Swiss German Speech to Standard German Text
SwissText.org Shared Task 3

Alex Wolf
University of Zurich
`alex.wolf@uzh.ch`

Deborah Noemie Jakobi
University of Zurich
`deborahnoemie.jakobi@uzh.ch`

Abstract

This paper presents a solution to the Swiss-Text Conference (STC) shared task 3, 2021. The shared task challenges participants to implement models for translating Swiss German speech to standard German text. The authors implemented multiple *DeepSpeech* models using the data provided by SwissText.org as well as additional data. The use of the additional data in combination with the official data lead to some promising improvements of the model trained on the official data only. An additional experiment with a sequence to sequence translation model was trained in order to improve our score. We achieved a BLEU score of up to 0.17 on the test set of SwissText. Although the performance of the final model did not score well in comparison with other submissions, there is plenty of room for possible improvements.

1 Introduction

Automatic speech recognition (ASR) means translating a spoken utterance into written text. It can, for example, be used for voice assistants or automatic transcription of audio or video files. While Speech-to-Text (STT) pre-trained models for English are available, German models or even Swiss German models are rare or not existing (**?**). Swiss German has a wide variety of different Swiss German dialects, with a huge difference in words, pronunciation, even to the point of sounding like a different language. Swiss German has relatively little speakers (around 5 million) and there is hardly any standardized spelling. This leads to Standard German being one of the official writing language in Switzerland. As there is no official Swiss German spelling, most speakers using written Swiss German just use their own spelling which resembles mostly a phonetical translation. This leads to huge variance within Swiss German writing (**?**). It makes therefore sense to translate spoken Swiss German

to Standard German in order to correspond to the official language situation. Tackling a standardized translation of different spoken Swiss German dialects into standardized German text requires a vast amount of data and fine tuning. The STC 2021 proposed a shared task to tackle this problem and provided a dataset (SwissText Conference dataset (STCD) to train and fine tune on. This paper shows what kind of experiments, data, and approaches the authors used to tackle this problem. The proposed task is very complex as it includes not only a STT conversion but also translation from Swiss German to Standard German which can be refered to as speech translation (**?**). Additionally, it possibly includes domain shift, as the training data stems only from Swiss parliament speeches while the domain of the test set is unclear. This paper presents a overview on the previous research done within this area, an introduction to the DeepSpeech model used for tackling the task and our experiments and results.

2 Literature review

add more general stuff about STT.

The shared task of a translating spoken Swiss German into standard written German was already presented by the STC in 2020. While the current 2021 task is about reaching the highest BLEU score, last year's submissions where ranked based on the least Word Error Rate (WER). **?** achieved the best WER of 40.29%. The authors used a CNN acoustic model named Jasper. They used additional Standard German data and fine-tuned on the official data set. They used different augmentation techniques and a language model. **?** add text. (**?**) achieved the second best WER. ...
? used an end-to-end model called DeepSpeech and achieved an WER of 58.93%. This is the model we decided to use as well. It is publicly available

on GitHub and can easily be fine-tuned or used for transfer learning.

3 DeepSpeech

Mozilla DeepSpeech is an end-to-end STT model using tensorflow. It was first developed for translating English speech to English text (?). ? have implemented a German STT model using DeepSpeech. It is trained with machine translation techniques. They provide all of their code and pre-trained models including ready-made scripts for transfer-learning and fine-tuning (QUOTE GITHUB https://github.com/AASHISHAG/deepspeech-german). This avoids privacy issues of common web-services that require uploading potentially private data. Additionally, researchers are free to adjust and extend the model according to their requirements. DeepSpeech is a deep recurrent neural network (RNN) on character level. It can be trained using supervised learning.
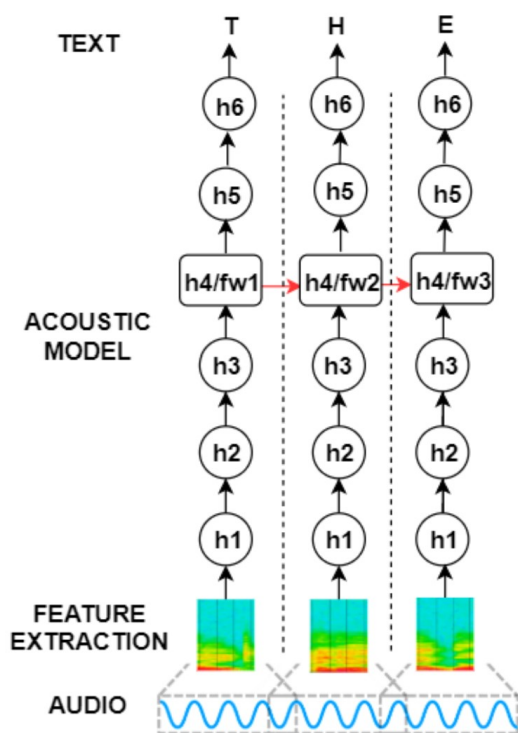

Figure 1: DeepSpeech architecture according to ?

