Coarse-To-Fine And Cross-Lingual ASR Transfer

Peter Polák, Ondřej Bojar

Charles University
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
<polak,bojar>@ufal.mff.cuni.cz

Abstract: End-to-end neural automatic speech recognition systems achieved recently state-of-the-art results but they require large datasets and extensive computing resources. Transfer learning has been proposed to overcome these difficulties even across languages, e.g., German ASR trained from an English model. We experiment with much less related languages, reusing an English model for Czech ASR. To simplify the transfer, we propose to use an intermediate alphabet, Czech without accents, and document that it is a highly effective strategy. The technique is also useful on Czech data alone, in the style of coarse-to-fine training. We achieve substantial reductions in training time as well as word error rate (WER).

1 Introduction

Contemporary end-to-end, deep-learning automatic speech recognition systems achieved state-of-the-art results on many public speech corpora, see e.g. Chiu et al. [2], Park et al. [20], Han et al. [8].

To outperform traditional hybrid models, deep-learning ASR systems must be trained on vast amounts of training data in the order of a thousand hours. Currently, there is only a limited number of public datasets that meet these quantity criteria. The variety of covered languages is also minimal. In fact, most of these large datasets contain only English [24]. Although new speech datasets are continually emerging, producing them is a tedious and expensive task.

Another downside of new end-to-end speech recognition systems is their requirement of an extensive computation on many GPUs, taking several days to converge, see, e.g., Karita et al. [10].

These obstacles are often mitigated with the technique of transfer learning [22] when a trained model or a model part is reused in a more or less related task. Furthermore, it became customary to publish checkpoints alongside with the neural network implementations and there emerge repositories with pre-trained neural networks such as *TensorFlow Hub*[1] or *PyTorch Hub*.[2] This allows us to use pre-trained models, but similarly, most of the published checkpoints are trained for English speech.

[1] https://tfhub.dev/

[2] https://pytorch.org/hub/

In our work, we propose a cross-lingual coarse-to-fine intermediate step and experiment with transfer learning [22], i.e., the reuse of pre-trained models for other tasks. Specifically, we reuse the available English ASR checkpoint of QuartzNet [14] and train it to recognize Czech speech instead.

This paper is organized as follows. In Section 2, we give an overview of related work. In Section 3, we describe the used models and data. Our proposed method is described in Section 4, and the results are presented and discussed in Section 5. Finally, in Section 6 we summarize the work.

2 Related Work

Transfer learning [22] is an established method in machine learning because many tasks do not have enough training data available, or they are too computationally demanding. In transfer learning, the model of interest is trained with the help of a more or less related "parent" task, reusing its data, fully or partially trained model, or its parts.

Transfer learning is gradually becoming popular in various areas of NLP. For example, transferring some of the parameters from parent models of high-resource languages to low-resource ones seem very helpful in machine translation [26] even regardless the relatedness of the languages [12].

Transfer learning in end-to-end ASR is studied by Kunze et al. [16]. They show that (partial) cross-lingual model adaptation is sufficient for obtaining good results. Their method exploits the layered structure of the network. In essence, they take an English model and freeze weights in the upper part of the network (closer to the input). Then they adapt the lower part for German speech recognition yielding very good results while reducing training time and the amount of needed transcribed German speech data.

More recent study on cross-lingual ASR transfer is Huang et al. [9]. They start with an English QuartzNet [14] model trained on more than three thousand hours of transcribed speech. Further, they fine-tune the model on German, Spanish and Russian. Because the languages have a different alphabet, the authors randomly initialize a new shallow decoder. Compared with the baseline (trained from scratch), the authors report a substantial reduction in terms of WER ranging from 20 % relative for German, 25 % for Spanish and even 46 % for Russian.

Other works concerning end-to-end ASR are Tong et al.

[23] and Kim and Seltzer [11]. The former proposes unified IPA-based phoneme vocabulary while the latter suggests a universal character set. The first demonstrates that the model with such an alphabet is robust to multilingual setup and transfer to other languages is possible. The latter proposes language-specific gating enabling language switching that can increase the network's power.

