Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019)

German End-to-end Speech Recognition based on DeepSpeech

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

Abstract

While automatic speech recognition is an important task, freely available models are rare, especially for languages other than English. In this paper, we describe the process of training German models based on the Mozilla DeepSpeech architecture using publicly available data. We compare the resulting models with other available speech recognition services for German and find that we obtain comparable results. Acceptable performance under noisy conditions would, however, still require much more training data. We release our trained German models and also the training configurations.

1 Introduction

Automatic speech recognition (ASR) is the task of translating a spoken utterance into a textual transcript. It is a key component of voice assistants like Google Home (Li et al., 2017), in spoken language translation devices (Krstovski et al., 2008), or for automatic transcription of audio and video files (Liao et al., 2013). For any language beyond English, readily available pre-trained models are still rare. For German, we are only aware of the model by Milde and Köhn (2018) for the Kaldi framework (Povey et al., 2011). For the recently introduced Mozilla DeepSpeech framework, a German model is still missing. This is a serious obstacle to applied research on German speech data, as available web-services by Google, Amazon, or Microsoft are problematic due to data privacy reasons. We thus use publicly available speech data to train a German DeepSpeech model. We release our trained German model and also publish the code and configurations enabling researchers to (i) directly use the model in applications, (ii) reproduce state-of-the-art results, and (iii) train new models based on other source corpora.

2 Speech Recognition Systems

Due to the underlying complexity of recognizing spoken language and the wish of the service provider to keep the model private, many systems are offered as *web services*. This includes commercial services like Google Cloud Speech-to-Text (He et al., 2018), Amazon Alexa Voice Services[1], IBM Watson Speech to Text (Saon et al., 2017) or Speechmatics[2] as well as academic services like BAS.[3] While web services are convenient, there are many situations where they cannot be used:

- sending data to a web service might violate data privacy protection laws
- as the data throughput of a web service is limited; it might rule out batch processing of large amounts of speech data
- the user cannot control (or change) the functionality of a remotely deployed web service
- research results based on web service calls are not easily replicable, as services might change without notice or become unavailable altogether.

For this work, we therefore consider only frameworks that can be used locally and without restrictions. One such framework is **Kaldi** (Povey et al., 2011) which was found to be the best performing open-source ASR system in a previous study (Gaida et al., 2014). It is open-source toolkit written in C++ that supports conventional models (e.g. Gaussian Mixture Models) as well as deep neural networks. Recently, end-to-end neural systems like **wav2letter++** (Pratap et al., 2018) provided by Facebook, or **DeepSpeech**[4] provided by Mozilla have been introduced. To our knowledge, there is

[1] https://developer.amazon.com/alexa/science

[2] https://www.speechmatics.com

[3] https://clarin.phonetik.uni-muenchen.de/BASWebServices/interface/ASR

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

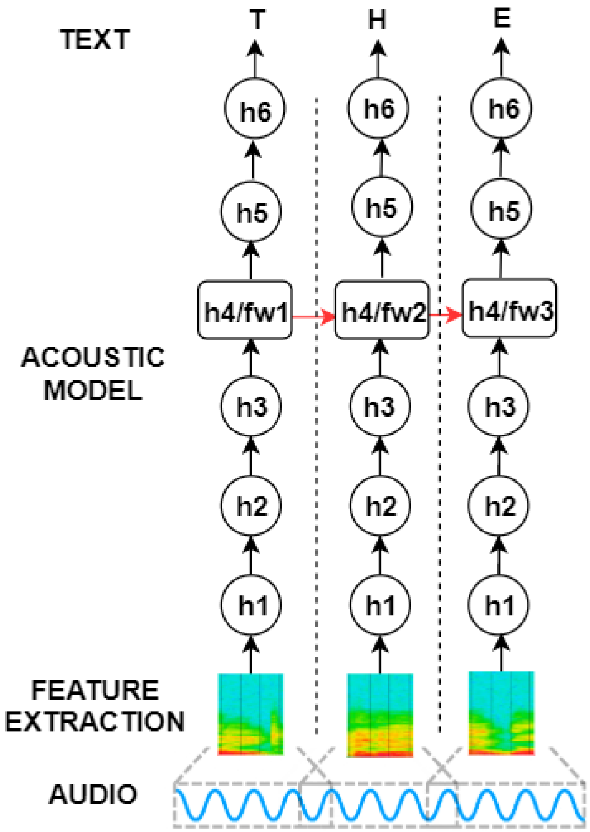


Figure 1: DeepSpeech architecture (adapted from *Mozilla Blog*[5])

only one German model for any of these frameworks that is publicly available, which is the one by Milde and Köhn (2018) for Kaldi. Other German models, e.g. a Kaldi model from Fraunhofer IAIS (Stadtschnitzer et al., 2014), rely on in-house datasets and are not publicly available.

