Continued Finetuning as Single Speaker Adaptation

Jim O'Regan
Speech, Music & Hearing, KTH Royal Institute of Technology, Sweden
joregan@kth.se

Abstract

The adaptation of unsupervised learning techniques to speech recognition have enabled the training of accurate models with less labelled training data, by finetuning a supervised classifier on top of a network pretrained using self-supervised methods. In this paper, we investigate if continuing the fine-tuning of such a model is suitable as a method of speaker adaptation for a single speaker, considering two kinds of user: the casual user, with data measurable in minutes, and the professional user, with data measurable in hours. We conduct experiments across a range of dataset sizes, in an attempt to provide a basis for estimates on how much data would be needed.

Introduction

Modern speech recognition has benefitted from advances in deep learning, to the extent that it is now ubiquitous, powering phones, computers, and a range of appliances from smart televisions to smart speakers.

This has been enabled by initiatives to increase the availability of data, from Librivox, a platform for user-created audiobooks which has been used as a source for several speech corpora, such as Librispeech (Panayotov et al., 2015) and Libri-Light (Kahn et al., 2020), to Common Voice (Ardila et al., 2020), which collects data specifically for speech recognition, in a range of languages.

Despite impressive advances in the accuracy of speech recognition systems in recent years, the fact that a system achieves a certain level of accuracy on a test set is of little interest to an individual user whose own results are poor, and even in speech recognition systems using newer methods, performance tends to decrease on an unseen speaker (Meng et al., 2019).

Applications in which speech recognition is a component, such as digital assistants like Siri or Alexa, can typically have a level of robustness to the accuracy of speech recognition, either through task-specific, limited vocabulary and grammar, or by directly modelling errors. Tasks which involve dictation, on the other hand, because of the freer nature of their input, tend not to have a level of correction beyond that which can be provided by a language model, which, despite recent advances, tend to perform less well when there is a higher degree of novelty in their input.

With speaker adaptation, speech recognition can achieve good enough results for dictation: the British author Terry Pratchett notably made use of speech recognition to write his final books after early-onset Alzheimer's left him unable to type (Flood, 2011), while respeakers for subtitles of live programming are used throughout Europe (Romero-Fresco, 2018).

In this work, we attempt determine if continued fine tuning of a publicly available model is viable as a single speaker adaptation method, considering two in-domain cases: the low resource case of a casual user, who may only have minutes of transcribed speech available; and the higher resource case of a professional user, who can be expected to have hours of speech. We also consider the out-of-domain case of accented speech, again in a low resource situation.

We aim to provide information as a basis for estimates; to give some intuition into the trade-offs between available data and training time, particularly in low-resource situations.

Unsupervised learning

In recent years, it has become common to initialise neural networks by *pretraining* on a general task, which is later *finetuned* on the desired task by freezing the earlier layers of the network, and training only the final, classifier layer, which drastically reduces the amount of computation and task-specific data required, while typically improving performance.

In computer vision, it has long been common to pretrain on a large dataset, such as ImageNet (Vinyals et al., 2017), though these tasks were initially supervised. Transformer-based models in NLP, such as BERT (Devlin et al., 2019), achieved great successes by using an unsupervised task or tasks (gap filling and next sentence prediction, in the case of BERT), which allows the network to learn a generalised representation of the input without requiring labelled data, which can then be fine-tuned for a more specific task. This type of unsupervised pretraining was applied to speech recognition in Schneider et al., (2019) with the wav2vec model, in which the model learned features directly from the audio using an unsupervised contrastive task.

Frameworks for deep learning typically employ checkpoints, with the ability to resume from a previous checkpoint, for a range of purposes. In the simplest case, they allow resumption of training in the event of system failure; they are also used as the basis for *early stopping*, where at least two checkpoints are saved: one containing the most recent update, the other containing the best. If, after a specified number of updates, no improvement has been made over the best results, the training process exits, rather than continuing. Continued fine tuning has been shown to be useful as a method of domain adaptation (Xu et al., 2021).

Speaker Adaptive Training

Older systems based on hidden Markov models (HMM) typically included facilities for speaker adaptation, for example, using maximum likelihood linear regression (MLLR) (Leggetter & Woodland, 1994); consumer speech recognition systems typically included such facilities, so the accuracy of the recognition of the user's speech increased as the application was used (e.g., Nuance, 2014).