Multilingual transfer learning in ASR is studied by Cho et al. [3]. First, they jointly train one model (encoder and decoder) on ten languages (approximately 600 hours in total). Second, they adapt the model for a particular target language (4 languages, not included in the previous 10, with 40 to 60 hours of training data). They show that adapting both encoder and decoder boosts the performance in terms of character error rate.

Coarse-to-fine processing [21] has a long history in NLP. It is best known in the parsing domain, originally applied for the surface syntax [1] and more recently for neural-based semantic parsing [5]. The idea is to train a system on a simpler version of the task first and then gradually refine the task up to the desired complexity. With neural networks, coarse-to-fine training can lead to better internal representation, as e.g., Zhang et al. [25] observe for neural machine translation.

The term coarse-to-fine is also used in the context of hyperparameter optimization, see, e.g., Moshkelgosha et al. [18] or the corresponding DataCamp class,[3] to cut down the space of possible hyperparameter settings quickly.

Our work is novel and differs from the above-mentioned ones in two ways: First, we reuse existing models and checkpoints to improve the speed of training and ASR accuracy for an unrelated language.[4] Second, in the coarse-to-fine method, we simplify (instead of unifying) the Czech character set to improve cross-lingual transfer and enhance monolingual training significantly.

3 Data and Models Used

3.1 Model architecture

We use the QuartzNet [14] neural network. It is an end-to-end, convolutional neural network trained with CTC (Connectionist Temporal Classification) [7], based on a larger network Jasper [17]. While performing only slightly worse then the larger Jasper, it has only a fraction of parameters (18.9 million versus 333 million).

The model input is 64 MFCC (mel-frequency cepstrum coefficient) features computed from 20 ms windows with an overlap of 10 ms. For a given time step, the model outputs probability over the given alphabet. The model starts

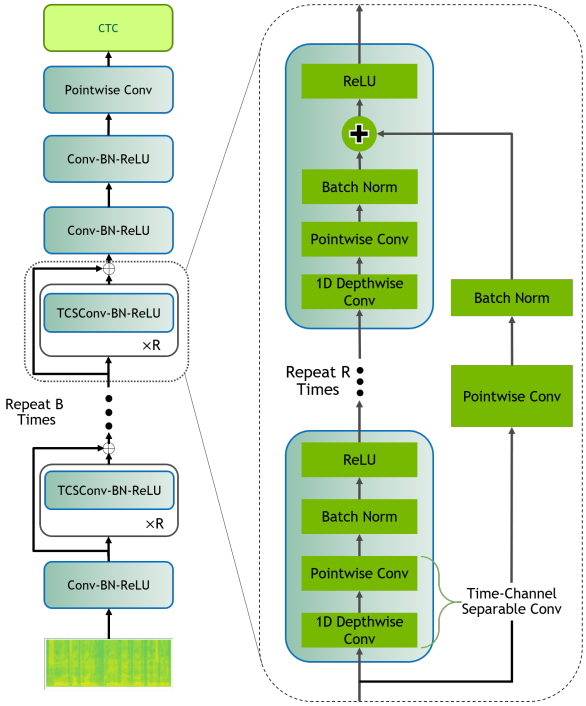

Figure 1: QuartzNet BxR architecture. Taken from Kriman et al. [14]. First layer from the input is C_1 layer, the last three layers are C_2, C_3 and C_4. From input up to the C_3 layer is encoder, C_4 layer is decoder.

Block	R	K	c_{out}	S
C_1	1	33	256	1
B_1	5	33	256	3
B_2	5	39	256	3
B_3	5	51	512	3
B_4	5	63	512	3
B_5	5	75	512	3
C_2	1	87	512	1
C_3	1	1	1024	1
C_4	1	1	||labels||	1

Table 1: QuartzNet 15x5 — 5 block types (B_1..B_5) x 3 block repeats (column S) = 15 x 5 (column R) summarized. Each block consists of R K-sized modules. Each block is repeated S times. Horizontal line marks the encoder and decoder parts of the networks.

with a 1D time-channel separable convolutional layer C_1 with a stride of 2. This layer is then followed by B blocks. Each i-th block consists of the same modules repeated R_i times. Each block is repeated S_i times. The basic module has four layers: K-sized depthwise convolution layer with c_{out} channels, a pointwise convolution, a normalization layer and activation (ReLU). Next follows a time-channel separable convolution C_2 and two 1D convolutional layers C_3 and C_4 (with dilatation of 2). The part of the network from C_1 to C_3 is encoder and layer C_4 is decoder.