In this work, we focus on Mozilla's DeepSpeech framework, as it is an end-to-end neural system that can be quite easily trained, unlike Kaldi, which requires more domain knowledge or wav2letter++, which is not yet widely tested by the community.

Mozilla DeepSpeech DeepSpeech (v0.1.0) was based on a TensorFlow (Abadi et al., 2016) implementation of Baidu's end-to-end ASR architecture (Hannun et al., 2014). As it is under active development, the current architecture deviates from the original version quite a bit. In Figure 1, we give an overview of the architecture of version v0.5.0, which we also used for our experiments in this paper.[6]

DeepSpeech is a character-level, deep recurrent neural network (RNN), which can be trained end-to-end using supervised learning.[7] It extracts Mel-Frequency Cepstral Coefficients (Imai, 1983) as features and directly outputs the transcription, without the need for forced alignment on the input or any external source of knowledge like a Grapheme to Phoneme (G2P) converter. Overall, the network has six layers: the speech features are fed into three fully connected layers (dense), followed by a unidirectional RNN layer, then a fully connected layer (dense) and finally an output layer as shown in Figure 1. The RNN layer uses LSTM cells, and the hidden fully connected layers use a ReLU activation function. The network outputs a matrix of character probabilities, i.e. for each time step the system gives a probability for each character in the alphabet, which represents the likelihood of that character corresponding to the audio. Further, the Connectionist Temporal Classification (CTC) loss function (Graves et al., 2006) is used to maximize the probability of the correct transcription.

DeepSpeech comes with a pre-trained English model, but while Mozilla is collecting speech samples[8] and is releasing training datasets in several languages (see paragraph on Mozilla Common Voice in Section 3), no official models other than English are provided. Users have reported on training models for French[9] and Russian (Iakushkin et al., 2018), but the resulting models do not seem to be available.

3 Model Training

In this section, we describe in detail our setup for training the German model in order to ease subsequent attempts to train DeepSpeech models.

3.1 Datasets

To train the German Deep Speech model, we utilize the following publicly available datasets:

The **Voxforge**[10] corpus, which is about 35 hours of German speech clips. Nearly 180 speakers have read aloud sentences from German Wikipedia, protocols from the European Parliament, and some individual commands. The clips vary in length, ranging from 5 to 7 seconds.

The **Tuda-De** (Milde and Köhn, 2018) corpus, is similar to Voxforge. It uses the same sources

[5] https://hacks.mozilla.org/2018/09/speech-recognition-deepspeech

[6] https://github.com/mozilla/DeepSpeech/releases/tag/v0.5.0

[7] https://hacks.mozilla.org/2017/11/a-journey-to-10-word-error-rate/

[8] https://voice.mozilla.org/

[9] http://bit.ly/discourse-mozilla-org

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

Dataset	Size	Median Length	# Speakers	Condition	Type
Voxforge	35h	4.5s	180	noisy	read
Tuda-De	127h	7.4s	147	clean	read
Mozilla Common Voice	140h	3.7s	>1,000	noisy	read

Table 1: Overview of German datasets

(Wikipedia, parliament speeches, commands), but the recordings are under more controlled conditions. The final data was also curated "to reduce speaking errors and artefacts". Each recording was made with 4 different microphones at the same time. This means that while the overall size of the dataset is larger than Voxforge and a model based on this dataset is supposed to be more robust, the actual amount of unique speech hours in both datasets are about the same.

The **Mozilla Common Voice** project[11] aims to make speech recognition open to everyone. The multilingual dataset currently covers 18 languages - including English, French, German, and Mandarin. The German corpus contains clips with lengths varying from 3 to 5 seconds. However, the corpus is recorded outside controlled conditions as per the comfort of the speaker. The utterances have background noise, and users have varied accents. Therefore we expect this dataset to be relatively challenging. Speakers in this dataset are relatively young, and the male/female ratio is about 5:1, which might result in a severe bias when trying to transfer the model.[12] The version used in our experiments has 140 hours of recordings, but as Mozilla aims at adding more recordings, there might already be a larger dataset available.

