IMS-SPEECH: A SPEECH TO TEXT TOOL

Pavel Denisov, Ngoc Thang Vu

Institute for Natural Language Processing (IMS), University of Stuttgart
{pavel.denisov\thang.vu}@ims.uni-stuttgart.de

Abstract: We present the IMS-Speech, a web based tool for German and English speech transcription aiming to facilitate research in various disciplines which require accesses to lexical information in spoken language materials. This tool is based on modern open source software stack, advanced speech recognition methods and public data resources and is freely available for academic researchers. The utilized models are built to be generic in order to provide transcriptions of competitive accuracy on a diverse set of tasks and conditions.

1 Introduction

There is a considerable amount of spoken language materials in form of audio recordings, which researchers in e.g. humanities and social science could incorporate into their studies. However, to be able to access to their content, one needs to automatically transcribe these recordings. While all needed resources for building of an automatic speech recognition (ASR) system are typically available for academic usage, their utilization requires specialized knowledge and technical experience [1], [2]. Therefore, in order to provide people easy accesses to information in spoken language materials, a speech to text tool with a user interface should be helpful.

This paper presents the IMS-Speech[1], a web based tool for German and English speech transcription aiming to facilitate research in various disciplines. We are willing to provide a speech transcription service with an intuitive web interface accessible with a wide range of computing devices and to people with various backgrounds. The service is based on modern open source software stack, advanced speech recognition methods and public data resources and is freely available for academic researchers. The utilized models are built to be generic in order to provide transcriptions of competitive accuracy on a diverse set of tasks and conditions. In addition to that, they can serve as a strong base for customized task specific applications.

2 System description

In order to produce a meaningful transcription for the most of recordings that might be uploaded by users, two tasks must be performed for every recording sequentially. First, a recording must be split to segments not exceeding some short duration and corresponding to speech intervals. Second, actual ASR must be performed over each speech segment for finding the most probable sequence of words being said in the segment and thus constructing final transcription.

2.1 Speech Segmentation

Speech segmentation is performed with a speech activity detection (SAD) system based on Time-Delay Neural Network (TDNN) [3] with statistics pooling [4] for long-context information. TDNN is trained to estimate probability of 3 classes, *Silence*, *Speech* and *Garbage*, for

[1] `http://www.ims.uni-stuttgart.de/forschung/ressourcen/werkzeuge/IMS-Speech.html`

each frame. Training targets are assigned based on lattices produced by Gaussian Mixture Model (GMM) based acoustic models and predefined lists of phones for each class. GMM is used for forced alignment as well as for unconstrained decoding. Training targets are obtained from both procedures separately and consequently merged by weighted summing, while samples with high disagreement between two methods are discarded. During the decoding, 3 estimated probabilities are transformed to pseudo-likelihoods of 2 states, *Silence* and *Speech*, using priors of 3 classes and manually chosen proportions of 2 states in 3 classes. Decoding is performed with Viterbi algorithm [5].

2.2 End-to-end ASR

End-to-end approach implements ASR system as a single neural network based model that takes a T-length sequence of d dimensional feature vectors $X = \{x_t \in \mathbb{R}^d | t = 1, \ldots, T\}$ in the input and provides a U-length sequence of output labels $Y = \{y_u \in \mathscr{U} | u = 1, \ldots, U\}$, where $\mathscr{U}$ is a set of distinct output labels and usually $U < T$. Common architecture for such models is attention-based encoder-decoder network trained to minimize cross-entropy loss:

$$\mathscr{L}_{\text{att}} = -\log p_{\text{att}}(Y|X) \tag{1}$$

$$p_{\text{att}}(Y|X) = \prod_u p(y_u|X, y_{1:u-1}) \tag{2}$$

$$p(y_u|X, y_{1:u-1}) = \text{Decoder}(\mathbf{r}_u, \mathbf{q}_{u-1}, y_{u-1}) \tag{3}$$

$$\mathbf{h}_t = \text{Encoder}(X) \tag{4}$$

$$a_{ut} = \text{Attention}(\{a_{u-1}\}_t, \mathbf{q}_{u-1}, \mathbf{h}_t) \tag{5}$$

