LTL-UDE at Low-Resource Speech-to-Text Shared Task: Investigating Mozilla DeepSpeech in a low-resource setting

Aashish Agarwal and **Torsten Zesch**
Language Technology Lab
University of Duisburg-Essen
Duisburg, Germany

Abstract

We describe our system participating in the SwissText/KONVENS shared task on low-resource speech-to-text (Plüss et al., 2020). We train an end-to-end neural model based on Mozilla DeepSpeech. We examine various methods to improve over the baseline results: transfer learning from standard German and English, data augmentation, and post-processing. Our best system achieves a somewhat disappointing WER of 58.9% on the held-out test set, indicating that it is currently challenging to obtain good results with this approach in a low-resource setting.

1 Introduction

Recently, end-to-end models like DeepSpeech[1] have been introduced as an alternative to traditional HMM-DNN based models like Kaldi (Povey et al., 2011). However, they are relatively data hungry, i.e. they require large amounts of annotated data to work well. For example, the original DeepSpeech implementation from Baidu (Hannun et al., 2014) was trained on 7,380 hours of data, DeepSpeech2 (Amodei et al., 2015) was trained on 11,940 hours of data and DeepSpeech3 (Battenberg et al., 2017) was trained on about 10,000 hours of data. Such large datasets are usually only available for languages like English or Mandarin, but even for major languages like German much less data is available and consequently DeepSpeech models do not perform well (Agarwal and Zesch, 2019).

In this paper, we examine how well DeepSpeech performs in a truly low-resource setting like Swiss German, where less than 100 hours of annotated data are available. Previous speech recognition systems for Swiss German (Garner et al., 2014; Stadtschnitzer and Schmidt, 2018) are based on Kaldi.

Language	Dataset	Size [h]
Swiss German	SwissText Shared Task	70
	ArchiMob	57
German	Voxforge	57
	TUDA-De	184
	M-AILABS	233
	MCV_v4	454
English	LibriSpeech	1,000
	MCV	1,488

Table 1: Dataset overview

2 Model Training

We used DeepSpeech version 0.6.0 for all experiments.[2]

2.1 Datasets

To train the Swiss German DeepSpeech model, we utilized the following publicly available datasets as showed in Table 1.

For **Swiss German** we used the official data provided by the shared task (Plüss et al., 2020). The corpus contains 70 hours of spoken Swiss German (predominantly in the Bernese dialect) and some Standard German speech from the parliament of the canton of Bern[3]. We additionally use the ArchiMob (Samardžić et al., 2016) corpus, which represents German linguistic varieties spoken within the territory of Switzerland and contains long samples of transcribed text in Swiss German. The corpus contains 57 hours and is

[1] https://github.com/mozilla/DeepSpeech

[2] https://github.com/mozilla/DeepSpeech/releases/tag/v0.6.0

[3] https://swisstext-and-konvens-2020.org/low-resource-speech-to-text/

available under creative commons licence 4.0.[4]

As the amount of data is probably not sufficient to train a good model, we will experiment with transfer learning from standard **German**. Publicly available datasets include Voxforge[5], TUDA-De (Milde and Köhn, 2018), M-AILabs[6], and Mozilla Common Voice (Ardila et al., 2019). Together those datasets add almost 1,000 hours of additional training data (although in the wrong German dialect). The datasets also do not contain political speeches and thus are a less than ideal starting point for transfer learning.

As there has been previous work on transfer learning models starting with a different language (Kunze et al., 2017; Bansal et al., 2018), we also consider **English** corpora: LibriSpeech (Panayotov et al., 2015) and Mozilla Common Voice.[7] These are among the largest and widely used open-source corpora. LibriSpeech consists of 16kHz read English speech derived from audiobooks from the LibriVox project and has been carefully segmented and aligned.[8] On the other hand, the Mozilla Common Voice project employs crowdsourcing to collect data on its portal.

2.2 Server & Runtime

We trained and tested our models on a compute server having 56 Intel(R) Xeon(R) Gold 5120 CPUs @ 2.20GHz, 3 Nvidia Quadro RTX 6000 with 24GB of RAM each. Typical training time with augmentation for the SwissText dataset is 1.5 hours, for German 12 hours, and for English 30 hours. Without augmentation, the training time was approximately 10% less than with augmentation.

2.3 Preprocessing

We cleaned the data by using only the allowed set of characters listed by the shared task. We converted all transcriptions to lower case and further ensured that all audio clips are in *wav* format. The resulting samples were split into training (70%), validation (15%), and test data (15%). The preprocessing scripts can be referenced at GitHub [9]

[4] https://www.spur.uzh.ch/en/departments/research/textgroup/ArchiMob.html

[5] http://www.voxforge.org/home/forums/other-languages/german/open-speech-data-corpus-for-german

[6] https://www.caito.de/2019/01/the-m-ailabs-speech-dataset/

[7] https://voice.mozilla.org/en

[8] http://www.openslr.org/12/

[9] https://github.com/AASHISHAG/deepspeech-swiss-german

Hyperparameter	Value
Batch Size	24
Dropout	0.25
Learning Rate	0.0001
English	
α	0.75
β	1.85
German	
α	0.40
β	1.10

Table 2: Hyperparameters used in the experiments

2.4 Hyperparameters

For the acoustic model, we use the best hyperparameters as reported by (Agarwal and Zesch, 2019) and listed in Table 2.