3.2 Preprocessing

DeepSpeech expects audio and transcription data to be prepared in a specific format so that they can be read directly by the input pipeline (see Figure 2 for an example). We cleaned the transcriptions by removing commas as well as punctuation and converting all transcriptions to lower case. We further ensured all audio clips are in *.wav* format. The pruned results were split into training (70%), validation (15%), and test data (15%).

For more details on data preprocessing parameters, we refer the reader to the code release.[13]

Hyperparameter	Value
Batch Size	24
Dropout	0.25
Learning Rate	0.0001

Table 2: Hyperparameters used in the experiments

3.3 Hyperparameter Setup

We searched for a good set of hyperparameters as shown in Figure 3. In the first iteration, we select learning rate and train batch-size and plot the graph showing the relationship of dropout and word-error rate, to determine the dropout with the lowest WER. We then used the best dropout (0.25) from the above iteration and kept the train batch size, to identify the best learning rate. Finally, we took the best dropout (0.25) and learning rate (0.0001) to determine the effect on batch size which shows that our initial choice of 24 was reasonable, even if somewhat better results seem possible using smaller batches.

Since Deep Speech employs early stopping, which stops the training of a neural network early before it overfits the training data, we did not experiment much with the number of epochs. The remaining hyperparameters were set to be the same as those pre-configured in Mozilla Deepspeech. The best results are obtained with the hyper-parameters mentioned in Table 2. We train the network using the Adam optimizer (Kingma and Ba, 2014).

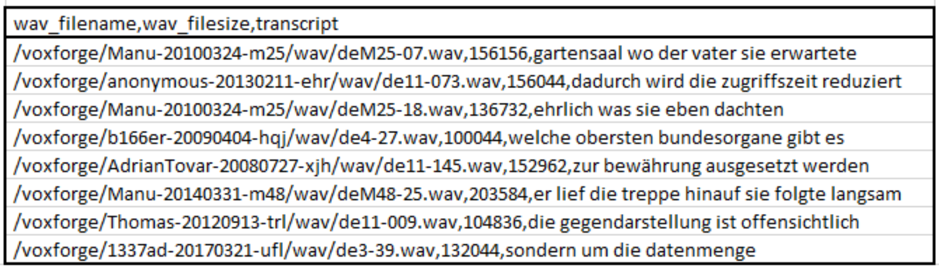
Language Model We apply a probabilistic language model using KenLM toolkit (Heafield, 2011) to train a 3-gram model on the pre-processed corpus provided by Radeck-Arneth et al. (2015). It consists of eight million filtered sentences comprising 63.0% Wikipedia, 22.0% Europarl, and 14.6% crawled sentences. MaryTTS[14] has been used to canonicalize the corpus, i.e. normalized to a form that is close to how a reader would speak the sentence, especially changing numbers, abbreviations, and dates. Additionally, punctuations were discarded, as it is usually also not pronounced. We

[11] https://voice.mozilla.org/de/datasets

[12] Speaker Information is based on the self-reported statistics provided on the project homepage for each dataset.

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

[14] http://mary.dfki.de/



```
wav_filename,wav_filesize,transcript
/voxforge/Manu-20100324-m25/wav/deM25-07.wav,156156,gartensaal wo der vater sie erwartete
/voxforge/anonymous-20130211-ehr/wav/de11-073.wav,156044,dadurch wird die zugriffszeit reduziert
/voxforge/Manu-20100324-m25/wav/deM25-18.wav,136732,ehrlich was sie eben dachten
/voxforge/b166er-20090404-hqj/wav/de4-27.wav,100044,welche obersten bundesorgane gibt es
/voxforge/AdrianTovar-20080727-xjh/wav/de11-145.wav,152962,zur bewährung ausgesetzt werden
/voxforge/Manu-20140331-m48/wav/deM48-25.wav,203584,er lief die treppe hinauf sie folgte langsam
/voxforge/Thomas-20120913-trl/wav/de11-009.wav,104836,die gegendarstellung ist offensichtlich
/voxforge/1337ad-20170321-ufl/wav/de3-39.wav,132044,sondern um die datenmenge
```