In our work, we use QuartzNet 15x5. This model has

[3] `https://campus.datacamp.com/courses/hyperparameter-tuning-in-python/informed-search?ex=1`

[4] In contrast to the Germanic English, Czech is a Slavic language with rich morphology and relatively free word order. To the best of our knowledge, the phonetic similarity of Czech and English has not been rigorously studied, although the common belief is that Czech is more phonetically consistent.

5 blocks, each repeated $S_i = 3$ times and each module repeated $R_i = 5$ times. The model schema is in Figure 1 and summarized in Table 1.

3.2 Training

We work with the original implementation in the NeMo toolkit [15]. The training is performed on 10 NVIDIA GeForce GTX 1080 Ti GPUs with 11 GB VRAM. We use the `O1` optimization setting, which primarily means mixed-precision training (weights are stored in single precision, gradient updates are computed in double precision). Batch size is 32 per GPU ($10 \times 32 = 320$ global batch size). We use warm-up of 1000 steps. The learning rate for training is 0.01 and 0.001 for the fine-tuning. The optimizer is NovoGrad [6] with weight decay 0.001 and $\beta_1 = 0.95$ and $\beta_2 = 0.5$. Additionally, we use Cutout [4] with 5 masks, maximum time cut 120 and maximum frequency cut 50.

3.3 Pre-Trained English ASR

As the parent English model, we use the checkpoint available at the *NVIDIA GPU Cloud*.[5] It is trained on LibriSpeech [19] and Mozilla CommonVoice[6] for 100 epochs on 8 NVIDIA V100 GPUs. The model achieves 4.19 % WER on LibriSpeech test-clean using greedy decoding without language model.

3.4 Czech Speech Data

In our experiments, we use Large Corpus of Czech Parliament Plenary Hearings [13]. At the time of writing, it is the most extensive available speech corpus for the Czech language, consisting of approximately 400 hours.

The corpus includes two held out sets: the development set extracted from the training data and reflecting the distribution of speakers and topics, and the test set which comes from a different period of hearings. We choose the latter for our experiments because we prefer the more realistic setting with a lower chance of speaker and topic overlap.

4 Examined Configurations

Figure 2 presents the examined setups. In all cases, we aim at the best possible Czech ASR, disregarding the model's performance in the original English task. The baseline (not in the diagram) is to train the network from scratch on the whole Czech dataset, converting the speech signal directly to Czech graphemes, i.e., words in fully correct orthography, except punctuation and casing which are missing in both the training and evaluation data.

[5] `https://ngc.nvidia.com/catalog/models/nvidia:quartznet15x5`

[6] `https://commonvoice.mozilla.org/en/datasets`

4.1 Basic Transfer Learning

Our first method is very similar to Kunze et al. [16] and Huang et al. [9]. We use the English checkpoint with the (English) WER of 4.19 % on LibriSpeech test-clean, and continue the training on Czech data.

The Czech language uses an extended Latin alphabet, with diacritic marks (acute, caron, and ring) added to some letters. The Czech alphabet has 42 letters, including the digraph "ch". Ignoring this digraph (it is always written using the letters "c" and "h"), we arrive at 41 letters. Only 26 of them are known to the initial English decoder.

To handle this difference, we use a rapid decoder adaptation (unlike Huang et al. [9]). For the first 1500 steps, we keep the encoder frozen and only train the decoder (randomly initialized; Glorot uniform).

Subsequently, we unfreeze the encoder and train the whole network on the Czech dataset.

4.2 Transfer Learning with Vocabulary Simplification

In this experiment, we try to make the adaptation easier by first keeping the original English alphabet and extending it to the full Czech alphabet only once it is trained.