$$\mathbf{r}_u = \sum_t a_{ut} \mathbf{h}_t. \tag{6}$$

Here, Encoder$(\cdot)$ and Decoder$(\cdot)$ are recurrent neural networks, Attention$(\cdot)$ is an attention mechanism and $\mathbf{h}_t$, $\mathbf{q}_{u-1}$ and $\mathbf{r}_u$ are the hidden vectors. Attention mechanism has been developed in the context of machine translation problem [6] and provides a means to model correspondence of all elements of hidden representations sequence to all elements of output sequence in the decoder. Attention mechanism allows to learn non-sequential mapping between its inputs and outputs, meaning that order of output elements is not always the same as order of input elements corresponding to them, what can be an advantage in case of machine translation task, as word order sometimes differs between languages. However, this property of attention mechanism makes training of speech recognition suboptimal, because it is known in advance that word order is the same in audio and in transcription. Connectionist Temporal Classification (CTC) sequence level loss function [7] has been adopted as a secondary learning objective for end-to-end ASR models in order to suppress this drawback:

$$\mathscr{L} = \lambda \mathscr{L}_{\text{ctc}} + (1 - \lambda) \mathscr{L}_{\text{att}}, \tag{7}$$

where $0 \leq \lambda \leq 1$. Encoder output followed by a single linear layer serves as estimated output label sequence in CTC loss calculation, while target is set to be all possible T-length sequences of an extended output labels set $Z = \{z_t \in \mathscr{U} \cup \texttt{<blank>} | t = 1, \ldots, T\}$, corresponding to the original output labels sequence Y:

$$\mathscr{L}_{\text{ctc}} = -\log p_{\text{ctc}}(Y|X) \tag{8}$$

$$p_{\text{ctc}}(Y|X) \triangleq \sum_Z \prod_t p(z_t|z_{t-1}, Y) p(z_t|X) \tag{9}$$

$$p(z_t|X) = \text{Softmax}(\text{Lin}(\mathbf{h}_t)). \tag{10}$$

It has been found that CTC output can also improve decoding results when combined with the main attention-based probabilities during the search:

$$\hat{Y} = \arg\max_{Y} \{\lambda \log p_{\text{ctc}}(Y|X) + (1-\lambda) \log p_{\text{att}}(Y|X)\}. \tag{11}$$

External language model (LM) is commonly-used technique to improve ASR results. LMs are trained on text corpora, which usually contain order of magnitude more examples of written language compared to acoustic corpora, and therefore provide a reliable source of information for selection of well formed transcriptions from hypotheses. In end-to-end ASR, this information is used during the decoding by adding LM probability of hypothetical output label sequence with scaling factor γ to probabilities obtained from the main model:

$$\hat{Y} = \arg\max_{Y} \{\lambda \log p_{\text{ctc}}(Y|X) + (1-\lambda) \log p_{\text{att}}(Y|X) + \gamma \log p_{\text{lm}}(Y)\}. \tag{12}$$

Encoder-decoder architecture allows output sequence (transcription) to have any length that does not exceed length of input sequence (audio recording). Consequently, it is possible to employ different kinds of output units, for example words or characters. In case of words, transcription hypotheses are limited by words presenting in vocabulary, what causes out of vocabulary problems and requires large dimensionality of final layers. In case of characters, output sequences become very long for alphabetical languages, what leads to high number of hypothetical transcriptions and slows down the decoding. Sub-word units have been suggested first as a trade-off solution in machine translation [8] and recently have been adopted in speech recognition [9]. Sub-word units include single characters and can be used to encode any word. In addition to that, sub-word units include combinations of several characters and encode words to shorter sequences compared to single characters. Unigram language model algorithm [10] performs segmentation of a string X by searching for the most probable sequence of sub-word units composing the string:

$$\mathbf{x}^* = \arg\max_{\mathbf{x} \in \mathscr{S}(X)} P(\mathbf{x}), \tag{13}$$

where probability $P(\mathbf{x})$ of a sequence of sub-word units $\mathbf{x} = (x_1, \ldots, x_M)$ is defined as the product of occurrence probabilities of sub-word units:

$$P(\mathbf{x}) = \prod_{i=1}^{M} p(x_i), \tag{14}$$
$$\forall i \; x_i \in \mathscr{V}, \; \sum_{x \in \mathscr{V}} p(x) = 1.$$