Figure 2: Screenshot of the input file format

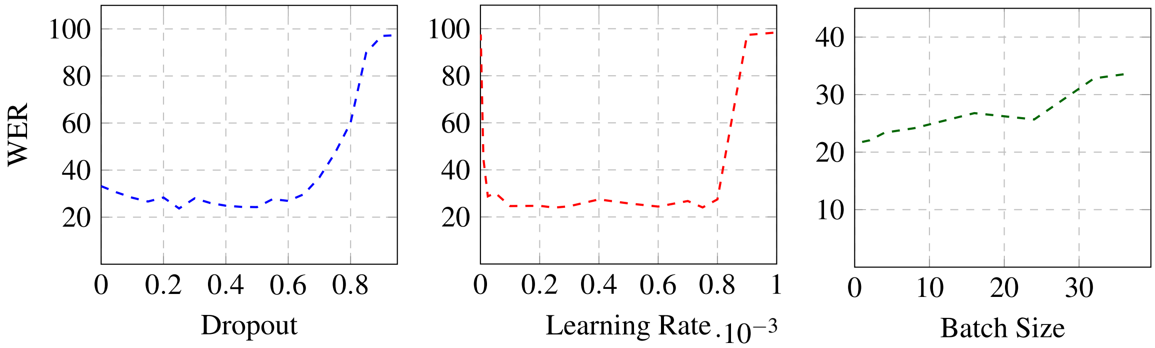


Figure 3: Hyperparameter search space

Dataset	WER
Mozilla	79.7
Voxforge	72.1
Tuda-De	26.8
Tuda-De + Mozilla	57.3
Tuda-De + Voxforge	15.1
Tuda-De + Voxforge + Mozilla	21.5

Table 3: German DeepSpeech results

used the unpruned Language Model that has a rather large vocabulary size of over 2 million types, but we expect pruning would only affect runtime, not recognition quality.

3.4 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 on a single dataset under this setup was in the range of 1 hour.

4 Results & Discussion

Table 3 shows the word error rates (WER) obtained when training and testing DeepSpeech on the available German datasets and their combinations. The best configuration in Milde and Köhn (2018) using only the Tuda-De corpus yields a WER of 28.96%. Our model only trained on Tuda-De yields a comparable WER of 26.8%.

Results for the other datasets are much lower, but apparently combining several datasets improves the results. While the combination of Tuda and Mozilla yields a WER of 57.3%, the combination of Tuda, Voxforge, and Mozilla gives a WER of 21.5%. Combining the very similar Tuda-De and Voxforge yields a WER of 15.1%, which is a remarkable improvement over using only a single dataset. Note that this is the black-box performance, as we used DeepSpeech as is and only slightly tuned hyperparameters. See Section 6 for ideas on how to improve over these results.

To put our results into perspective, in Table 4, we present results in other languages for training different versions of the DeepSpeech architecture. Our best results are in the same range as for the other languages, but cross-dataset comparisons are hard to interpret. However, it is safe to say that training a DeepSpeech model can result in acceptable in-domain word error rates with considerably less training data than previously considered.

4.1 Influence of Training Size

Figure 4 depicts the relation between the amount of training data and its impact on the word-error-rate. To plot the learning curve, we split the training data into 10 subsets containing each 10% of the