To coerce Czech into the English alphabet, it is sufficient to strip diacritics (e.g. convert "čárka" to "carka"). This simplification is quite common in Internet communication but it always conflates two sounds ([ts] written as "c" and [tʃ] written as "č") or their duration ([aː] for "á" and [a] for "a").

In this experiment, we first initialize both encoder and decoder weights from the English checkpoint (English and simplified Czech vocabularies are identical so the decoder dimensions match), and we train the network on the simplified Czech dataset for 39 thousand steps.

The rest (adaptation and training on the full Czech alphabet) is the same as in Section 4.1. Overall, this can be seen as a simple version of coarse-to-fine training where a single intermediate model is constructed with a reduced output alphabet.

4.3 Vocabulary Simplification Only

In this experiment, we first simplify target vocabulary: we use standard Latin alphabet with 26 letters plus space and apostrophe (to preserve compatibility with English). Czech transcripts are then encoded using this simplified alphabet (e.g. "čárka" as "carka").

With transcripts encoded in this manner, we train a randomly (Glorot uniform) initialized QuartzNet network for 39 thousand steps.

From our previous experience with vocabulary adaptation, we make a short adaptation of the model for a different alphabet. We initialize the encoder with weights obtained in the previous step and modify the target vocabulary to all Czech letters (41 plus space and apostrophe).

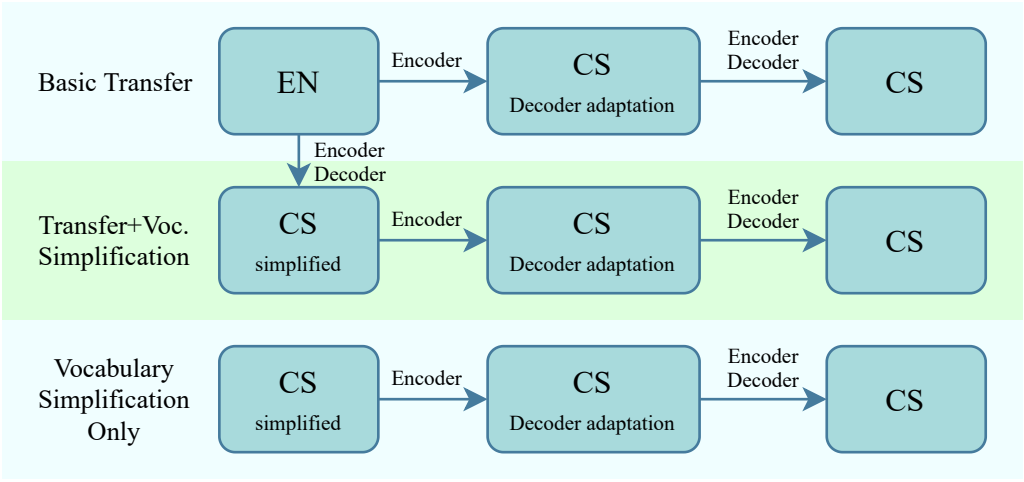

Figure 2: Examined setups of transfer learning. The labels on the arrows indicate which model parts are transferred, i.e., used to initialize the subsequent model. Parameter freezing is involved only for the encoder weights in the "CS decoder adaptation" phase.

Experiment	Simplified	Adaptation	Full
Baseline CS	-	-	20.19
EN → CS	-	97.35	19.06
EN → sim. CS → CS	17.11	22.01	16.88
sim. CS → CS	20.56	24.59	16.57

Table 2: Results in % of word error rate on the Czech (CS) test set. "Simplified" column reflects WER after the training on simplified dataset (both training and test data without accents). "Adaptation" column contains WER immediately after the decoder adaptation phase to the full Czech alphabet including accents. Finally, "Full" column contains performance on the test set with accents after the full training.

The decoder is initialized with random weights. We freeze the encoder and train this network shortly for 1500 steps. Note that the original decoder for simplified Czech is discarded and trained from random initialization in this adaptation phase.

After this brief adaptation step, we unfreeze the encoder and train the whole network for 39 thousand steps.