Sub-word units vocabulary $\mathscr{V}$ is derived during the training of segmentation model by starting from some large set of frequent in the training data substrings and iterative elimination of certain percent of substrings having lowest impact on total likelihood of all possible sequences of sub-word units for all sentences until some predefined size of vocabulary is reached.

3 Implementation

The frontend is implemented as a Node.js/React application and utilizes WebSocket protocol to communicate with the backend. Users can sign in and upload their recordings for transcription. We plan to add the users' feedback with the main focus on customization and fine tuning.

Speech segmentation is performed with Kaldi toolkit [1]. We use the pretrained SAD model downloaded from `http://kaldi-asr.org/models/m4`. The model is trained on Fisher-English

corpus [11] augmented with room impulses and additive noise from Room Impulse Response and Noise Database [12]. The input features of SAD model are 40-dimensional Mel Frequency Cepstral Coefficients (MFCC) without cepstral truncation with a frame length 25 ms and shift of 10 ms. We use the segmentation parameters suggested in `aspire` Kaldi recipe, but extend maximum speech segment duration from 10 to 30 seconds and enable consecutive speech segments merging when duration of merged segment does not exceed 10 seconds.

Speech recognition is implemented with ESPnet end-to-end speech recognition toolkit [2] with PyTorch backend. We follow LibriSpeech ESPnet recipe and use 80-dimensional log Mel filterbank coefficients concatenated with 3-dimensional pitch having a frame length of 25 ms and shift of 10 ms as acoustic features and sub-word units as output labels. Kaldi toolkit is used to extract and normalize input features. Normalization to zero mean and unit variance is done with global statistics from the training data. SentencePiece unsupervised text tokenizer[2] is used to generate list of sub-word units based on the language model training data and to segment all kinds of text data. We evaluated several sizes of sub-word unit vocabulary between 50 and 5000 and found that 100 resulted in better results for both English and German systems. The ASR model is an encoder-decoder neural network. The encoder network consists of 2 VGG [13] blocks followed by 5 Bidirectional Long Short-Term Memory Network (BLSTM) layers [14] with 1024 units in each layer and direction. The decoder network consists of 2 Long Short-Term Memory Network (LSTM) [15] layers with 1024 units and location based attention mechanism with 1024 dimensions, 10 convolution channels and 100 convolution filters. CTC weight λ is set to 0.5 for both training and decoding. Training is performed with AdaDelta optimizer [16] and gradient clipping on 4 Graphics Processing Units (GPUs) in parallel with a batch size of 24 for 10 epochs. The optimizer is initialized with $\rho = 0.95$ and $\varepsilon = 10^{-8}$. ε is halved after an epoch if performance of the model did not improve on validation set. The model with the highest accuracy on validation set is used for the decoding with beam size of 20.

External LM for the English system contains 2 layers of 650 LSTM units and is trained with stochastic gradient descent optimizer with batch size 256 for 60 epochs. LM scaling factor γ is set to 0.5 during decoding for the English system. External LM for the German system contains 2 layers of 3000 LSTM units and is trained with Adam optimizer [17] with batch size 128 for 10 epochs. LM scaling factor γ is set to 1.1 during decoding for the German system.

4 Resources

Both English and German systems are trained on multiple speech databases, which are summarized in Table 1. We use data preparation scripts from `multi_en` Kaldi recipe and German ASR recipe [18]. German system is additionally improved by data augmentation, applied to 3 datasets (marked with (*) in the table) with Acoustic Simulator[3] package. This procedure gives an augmented dataset that is 10 times larger than original dataset.