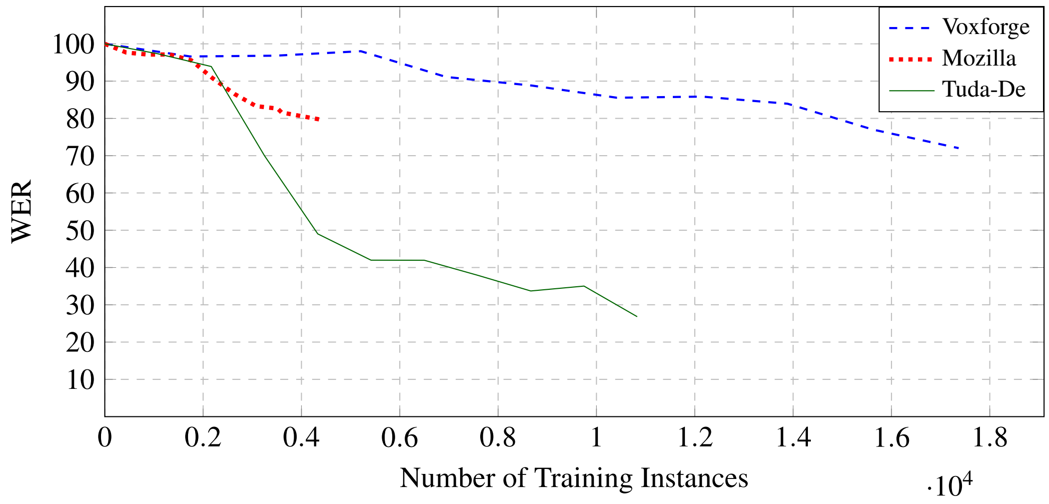


Figure 4: Learning curves for single datasets

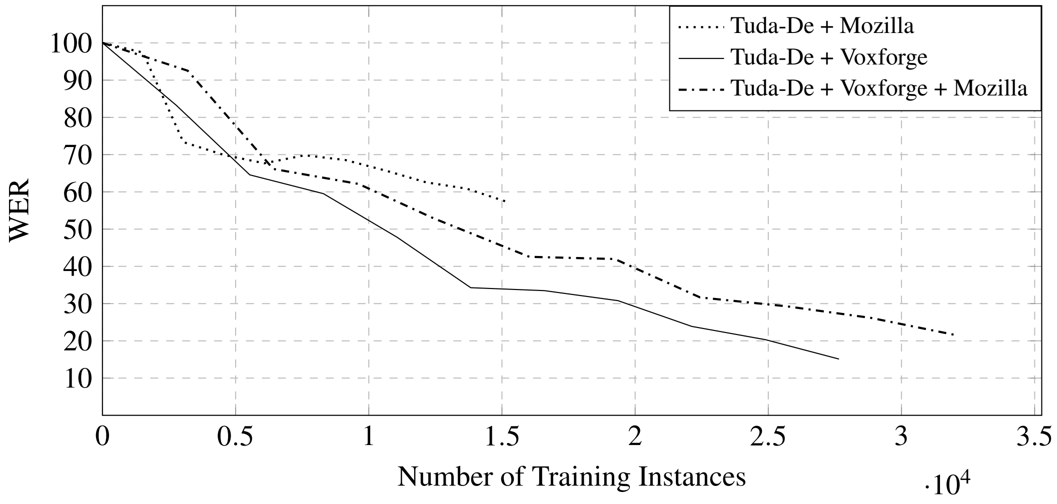


Figure 5: Learning curves when combining datasets

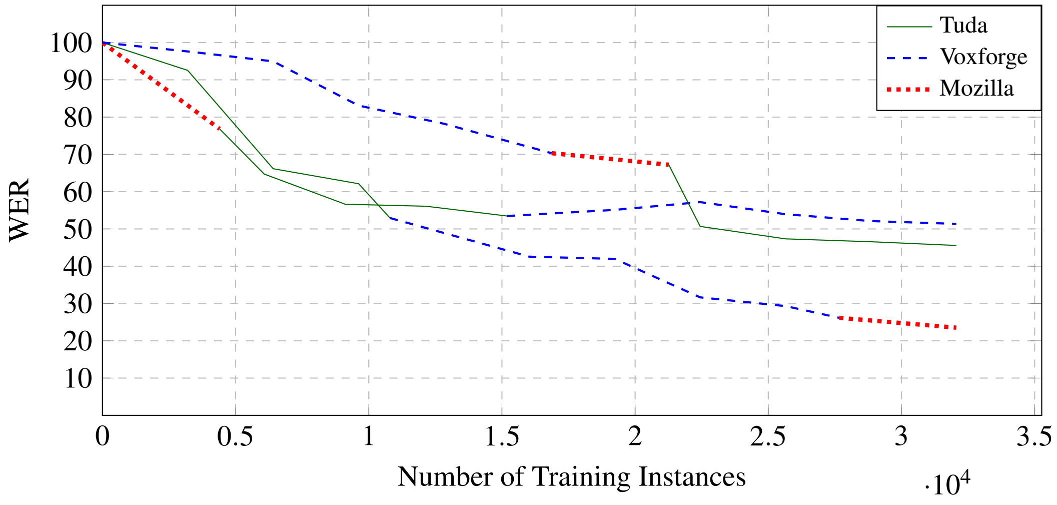


Figure 6: Order effects when combining datasets

Language	DeepSpeech version	Training Set	Size	Test Set	WER
English	Baidu (Hannun et al., 2014)	Switchboard Fisher WSJ Baidu	7,380h	Hub5 (LDC2002S23)	16.0
English	Mozilla v0.3.0	Switchboard Fisher LibriSpeech	3,260h	LibriSpeech (clean test)	11.0
English	Mozilla v0.5.0	Switchboard Fisher LibriSpeech	3,260h	LibriSpeech (clean test)	8.2
Russian	Mozilla v? (Iakushkin et al., 2018)	Yt-vad-1k Yt-vad-650-clean	1,650h	Voxforge (Russian)	18.0
German	Mozilla v0.5.0 (our)	Tuda-De + Voxforge	162h	Tuda-De + Voxforge (test)	15.1

Table 4: Comparison with previous results in other languages

training data. Then the model is trained on one subset and WER is calculated on a separate test dataset. Next, we introduce the new subset with more data, re-train the model, and compute its effect on the error rate. The model is trained on each subset for a maximum of 10 epochs and sometimes less when the model starts to overfit the training data, and early stopping is triggered. We observe that the rather noisy datasets Voxforge and Mozilla converge rather slowly, while the clean Tuda-De reaches much better results. This might also be a result of the different microphones that add increased robustness (not unlike other data augmentation strategies).

Figure 5 present the same learning curves when combining datasets showing that we can reach even better WER in this setting. Mixing the datasets seems to force the model to converge more quickly. However, combining the similar dataset Tuda-De and Voxforge yields a bit better performance than combining all three datasets.