We use a probabilistic 3-gram language model based on KenLM (Heafield, 2011) and trained on the German-English part of Europarl[10] as well as the corpus used to train the TUDA-De language model (Radeck-Arneth et al., 2015). For German, we searched for a good set of values and got the best results with the ones mentioned in Table 2. For English we referred the values of α and β from DeepSpeech release page[11]

3 Experiments

As the baseline model, we train DeepSpeech with the setup described above and using only the Swiss German data provided by the shared task. The model achieved a WER of 71.5%. As expected, DeepSpeech is not able to simply train a suitable model based on this amount of training data.

We try to improve over those results using data augmentation and transfer learning as discussed in the remainder of this section.

3.1 Data Augmentation

Augmentation is a useful technique for better generalization of machine learning models. Inspired by Park et al. (2019), Mozilla DeepSpeech has implemented several augmentation techniques like *frequency masking*, *time masking*, *speed scaling*, and *pitch scaling*. We used all the augmentation approaches with default hyperparameters, which can be referenced here.[12] Augmentation actually

[10] https://www.statmt.org/europarl/

[11] https://github.com/mozilla/DeepSpeech/releases/tag/v0.6.0

[12] https://deepspeech.readthedocs.io/en/v0.7.0/TRAINING.html#training-with-augmentation

Train	Test	WER w/o augmentation	WER w/ augmentation
SwissText	*SwissText*	*71.5*	*74.3*
Swiss → Swiss	SwissText	70.7	69.0
German → Swiss	SwissText	63.5	63.1
English → Swiss	SwissText	64.1	64.4
English → German → Swiss	SwissText	61.0	61.5

Table 3: Transfer learning results (on public data)

increases model error from 71.5% to 74.3%. However, we further test the impact of augmentation in our transfer learning results discussed below.

3.2 Transfer Learning

As we have discussed above, end-to-end training of automated speech recognition systems requires massive data. As we only have 70 hours of training data available from the shared task, we experiment with transferring the model from different starting points. Table 3 gives an overview of the results. Transferring from about 2,500 hours of English data gives about the same results are starting from about 1,000 hours of German data even if standard German is closer to Swiss German than English. However, the best results are achieved when starting with English, transferring to German and then transferring to Swiss. Data augmentation in this case improves results a bit for a final WER of 61.5%.

4 Error Analysis

When analyzing the errors made by DeepSpeech, one issue stands out: truncated output. Quite a lot of output texts are much shorter than the source transcript. Table 4 shows some examples. The performance of the model will be seriously impacted by not producing long enough output sentences. It might be informative to only look at output text that is about the same length as the original transcript. Figure 1 displays the distribution of samples with a certain ratio of sample length to source sample length in characters. The figure shows that almost all DeepSpeech outputs are shorter than the original. If we only look at the samples that are about the same size as expected (with a ratio higher than 0.75, which is still about half of all samples), we find that WER improves from 61.5% to 47.7%. This means that when the model outputs a string that is approximately of the correct length, it is actually much better than the

WER	Example
	`src:` ich habe diese nicht gefunden
1.00	`def:` es handelt
0.80	`opt:` der handel nicht
	`src:` lohnobergrenze für staatsbetriebe
1.00	`def:` der gefürsteten
0.67	`opt:` bergen für staatsbetrieben
	`src:` er ist ein erfahrener grossrat
1.00	`def:` die songs
0.53	`opt:` die so ein grossrat
	`src:` ich überlege mir jetzt folgendes
1.00	`def:` es bereitet
0.40	`opt:` ich überlege ich jetzt wenn
	`src:` sie sehen die gleichstellung ist leider noch gar nicht erreicht
0.90	`def:` die stellung scheidungen
0.60	`opt:` die stellung schleid und noch gar nicht ein

Table 4: Examples of truncated output with default (def) hyperparameters that improve when optimized (opt)

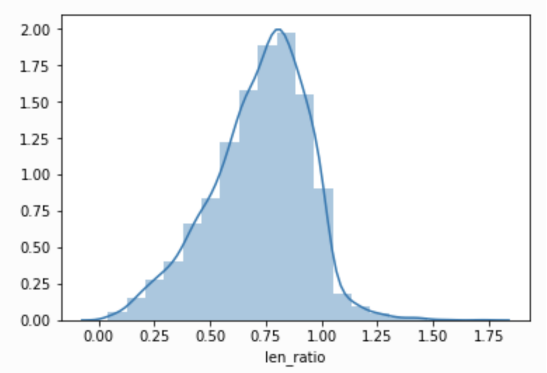

Figure 1: Distribution of output sample length to original sample length (in characters)

results in Table 3 indicate.