5 Results and Discussion

Table 2 presents our final WER scores and Figure 3 shows their development through the training. For simplicity, we use greedy decoding and no language model. We take the model from the last reached training iteration. Arguably, some of the setups are not yet fully converged but the main goal of this paper is to propose solutions for situations where hardware resources are capped (as was our case, too). With unlimited training time available, the results might differ. Little signs of overfitting are apparent for the "Simplified CS → CS" setup. An earlier iteration of this setup might have worked better, but we do not have another test set with unseen speakers to validate it. The development set of the Czech speech corpus is rather small (3 hours), so we prefer not to split it.

Without an independent test set and computing capacity to reach full convergence or overfitting for all the models, our analysis has to focus on the development of model performance in time (Figure 3) rather than on the final performance of models chosen by an automatic stopping criterion on such a held-out test set.

5.1 Transfer across Unrelated Languages

We observe that initialization with an unrelated language helps to speed up training.

This is best apparent by comparing the first 39k steps of the learning curves for "English → Czech simplified" and "Czech (simplified) → Czech" in Figure 3, thin lines. Here the target alphabet was simplified in both cases, but the weights initialized from English allow a much faster decrease of WER. The benefit is also clear for the full alphabet (baseline vs. "English → Czech"), where the baseline has a consistently higher WER.

The training of the English parent is so fast that WER for the simplified alphabet (thin dashed green line) drops under 30 % within the first 2000 steps (1 hour of training). This can be particularly useful if the final task does not require the lowest possible WER, such as sound segmentation.

While Basic transfer ("English → Czech") boosts the convergence throughout the training, its final performance is only 1 to 2 % points of WER better than the baseline, see the plot or compare 20.19 with 19.06 in Table 2. The intermediate vocabulary simplification is more important and allow a further decrease of 2.2 % WER absolute (19.06 vs. 16.88) when transferring from English.

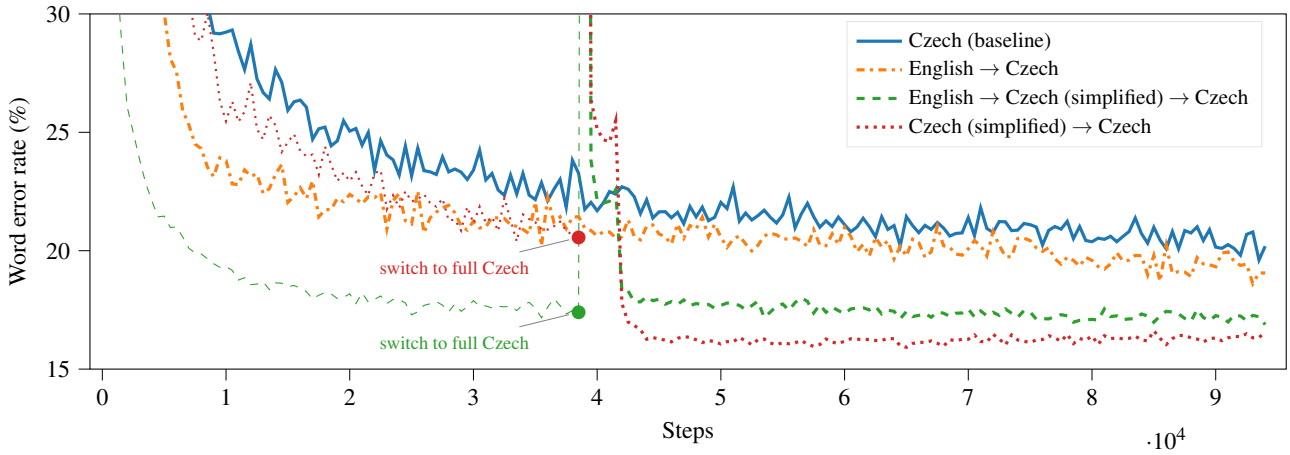

Figure 3: Evaluation on the test set during training. In setups with transfer learning, the same colour and dashing are used across training stages. Note that WER curves for experiments with simplified vocabulary (thin lines) are not directly comparable with other curves until step 39,000 as the test set is on different (simplified) vocabulary. 10,000 steps take approximately 5 hours. Better viewed in colour.