We also tested against a mix of all datasets in combination, but add training data one dataset at a time. Thus, the order in which datasets are introduced into the training process might influence performance. Figure 6 shows the results for different order in which the datasets are introduced into the training process. Adding the noisy Mozilla dataset too early in the process seems to slow down convergence, while it adds a little bit of improved performance when added in the end.

4.2 Cross-dataset Performance

So far, we used training and testing data either from the same dataset or a mix of the available datasets, while of course keeping train and test data separate. To get a more realistic estimate of performance when used in a general setting, we assess cross-dataset performance, i.e. we train and develop on one or two datasets and test on a third one.

Train	Test	WER
Voxforge	Voxforge	*72.1*
Tuda-De		96.8
Mozilla		73.1
Tuda-De, Mozilla		66.2
Tuda-De	Tuda-De	*26.8*
Voxforge		98.5
Mozilla		84.9
Voxforge, Mozilla		83.8
Mozilla	Mozilla	*79.7*
Tuda-De		94.8
Voxforge		87.1
Tuda-De, Voxforge		80.5

Table 5: Results across datasets

Table 5 shows the resulting word error rates. Apparently, the cross-domain results are much worse than in the in-domain setting in Table 3. For example, training on Mozilla or Voxforge and Mozilla and testing on Tuda-De yield unacceptable word error rates of 84.9 and 83.8 compared to 26.8 when training on Tuda-De. Interestingly, in this case, as we have seen already above, adding Voxforge in the mix does not help much, even if it is similar to Tuda-De. We see a similar picture for the other test datasets, transferring from a single dataset does not work at all, as in the training process the model is never forced to generalize beyond its properties.

However, training on the Tuda-De and Mozilla combination yields WER of 66.2 on Voxforge,

