1+R1	38.5	38.4	39.5	43.2
W1+W2+R1	38.2	40.2	43.8	43.2
W1+W2+R1+R2	32.2	34.1	35.8	41.6

domain datasets are shown in Table 4 and Table 5. Simply put, we can see that VAE ASR satisfies all 3 evaluation conditions for both the out-of-domain test sets.

On the other hand, VAE INP produces significantly worse results compared to SAME WEIGHT. This is due to the fact that the scores produced by VAE INP are not reliable estimates of the operational conditions of different classifiers and in fact are usually overestimated for R1 and R2 as compared to W1 and W2. This observation is explainable because these two out-of-domain data sets have more relatable characteristics to noisy and reverberant REVERB data set as compared to WSJ.

Table 4: WER of combination of multiple ASR models on Aurora-4 test set [%].

AURORA-4				
Approach	VAE INP	VAE ASR	SAME WEIGHT	BEST SINGLE
W1+W2	20.6	20.3	20.4	21.9
R1+R2	42.7	42.5	42.2	51.0
W1+R1	24.5	20.7	21.7	21.9
W1+W2+R1	22.8	19.6	20.2	21.9
W1+W2+R1+R2	24.7	19.6	20.7	21.9

Table 5: WER of combination of multiple ASR models on CHiME test set [%].

CHiME-4				
Approach	VAE INP	VAE ASR	SAME WEIGHT	BEST SINGLE
W1+W2	38.6	37.8	38.3	39.8
R1+R2	61.5	61.9	61.3	71.0
W1+R1	49.0	37.7	38.2	39.8
W1+W2+R1	47.9	36.5	37.3	39.8
W1+W2+R1+R2	48.8	36.1	37.4	39.8

To show that scores from VAE ASR provide more reliable estimates of the performance of an ASR, we illustrate the correlation between average scores from VAE INP and VAE ASR with the normalized ASR WER in Figure 1 and Figure 2. Each one of the 4 points on one curve corresponds to the performance on one test set. The horizontal axis represents normalized⁶ ASR

⁶The leftmost and rightmost points for each classifier correspond to

WER reported in Table 1 and the vertical axis shows the average VAE score for this test set.

Figure 2 clearly shows a stronger negative correlation between average VAE scores and normalized WER for VAE ASR as compared to VAE INP - suggesting that VAE ASR score is indeed a better estimator of the ASR performance.

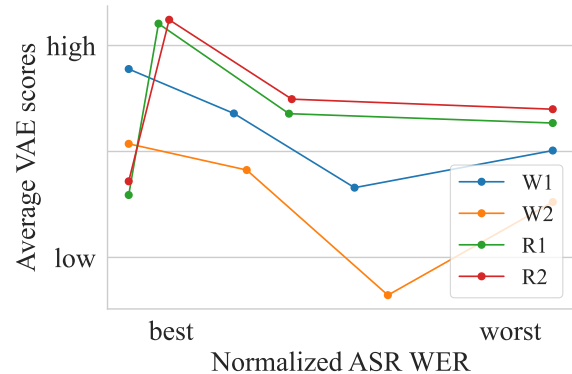


Figure 1: Correlation between average score from VAE INP and normalized WER.

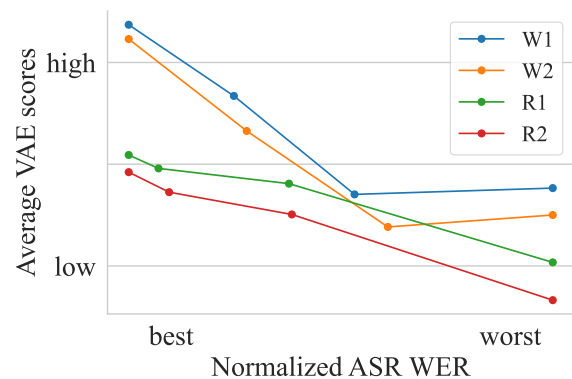


Figure 2: Correlation between average score from VAE ASR and normalized WER.