5.2 Transfer across Target Vocabularies

In the course of two experiment runs, we altered the target vocabulary: the training starts with simplified Czech, and after about 39,000 steps, we switch to the full target vocabulary. This sudden change can be seen as spikes in Figure 3. Note that WER curves before the peak use the simplified Czech reference (the same test set but with stripped diacritics), so they are not directly comparable to the rest.

The intermediate simplified vocabulary always brings a considerable improvement. In essence, the final WER is lower by 2.18 (16.88 vs. 19.06 in Table 2) for the models transferred from English and by 3.62 (16.57 vs. 20.19) for Czech-only runs. One possible reason for this improvement is the "easier" intermediate task of simpler Czech. Note that the exact difficulty is hard to compare as the target alphabet is smaller than with the full vocabulary, but more spelling ambiguities may arise. This intermediate task thus seems to help the network to find a better-generalizing region in the parameter space. Another possible reason that this sudden change and reset of the last few layers allows the model to reassess and escape a local optimum in which the "English → Czech" setup could be trapped.

6 Conclusion and Future Work

We presented our experiments with transfer learning for automated speech recognition between unrelated languages. In all our experiments, we outperformed the baseline in terms of speed of convergence and accuracy.

We gain a substantial speed-up when training Czech ASR while reusing weights from a pre-trained English ASR model. The final word error rate improves over the baseline only marginally in this basic transfer learning setup.

We are able to achieve a substantial improvement in WER by introducing an intermediate step in the style of coarse-to-fine training, first training the models to produce Czech without accents, and then refining the model to the full Czech. This coarse-to-fine training is most successful within a single language: Our final model for Czech is better by over 3.5 WER absolute over the baseline, reaching WER of 16.57%. Further gains are expected from beam search with language model or better iteration choice to avoid overfitting.

As we documented in Section 5, transfer learning leads to a substantial reduction in training time. We achieved speed-up even in unrelated languages. We also demonstrated that the coarse-to-fine approach leads not only to training time reduction but also yields better accuracy.

We see further potential in the coarse-to-fine training. We want to explore this area more thoroughly, e.g., by introducing multiple simplification stages or testing the technique on more languages.

Acknowledgements

The work was supported by the grant 19-26934X (NEUREM3) of the Czech Science Foundation and START/SCI/089 (Babel Octopus: Robust Multi-Source Speech Translation) of the START Programme of Charles University.

References

[1] Eugene Charniak, Mark Johnson, Micha Elsner, Joseph Austerweil, David Ellis, Isaac Haxton, Catherine Hill, R. Shrivaths, Jeremy Moore, Michael

Pozar, and Theresa Vu. 2006. Multilevel coarse-to-fine PCFG parsing. In *HLT-NAACL*. The Association for Computational Linguistics.

[2] Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, Ekaterina Gonina, et al. 2018. State-of-the-art speech recognition with sequence-to-sequence models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4774–4778. IEEE.

[3] Jaejin Cho, Murali Karthick Baskar, Ruizhi Li, Matthew Wiesner, Sri Harish Mallidi, Nelson Yalta, Martin Karafiat, Shinji Watanabe, and Takaaki Hori. 2018. Multilingual sequence-to-sequence speech recognition: architecture, transfer learning, and language modeling. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 521–527. IEEE.

[4] Terrance DeVries and Graham W Taylor. 2017. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*.

[5] Li Dong and Mirella Lapata. 2018. Coarse-to-fine decoding for neural semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 731–742, Melbourne, Australia. Association for Computational Linguistics.

[6] Boris Ginsburg, Patrice Castonguay, Oleksii Hrinchuk, Oleksii Kuchaiev, Vitaly Lavrukhin, Ryan Leary, Jason Li, Huyen Nguyen, Yang Zhang, and Jonathan M Cohen. 2019. Stochastic gradient methods with layer-wise adaptive moments for training of deep networks. *arXiv preprint arXiv:1905.11286*.

[7] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

[8] Kyu J Han, Ramon Prieto, and Tao Ma. 2019. State-of-the-art speech recognition using multi-stream self-attention with dilated 1d convolutions. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 54–61. IEEE.