The DeepSpeech architecture, as shown in 1, consists of 6 layers. A more in depth description can be found in ?. As becomes clear, the model has no additional phoneme-to-grapheme model but directly outputs the transcribed characters, respectively their probabilities. The Connectionist Tem-poral Classification (CTC) loss is used to maximize the probability of the output characters. This loss is specifically designed for tasks where the prediction categories have unclear boundaries. In this case characters which can refer to phonemes that span over times frames of various length.

4 Experiments

4.1 Training Data

To train our model we use a subset of the dataset provided by the STC as well as the ArchiMob corpus. The STCD contains 38 GB of labeled, 65 GB of unlabeled spoken swiss german audio data and an additional validation set containing 1.5 GB of data (?). The ArchiMob corpus (Release 2) contains X GB of spoken swiss german data (?). We remove all audio files, from the STCD, where the quality of the translation is rated lower than 0.7 (rating provided by STC). Additionally, we also need to remove files larger than XY GB due to memory limitations.

4.2 Evaluation metrics

We evaluate our model through two types of metrics. The BLEU score (?), which is required by the STC to compare the results. The STC specifically requires a corpus based BLEU score (todo cite stc), which aims at measuring the distance/similarity of the generated text and the provided ground truth.

Additionally, we measure the WER for our models, as this is a common metric for speech recognition systems (?).

4.3 Environment or Setup?

We train the model on a single XYZ GPU (XYG memory) for X epochs, with batch size of X and used a learning rate of XY, ...
We perform minimal tuning of our model's hyper-parameters following the work of ? as this model achieved the best results in previous challenges.

4.4 Models

Following ? we use a DeepSpeech architecture (?) as our main model for speech-to-text translation. In order to get better results we use a a pre-trained DeepSpeech model (?) as the base model for most of our experiments. The first model we trained served as our private baseline.

Baseline model We trained a bare DeepSpeech model, with the default hyperparameters from **?**, on the labelled STCD, which achieved a BLEU score of XY BLEU. The baseline model will be revered to as baseline model or model #0 henceforth.

Pre-trained model #1 In order to improve our baseline model we use a pre-trained german Deep-Speech model (**?**), which is fine-tuned on the STCD dataset for 75 epochs, with a learning rate of XY, and a batch size of XY. The model achieved a BLEU score of XY on the validation set and XY on the test set.

Additional data model #2 To further improve our 1st model we added additional data to our fine-tuned model. We added the ArchiMob data and fine-tuned for an additional 75 epochs, with the same hyperparameters as in model #1 and achieved the following BLEU scores on the validation and trainig set, XYZ, XYZ2, respectively.

ArchiMob data model #3 As the 2nd model did not achieve the expected results we decided to fine-tune our model on the ArchiMob data, as we expected the data distribution to be closer to the actual test set. The model was trained for 75 epochs, with a learning rate of XY, and a batch size of XY. We achieved the following BLEU scores for the validation and test set, XYZ, XYZ2, respectively.

Augmented data model #4 In order to produce a more robust model, that can handle different pronunciations and speeds, we added data augmentation to our 3rd model. We fine-tuned our model with the following data augmentations:

- warp: "Applies a non-linear image warp to the spectogram. This is achieved by randomly shiting a grid of equally distribted warp points along time and frequency axis" (**?**)

- tempo: "Scales spectrogram on time axis and thus changes playback tempo." (**?**)

The STCD dataset is used for the augmentation fine-tuning on the 3rd model. This approach achieved a BLEU score of XY on the validation set and XY on the test set. The model was trained for X epochs with a warp probability of 0.1 and a tempo probably of 0.1. We increased the batch size to X, as the augmentation process per batch took more time than expected. Due to time limitations we were not able to continue training the model any further, but the model was still improving and we expect that this approach could produce even better results.

Text-to-Text model #5

5 Results & Discussion

Interestingly, we noticed that our internal deviations in the BLEU score are not as great as on the official test set.

Model#	Data	Train BLEU	Test BLEU
1	SwissText	0.23	0.0004
2	ArchiMob	0.27	**0.17**
3	ArchiMob	0.24	0.07

Table 1: Font guide.

6 Conclusion