The length of the output is partly controlled by the model's hyperparameters. We want to find a sequence c that maximizes the combined objective function:

$$Q(c) = \log(P(c|x)) + \alpha \log(P_{lm}(c)) + \beta \ wordcount(c)$$

where α and β controls the trade-off between the acoustic model, the language model constraint, and the length of the sentence. The term P_{lm} indicates the probability of the sequence c according to the language model. The weight α constrains the relative contributions of the CTC network and the language model and the weight β determines the count of words in the recognized transcription (Hannun et al., 2014; Amodei et al., 2015).

By changing the relative weight of acoustic model and language model by optimizing α and β, we can improve the model a bit as shown in the optimized model examples in Table 4. However, we were not able to eliminate the problem altogether.

Train	Test	WER w/o augmentation	WER w/ augmentation
SwissText	*SwissText*	*70.2*	*69.6*
Swiss → Swiss	SwissText	67.9	68.6
German → Swiss	SwissText	59.4	59.5
English → Swiss	SwissText	60.1	59.1
English → German → Swiss	SwissText	56.6	57.1

Table 5: Transfer learning results (on public data) - with optimized hyperparameters

Consequently, WER only improves from 61.5. to 57.1 (with augmentation). As the model with the optimized hyperparameters and without augmentation is still a bit better, we submitted that one in the shared task. It achieved a WER of 58.9% on the held-out test set.

5 Summary

The baseline system trained only on the Swiss-German data yields a quite high word error rate of 71.5. Data augmentation strategies implemented in DeepSpeech did not result in consistent improvements. Transfer learning has a much higher impact reducing the word error rate by over 10 percent points when transferring an English model to German and finally transferring to Swiss German. The best model yields a WER of 56.6% on our test set (58.9% in the public ranking based on the hidden test set of the shared task). When analyzing the results, the model seems to suffer from truncated output which we can somewhat improve by hyperparameter tuning. Overall, the results show that training an end-to-end neural speech recognition system with DeepSpeech in a low-resource setting remains challenging.

References

Aashish Agarwal and Torsten Zesch. 2019. German end-to-end speech recognition based on deepspeech. In *Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019)*, pages 111–119, Erlangen, Germany. GSCL.

Dario Amodei, Rishita Anubhai, Eric Battenberg, and Carl Case. 2015. Deep speech 2: End-to-end speech recognition in english and mandarin. *CoRR*, abs/1512.02595.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M. Tyers, and Gregor Weber. 2019. Common voice: A massively-multilingual speech corpus.

Sameer Bansal, Herman Kamper, Karen Livescu, Adam Lopez, and Sharon Goldwater. 2018. Pre-training on high-resource speech recognition improves low-resource speech-to-text translation. *CoRR*, abs/1809.01431.

Eric Battenberg, Jitong Chen, Rewon Child, Adam Coates, Yashesh Gaur, Yi Li, Hairong Liu, Sanjeev Satheesh, David Seetapun, Anuroop Sriram, and Zhenyao Zhu. 2017. Exploring neural transducers for end-to-end speech recognition. *CoRR*, abs/1707.07413.

Philip N. Garner, David Imseng, and Thomas Meyer. 2014. Automatic speech recognition and translation of a swiss german dialect: Walliserdeutsch.

Awni Y. Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, and Andrew Y. Ng. 2014. Deep speech: Scaling up end-to-end speech recognition. *CoRR*, abs/1412.5567.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland.

Julius Kunze, Louis Kirsch, Ilia Kurenkov, Andreas Krug, Jens Johannsmeier, and Sebastian Stober. 2017. Transfer learning for speech recognition on a budget. *CoRR*, abs/1706.00290.

Benjamin Milde and Arne Köhn. 2018. Open source automatic speech recognition for german. *CoRR*, abs/1807.10311.

V. Panayotov, G. Chen, D. Povey, and S. 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.

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. *Interspeech 2019*.

Michel Plüss, Lukas Neukom, and Manfred Vogel. 2020. Germeval 2020 task 4: Low-resource speech-to-text. In preparation.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. 2011. The kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*.

Stephan Radeck-Arneth, Benjamin Milde, Arvid Lange, Evandro Gouvêa, Stefan Radomski, Max Mühlhäuser, and Chris Biemann. 2015. Open source german distant speech recognition: Corpus

and acoustic model. In *Text, Speech, and Dialogue*, pages 480–488, Cham.

Tanja Samardžić, Yves Scherrer, and Elvira Glaser. 2016. ArchiMob - a corpus of spoken swiss German. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4061–4066, Portorož, Slovenia. European Language Resources Association (ELRA).

Michael Stadtschnitzer and Christoph Schmidt. 2018. Data-driven pronunciation modeling of swiss German dialectal speech for automatic speech recognition. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).