6. Conclusion

Good performance on test conditions similar to that of the training data is the primary requirement for most machine learning tasks. In our Continual Learning approach, this requirement can be sufficiently fulfilled with an ASR model combination strategy using scores from VAEs trained on input data. However, for the general scenario of an unknown out-of-domain test condition, our old combination strategy falls short. The similarity between the unknown test data under the evaluation and train data measured by such VAE models do not necessarily carry information about how well a particular ASR is operating. Instead, we propose a model combination strategy based on scores from VAEs trained on ASR encoder representation and demonstrate that these scores are better correlated to the actual ASR performance and can be used to generalize better to unknown test conditions.

7. Acknowledgement

This work was supported by a gift from Amazon.com, Inc. and the Center of Excellence in Human Language Technologies, The Johns Hopkins University.

the value of its minimal and maximal WER.

8. References

- [1] M. Karafiát, M. K. Baskar, S. Watanabe, T. Hori, M. Wiesner, J. Černocký *et al.*, “Analysis of multilingual sequence-to-sequence speech recognition systems,” *arXiv preprint arXiv:1811.03451*, 2018.
- [2] J. Barker, M. Cooke, and P. D. Green, “Robust asr based on clean speech models: an evaluation of missing data techniques for connected digit recognition in noise.” in *INTERSPEECH*, 2001, pp. 213–217.
- [3] J. Rajnoha, “Multi-condition training for unknown environment adaptation in robust asr under real conditions,” *Acta Polytechnica*, vol. 49, no. 2, 2009.
- [4] I. Kraljevski, F. Duckhorn, M. Wolff, and R. Hoffmann, “Multi-condition training and adaptation for noise robust speech recognition,” 08 2012.
- [5] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, “Continual lifelong learning with neural networks: A review,” *Neural Networks*, 2019.
- [6] Z. Li and D. Hoiem, “Learning without forgetting,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.
- [7] R. Aljundi, “Continual learning in neural networks,” *arXiv preprint arXiv:1910.02718*, 2019.
- [8] H.-J. Chang, H.-y. Lee, and L.-s. Lee, “Towards lifelong learning of end-to-end asr,” *arXiv preprint arXiv:2104.01616*, 2021.
- [9] S. Vander Eeckt *et al.*, “Continual learning for monolingual end-to-end automatic speech recognition,” *arXiv e-prints*, pp. arXiv–2112, 2021.
- [10] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, “Progressive neural networks,” *arXiv preprint arXiv:1606.04671*, 2016.
- [11] S. Sadhu and H. Hermansky, “Continual Learning in Automatic Speech Recognition,” in *Proc. Interspeech 2020*, 2020, pp. 1246–1250.
- [12] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2013.
- [13] T. Hori, S. Watanabe, and J. R. Hershey, “Joint ctc/attention decoding for end-to-end speech recognition,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 518–529.
- [14] K. Kinoshita, M. Delcroix, T. Yoshioka, T. Nakatani, E. Habets, R. Haeb-Umbach, V. Leutnant, A. Sehr, W. Kellermann, R. Maas *et al.*, “The REVERB challenge: A common evaluation framework for dereverberation and recognition of reverberant speech,” in *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. IEEE, 2013, pp. 1–4.
- [15] N. Parihar, J. Picone, D. Pearce, and H. G. Hirsch, “Performance analysis of the aurora large vocabulary baseline system,” in *2004 12th European Signal Processing Conference*, 2004, pp. 553–556.
- [16] E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, “An analysis of environment, microphone and data simulation mismatches in robust speech recognition,” *Computer Speech & Language*, vol. 46, pp. 535–557, 2017.
- [17] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen *et al.*, “Espnet: End-to-end speech processing toolkit,” *arXiv preprint arXiv:1804.00015*, 2018.
- [18] S. Sadhu and H. Hermansky, “Radically Old Way of Computing Spectra: Applications in End-to-End ASR,” in *Proc. Interspeech 2021*, 2021, pp. 1424–1428.