Model	WER	Example
original	-	*der bandbreitenverbrauch wird erheblich verringert*
Tuda-De	60	diese zeiten tonwoche erheblich verringert
Voxforge	80	zeiten epoche erheblich in
Tuda-De + Mozilla	160	es sind endete suche den ist es in
Tuda-De + Voxforge	60	der pen zeiten verprach wird erheblich verringert
Tuda-De + Voxforge + Mozilla	40	der bandbreiten verbrauch wird erheblich verringert
original	-	*ferner gibt es möglicherweise eine gewisse anonymität und sicherheit*
Tuda-De	78	weites mögliche welche in glichen unität und sicherheit
Voxforge	100	zitierweise sich entsichert
Tuda-De + Mozilla	100	hunde titisee gelten die die mitte zum
Tuda-De + Voxforge	44	den gibt es möglicherweise eine gewisse mietsicherheit
Tuda-De + Voxforge + Mozilla	11	er gibt es möglicherweise eine gewisse anonymität und sicherheit
original	-	*die einwilligung des schuldners war nicht erforderlich*
Tuda-De	100	ideen
Voxforge	86	die angebliche natacha vollich
Tuda-De + Mozilla	57	die einwilligung des schutzmacht erfordern
Tuda-De + Voxforge	86	die ein eigenes schuldnersicht erfordern
Tuda-De + Voxforge + Mozilla	43	die einigung des schuldner zwar nicht erforderlich
original	-	*die geschwindigkeit für die kunden kann erhöht werden*
Tuda-De	75	die geschwindigkeit und unterteilten
Voxforge	100	schinkelpreise
Tuda-De + Mozilla	88	wie die schmiede den trennendes
Tuda-De + Voxforge	38	die geschwindigkeit für die kunden kenterte
Tuda-De + Voxforge + Mozilla	0	die geschwindigkeit für die kunden kann erhöht werden
original	-	*mehrere arbeitgeberverbände sind zu einem dachverband zusammengeschlossen*
Tuda-De	114	der see aufweitungen des in einem tatorten samen erschossen
Voxforge	100	es recognitionszeichen
Tuda-De + Mozilla	100	in den sitzungen des entstandenen schaden
Tuda-De + Voxforge	29	mehrere arbeitgeberverbände sind zu einem tachodaten geschlossen
Tuda-De + Voxforge + Mozilla	14	der arbeitgeberverbände sind zu einem dachverband zusammengeschlossen

Table 6: Recognition results on random Voxforge test instances

which is even lower than using the training portion of Voxforge (which yields 72.1). Thus forcing the model to generalize over topics, recording conditions, speakers, etc. seem to be a crucial point.

5 Error Analysis

Table 6 shows the recognition results on randomly selected test instances from the Voxforge dataset. The models trained on only one dataset are surprisingly bad, resulting in rather poetic utterances that sometimes are quite far from the expected source. An example is the Tuda-De model recognizing *tatorten samen erschossen* instead of *dachverband zusammengeschlossen*.

As is to be expected for German, compounds are especially challenging as exemplified by *bandbreitenverbrauch* that is recognized as *bandbreiten verbrauch* or even *pen zeiten verprach*, where *verprach* is probably only in the language model as a common misspelling of *versprach*.

The models often fail in interesting ways, e.g. all models sometimes return very short results like *schinkelpreise* that should actually have low probability. We currently have no explanation for this behaviour and need to explore the issue further.

In cases like *des schuldners war* being recognized as *des schuldner zwar*, the phonetic ambiguity should have been resolved by a better language model.

6 Summary

In this paper, we presented the first results on building a German speech recognition model using Mozilla Deep Speech. Our best performing model reaches an in-domain WER of 15.1%, which is in line with the performance for other languages using the DeepSpeech framework. Our results thus support the idea that Mozilla Deep Speech can be easily transferred to new languages. Learning curve experiments highlight the importance of the amount of training data, but also quite strong order effects when mixing the datasets.

We publish our trained model along with configuration data for all our experiments in order to enable replicating all results. The model can easily be re-trained and optimised on new datasets by referring the code-release.[15] No specific hardware is required to run the trained model, and it works even on a normal desktop computer or laptop.

Future Work Our experiments only scratch the surface of possible approaches, and our analysis recommends several avenues for further exploration.

We mainly treated DeepSpeech as a black-box and only performed a light hyper-parameter search. The model can probably still be fine-tuned by exploring other hyper-parameters. We also did not experiment much with the language model, but used a simple 3-gram model.

Since the amount of publicly available training data is limited, it could be interesting to consider data augmentation strategies.[16] Another approach to improve recognition quality could be to use transfer learning by taking an English model (pre-trained with the larger English datasets) and re-training with the German data (Kunze et al., 2017; Bansal et al., 2018). In the light of recent discussions on the CO2 footprint of training deep learning models (Strubell et al., 2019), using re-training and providing trained models is desirable. Additionally, more research is needed to find neural architectures that perform equally well, but require less compute.

Finally, the training process described here could be easily used to train speech recognition models for other languages, where currently no pre-trained models are available.

Acknowledgments

We want to thank Andrea Horbach for her many helpful comments that significantly improved the paper. We also thank the developers at Mozilla DeepSpeech, who provided insight and expertise that greatly assisted the research.

References

[Abadi et al.2016] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation OSDI 16*, pages 265–283.