Experiments

Data

For in-domain experiments we made use of the LJSpeech dataset, version 1.1 (Ito & Johnson, 2017), which

contains approximately 24 hours of speech of a single speaker with a General American accent, taken from LibriVox. For accented speech, we used the AWB (Scottish) voice data from the CMU Arctic collection (Kominek & Black, 2004).

After extracting 5% each of the IDs for test and validation, for each split we greedily took IDs until adding the next would exceed the amount of the split; the remainder was then searched for the longest single segment that can be added.

For the casual user case, we took 5-minute increments, from 5 minutes to 120; for the professional case, 2-hour increments until 16 hours. For the accented data, as the amount of available data was less than 2 hours, we took 5-minute increments from 5 minutes to 60.

Models

We used the wav2vec 2.0 base models from the fairseq repository[1]; the "No finetuning" model was used as the pretrained base, while finetuning was continued from the "960 hours" split.

Hyperparameters were maintained as in the "960 split" model; the only script parameters that were changed pertained to distributed processing, or to memory use. All models were finetuned on a single NVIDIA GeForce RTX 3090 with 24Gb of RAM.

For the specific case of a casual user, we aimed at a scenario where free GPU access, such as that provided by Google Colab[2], would be employed. Although the documentation claims that up to 12 hours of GPU time may be available[3], in practice, we have found the limit to fall between 3 and 5 hours and aimed at setting early stopping so that training would complete within this range. We ran training on a set of splits from 5 to 40 minutes for 6 hours and chose 300 epochs as matching the best results from the majority of these.

Results

Low resource, in-domain

The finetuned model, "960 hours", without continued finetuning, achieved a word error rate (WER) of 5.4%. Results for the low data use case, from 5 minutes to 120 minutes, are collected in Table 1 and visualised in Figure 1. The result for the largest amount of data shows a relative improvement over the baseline of over 50%.

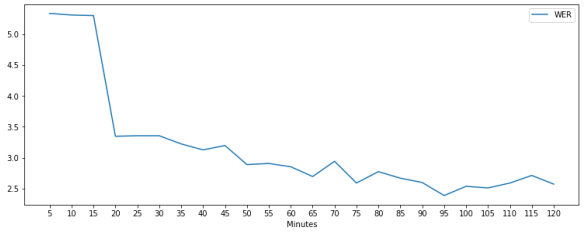

Figure 1: WER on 5 to 120 minute splits on LJSpeech.

Although the reduction in WER as the amount of data increased is not smooth, it does show a general trend, but the sharp drop from 15 minutes to 20 does not quite fit. We repeated the finetuning for the lowest amounts of data, 5 minutes to 30, using a larger number of epochs for early stopping: 1000 epochs. The results are compared with their 300-epoch counterparts in Table 2, and visualised in Figure 2.

Table 1: Results of LJSpeech splits, with times in minutes, for early stopping set to 300. The baseline model achieved 5.4%.

Time	WER	Time	WER	Time	WER
5	5.34	45	3.20	85	2.67
10	5.31	50	2.89	90	2.60
15	5.30	55	2.91	95	2.39
20	3.35	60	2.85	100	2.54
25	3.36	65	2.69	105	2.51
30	3.36	70	2.94	110	2.59
35	3.22	75	2.59	115	2.71
40	3.13	80	2.77	120	2.57

Table 2: Results of LJSpeech splits from 5 to 30 minutes, along with training times, for early stopping set to 300 and 1000 epochs.

Minutes	300		1000	
	WER	Time	WER	Time
5	5.34	10912.7	4.11	42180.1
10	5.31	10540.1	4.08	32147.6
15	5.30	8515.5	3.72	28848.0
20	3.35	16195.9	3.49	27926.7
25	3.36	14774.0	3.43	26710.9
30	3.36	14082.0	3.34	25955.9

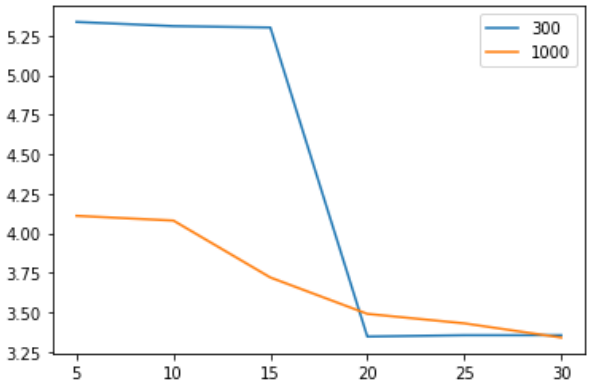

Figure 2: WER on LJSpeech for 5 to 30 minute splits, indicated by "Minutes", with early stopping at 300 and 1000 epochs. Training time in seconds is provided in the columns marked "Time".