[9] Jocelyn Huang, Oleksii Kuchaiev, Patrick O'Neill, Vitaly Lavrukhin, Jason Li, Adriana Flores, Georg Kucsko, and Boris Ginsburg. 2020. Cross-language transfer learning, continuous learning, and domain adaptation for end-to-end automatic speech recognition. *arXiv preprint arXiv:2005.04290*.

[10] Shigeki Karita, Nanxin Chen, Tomoki Hayashi, Takaaki Hori, Hirofumi Inaguma, Ziyan Jiang, Masao Someki, Nelson Enrique Yalta Soplin, Ryuichi Yamamoto, Xiaofei Wang, et al. 2019. A comparative study on Transformer vs RNN in speech applications. In *Proceedings of the ASRU 2019 IEEE Automatic Speech Recognition and Understanding Workshop*. (in print).

[11] S. Kim and M. L. Seltzer. 2018. Towards language-universal end-to-end speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4914–4918.

[12] Tom Kocmi and Ondřej Bojar. 2018. Trivial transfer learning for low-resource neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 244–252, Brussels, Belgium. Association for Computational Linguistics.

[13] Jonás Kratochvíl, Peter Polak, and Ondřej Bojar. 2020. Large corpus of czech parliament plenary hearings. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 6363–6367.

[14] Samuel Kriman, Stanislav Beliaev, Boris Ginsburg, Jocelyn Huang, Oleksii Kuchaiev, Vitaly Lavrukhin, Ryan Leary, Jason Li, and Yang Zhang. 2020. Quartznet: Deep automatic speech recognition with 1d time-channel separable convolutions. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6124–6128. IEEE.

[15] Oleksii Kuchaiev, Jason Li, Huyen Nguyen, Oleksii Hrinchuk, Ryan Leary, Boris Ginsburg, Samuel Kriman, Stanislav Beliaev, Vitaly Lavrukhin, Jack Cook, et al. 2019. NeMo: a toolkit for building AI applications using Neural Modules. *arXiv e-prints*, pages arXiv–1909.

[16] Julius Kunze, Louis Kirsch, Ilia Kurenkov, Andreas Krug, Jens Johannsmeier, and Sebastian Stober. 2017. Transfer learning for speech recognition on a budget. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 168–177, Vancouver, Canada. Association for Computational Linguistics.

[17] Jason Li, Vitaly Lavrukhin, Boris Ginsburg, Ryan Leary, Oleksii Kuchaiev, Jonathan M. Cohen, Huyen Nguyen, and Ravi Teja Gadde. 2019. Jasper: An End-to-End Convolutional Neural Acoustic Model. In *Proc. Interspeech 2019*, pages 71–75.

[18] V. Moshkelgosha, H. Behzadi-Khormouji, and M. Yazdian-Dehkordi. 2017. Coarse-to-fine parameter tuning for content-based object categorization.

In *2017 3rd International Conference on Pattern Recognition and Image Analysis (IPRIA)*, pages 160–165.

[19] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210. IEEE.

[20] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. 2019. Specaugment: A simple data augmentation method for automatic speech recognition. *Proc. Interspeech 2019*, pages 2613–2617.

[21] C. Raphael. 2001. Coarse-to-fine dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(12):1379–1390.

[22] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. 2018. A survey on deep transfer learning. In *International Conference on Artificial Neural Networks*, pages 270–279. Springer.

[23] Sibo Tong, Philip N. Garner, and Hervé Bourlard. 2018. Cross-lingual adaptation of a CTC-based multilingual acoustic model. *Speech Communication*, 104:39 – 46.

[24] Dong Wang and Thomas Fang Zheng. 2015. Transfer learning for speech and language processing. In *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pages 1225–1237. IEEE.

[25] Zhirui Zhang, Shujie Liu, Mu Li, Ming Zhou, and Enhong Chen. 2018. Coarse-to-fine learning for neural machine translation. In *Natural Language Processing and Chinese Computing*, pages 316–328, Cham. Springer International Publishing.

[26] Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.