[Bansal et al.2018] 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.

[Gaida et al.2014] Christian Gaida, Patrick Lange, Rico Petrick, Patrick Proba, Ahmed Malatawy, and David Suendermann-Oeft. 2014. Comparing open-source speech recognition toolkits.

[Graves et al.2006] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural 'networks. volume 2006, pages 369–376, 01.

[Hannun et al.2014] 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.

[He et al.2018] Yanzhang He, Tara N. Sainath, Rohit Prabhavalkar, Ian McGraw, Raziel Alvarez, Ding Zhao, David Rybach, Anjuli Kannan, Yonghui Wu, Ruoming Pang, Qiao Liang, Deepti Bhatia, Yuan Shangguan, Bo Li, Golan Pundak, Khe Chai Sim, Tom Bagby, Shuo-Yiin Chang, Kanishka Rao, and Alexander Gruenstein. 2018. Streaming end-to-end speech recognition for mobile devices. *CoRR*, abs/1811.06621.

[Heafield2011] 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.

[Iakushkin et al.2018] Oleg Iakushkin, George Fedoseev, Anna S. Shaleva, Alexander Degtyarev, and Olga S. Sedova. 2018. Russian-language speech recognition system based on deepspeech. In *Proceedings of the VIII International Conference on Distributed Computing and Grid-technologies in Science and Education (GRID 2018).*

[Imai1983] Satoshi Imai. 1983. Cepstral analysis synthesis on the mel frequency scale. In *ICASSP'83. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 8, pages 93–96. IEEE.

[Kingma and Ba2014] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv e-prints*, page arXiv:1412.6980, Dec.

[Krstovski et al.2008] Kriste Krstovski, Michael Decerbo, Rohit Prasad, David Stallard, Shirin Saleem, and Premkumar Natarajan. 2008. A wearable headset speech-to-speech translation system. In *Proceedings of the ACL-08: HLT Workshop on Mobile Language Processing*, pages 10–12, Columbus, Ohio, June. Association for Computational Linguistics.

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

[16] https://ai.googleblog.com/2019/04/specaugment-new-data-augmentation.html

[Kunze et al.2017] 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.

[Li et al.2017] Bo Li, Tara Sainath, Arun Narayanan, Joe Caroselli, Michiel Bacchiani, Ananya Misra, Izhak Shafran, Hasim Sak, Golan Pundak, Kean Chin, Khe Chai Sim, Ron J. Weiss, Kevin Wilson, Ehsan Variani, Chanwoo Kim, Olivier Siohan, Mitchel Weintraub, Erik McDermott, Rick Rose, and Matt Shannon. 2017. Acoustic modeling for google home.

[Liao et al.2013] Hank Liao, Erik McDermott, and Andrew W. Senior. 2013. Large scale deep neural network acoustic modeling with semi-supervised training data for youtube video transcription. In *ASRU*, pages 368–373. IEEE.

[Milde and Köhn2018] Benjamin Milde and Arne Köhn. 2018. Open source automatic speech recognition for german. *CoRR*, abs/1807.10311.

[Povey et al.2011] 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*, December.

[Pratap et al.2018] Vineel Pratap, Awni Hannun, Qiantong Xu, Jeff Cai, Jacob Kahn, Gabriel Synnaeve, Vitaliy Liptchinsky, and Ronan Collobert. 2018. wav2letter++: The fastest open-source speech recognition system. *CoRR*, abs/1812.07625.

[Radeck-Arneth et al.2015] 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.

[Saon et al.2017] George Saon, Gakuto Kurata, Tom Sercu, Kartik Audhkhasi, Samuel Thomas, Dimitrios Dimitriadis, Xiaodong Cui, Bhuvana Ramabhadran, Michael Picheny, Lynn-Li Lim, Bergul Roomi, and Phil Hall. 2017. English conversational telephone speech recognition by humans and machines. *CoRR*, abs/1703.02136.

[Stadtschnitzer et al.2014] Michael Stadtschnitzer, Jochen Schwenninger, Daniel Stein, and Joachim Koehler. 2014. Exploiting the large-scale German Broadcast Corpus to boost the Fraunhofer IAIS Speech Recognition System. In *Proceedings of LREC 2014*, pages 3887–3890, Reykjavik, Iceland.

[Strubell et al.2019] Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in nlp. In *Proceedings of ACL*.