Higher resource, in-domain

The higher resource splits showed similar results to the low resource setting, with the largest amount of data producing WER of 1.03. Results are collected in Table 3 and visualised in Figure 3.

Table 3: Results of LJSpeech splits from 2 to 16 hours.

Time	WER	Time	WER
2	2.57	10	1.38
4	1.47	12	1.22
6	1.66	14	1.13
8	1.47	16	1.03

[1] https://github.com/pytorch/fairseq/tree/main/examples/wav2vec

[2] https://colab.research.google.com/

[3] https://research.google.com/colaboratory/faq.html

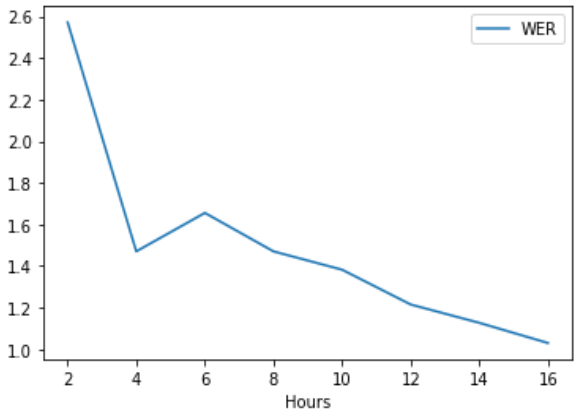

Figure 3: WER of LJSpeech splits from 2 to 16 hours.

Low resource, out-of-domain

The finetuned model, "960 hours", without continued finetuning, achieved a word error rate (WER) of 5.77% on the out-of-domain (differently accented) data. The results of the out-of-domain data are collected in Table 4 and visualized in Figure 4.

Table 4: Results of CMU AWB data (Scottish accent) in splits of 5 to 60 minutes.

Time	WER	Time	WER
5	5.77	35	3.65
10	5.77	40	3.65
15	5.58	45	2.50
20	4.62	50	3.65
25	3.85	55	2.69
30	4.62	60	3.27

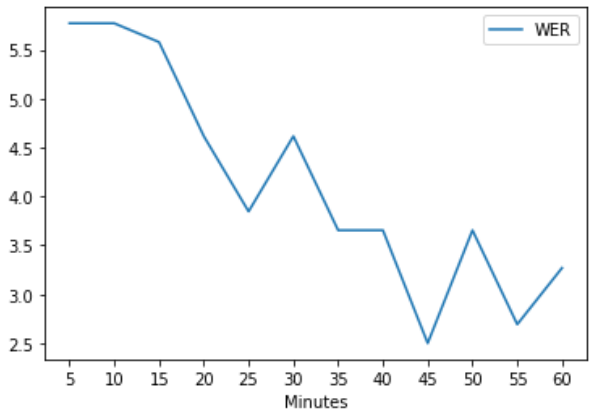

Figure 4: WER on 5 to 60 minute splits on CMU AWB data.

Discussion

Our results suggest that continuing finetuning is a viable method for single speaker adaptation.

There is much to recommend continued finetuning as an adaptation method: it is effectively an "off-the-shelf" method, relying on nothing more than extra data and the checkpointing facilities built in to deep learning frameworks, without requiring any modification to the software or specially trained models.

One drawback of continued finetuning is the possibility of *catastrophic forgetting*, where the network adapts to the new data at the cost of losing the ability to process the old. In the particular case of single speaker adaptation, this would typically not be considered a problem, but end users ought to be made aware that the adaptation may render the model less usable for other speakers.

The number of epochs used for early stopping ought to be balanced against the amount of training data: in the lowest cases of 5 and 10 minutes, we saw little to no improvement over the baseline, which was improved in the LJSpeech case by raising the number of epochs; in these cases, however, a better use of time might be to simply record or annotate more data and rerun the adaptation with this extra data.

In low resource settings, if more than one GPU is available, it may be worthwhile to remove some data, as when neighbouring splits produced the same WER, this seemed to indicate that it would be worth continuing training for longer, though our results are not extensive or robust enough to confirm this with any great certainty.

One particular weakness in our experiments was the use of unrealistically large validation sets, which may have skewed the results. In low resource settings like the ones we explored here, it would have been more conventional to not use a validation set, to not reduce the amount of training data. Although we saw nothing in the training loss to indicate a potentially different outcome, it would be better to more realistically match the conditions we expect in a low resource setting.

That the WER from the baseline model on the differently accented data was not significantly different from the in-domain data shows that our results are not particularly indicative: we would need to check again with data that performs less well with the baseline model to be able to draw any conclusions regarding accent.

Conclusions

In this paper, we have attempted to establish whether continued finetuning of an existing model could be suitable as an adaptation method for a single speaker, across a range of dataset sizes, ranging from minutes to hours, to provide a basis for estimates into the amount of data required. We find that amounts as low as 20-30 minutes can provide a notable decrease in word error rate with 3-4 hours of training, while with 16 hours, the word error rate can be brought as low as 1.03%. With increased training time, even amounts of data as low as 5-10 minutes can provide a noticeable reduction in WER.

References

Ardila, R., Branson, M., Davis, K., Kohler, M., Meyer, J., Henretty, M., Morais, R., Saunders, L., Tyers, F., & Weber, G. (2020). Common Voice: A Massively-Multilingual Speech Corpus. *Proceedings of the 12th Language Resources and Evaluation Conference*, 4218–4222. https://aclanthology.org/2020.lrec-1.520

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *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. https://doi.org/10.18653/v1/N19-1423

Flood, A. (2011, October 14). A life in writing: Terry Pratchett. *The Guardian*. https://www.theguardian.com/culture/2011/oct/14/terry-pratchett-life-writing

Ito, K., & Johnson, L. (2017). *The LJ Speech Dataset*. https://keithito.com/LJ-Speech-Dataset/

Kahn, J., Rivière, M., Zheng, W., Kharitonov, E., Xu, Q., Mazaré, P. E., Karadayi, J., Liptchinsky, V., Collobert, R., Fuegen, C., Likhomanenko, T., Synnaeve, G., Joulin, A., Mohamed, A., & Dupoux, E. (2020). Libri-Light: A Benchmark for ASR with Limited or No Supervision. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 7669–7673.

Kominek, J., & Black, A. W. (2004). The CMU Arctic speech databases. *SSW5-2004*, 223–224.

Leggetter, C. J., & Woodland, P. C. (1994). *Speaker adaptation of HMMs using linear regression* (Technical Report CUED/F-INFENG/TR. 181). Cambridge University Engineering Department.

Meng, Z., Gaur, Y., Li, J., & Gong, Y. (2019, September). Speaker Adaptation for Attention-Based End-to-End Speech Recognition. *Interspeech 2019*. https://doi.org/10.21437/interspeech.2019-3135

Nuance. (2014). *Dragon NaturallySpeaking 13 Installation Guide and User Guide*.

Panayotov, V., Chen, G., Povey, D., & Khudanpur, S. (2015). Librispeech: An ASR corpus based on public domain audio books. *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5206–5210. https://doi.org/10.1109/ICASSP.2015.7178964

Romero-Fresco, P. (2018). Respeaking: Subtitling through Speech Recognition. In L. Pérez-González (Ed.), *The Routledge Handbook of Audiovisual Translation*. Routledge.

Schneider, S., Baevski, A., Collobert, R., & Auli, M. (2019). *wav2vec: Unsupervised Pre-training for Speech Recognition*. arXiv. https://doi.org/10.48550/ARXIV.1904.05862

Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2017). Show and Tell: Lessons Learned from the 2015 MSCOCO Image Captioning Challenge. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, *39*(04), 652–663. https://doi.org/10.1109/TPAMI.2016.2587640

Xu, H., Ebner, S., Yarmohammadi, M., White, A. S., Van Durme, B., & Murray, K. (2021). Gradual Fine-Tuning for Low-Resource Domain Adaptation. *Proceedings of the Second Workshop on Domain Adaptation for NLP*, 214–221. https://www.aclweb.org/anthology/2021.adaptnlp-1.22