External LM for the English system is trained with on transcriptions from the training speech databases except of Common Voice. External LM for the German system is trained on all transcriptions form the training speech databases and additional text corpus[4] containing 8 millions of preprocessed read sentences from the German Wikipedia, the European Parliament Proceedings Parallel Corpus and a crawled corpus of direct speech.

[2] `https://github.com/google/sentencepiece`

[3] `https://github.com/idiap/acoustic-simulator`

[4] `http://ltdata1.informatik.uni-hamburg.de/kaldi_tuda_de/German_sentences_8mil_filtered_maryfied.txt.gz`

Table 1 – English and German training data covering data sets with different styles

Language	Corpus	Style	Hours
English	LibriSpeech [19]	Read	960
	Switchboard [20]	Spontaneous	317
	TED-LIUM 3 [21]	Spontaneous	450
	AMI [22]	Spontaneous	229
	WSJ [23]	Read	81
	Common Voice[5]	Read	240
	Total		*2277*
German	Tuda-De [24]	Read	109
	SWC [25]	Read	245
	M-AILABS[6] (*)	Read	2336
	Verbmobil 1 and 2 [26] (*)	Mixed	417
	VoxForge[7] (*)	Read	571
	RVG 1 [27]	Mixed	100
	PhonDat 1 [28]	Mixed	19
	Total		*3797*

5 ASR Performance

5.1 Results

Table 2 compares the results of IMS-Speech on several testing datasets with the best results for the corresponding datasets which we could find in various sources. In summary, these results suggest that our generic systems can compete with task specific systems and in some cases even outperform them, possibly due to better generalization from larger amount of training data.

Table 2 – ASR performance comparison with state of the art results (WER, %)

Language	Dataset	IMS-Speech	State of the art
English	WSJ eval'92	3.8	3.5 [29]
	LibriSpeech test-clean	4.4	3.2 [30]
	LibriSpeech test-other	12.7	7.6 [30]
	TED-LIUM 3 test	12.8	6.7 [21]
	AMI IHM eval	17.4	19.2[8]
	AMI SDM eval	38.5	36.7[9]
	AMI MDM eval	34.1	34.2[10]
German	Tuda-De dev	11.1	13.1 [18]
	Tuda-De test	12.0	14.4 [18]
	Verbmobil 1 dev	6.7	18.2 [18]
	Verbmobil 1 test	7.3	12.7 [31]

We evaluate the recognition speech with different beam widths and batched recognition with inference using CPU and GPU. The results in Table 3 show that batched recognition can

[5] https://voice.mozilla.org/en/datasets

[6] http://www.m-ailabs.bayern/en/the-mailabs-speech-dataset/

[7] http://www.voxforge.org/de/Downloads

[8] https://github.com/kaldi-asr/kaldi/blob/4bdb05ae78a842a07cae326aeb32aea87328fb2c/egs/ami/s5b/RESULTS_ihm#L87

[9] https://github.com/kaldi-asr/kaldi/blob/4bdb05ae78a842a07cae326aeb32aea87328fb2c/egs/ami/s5b/RESULTS_sdm#L105

[10] https://github.com/kaldi-asr/kaldi/blob/4bdb05ae78a842a07cae326aeb32aea87328fb2c/egs/ami/s5b/RESULTS_mdm#L97

significantly increase speed of recognition without any impact on WER.

Table 3 – Beam width effect on recognition performance and speed on Tuda-De test set

Beam width	Inference on 1 CPU core with batch size of 1		Inference on 1 GPU with batch size of 23	
	WER, %	RT factor	WER, %	RT factor
20	12.0	14.2	12.0	0.7
15	12.2	11.3	12.2	0.5
10	12.6	8.8	12.6	0.4
5	13.7	7.0	13.7	0.3

5.2 Comparisions with Google API

We use the ASR benchmark framework [32] to compare performance of IMS-Speech and Google API. The results of Google API were retrieved on 8.01.2019. As the framework uses custom WER computation method instead of NIST `sclite` utility used in ESPnet recipes, we had to perform scoring of IMS-Speech output with the framework as well. We excluded all utterances for which Google API transcriptions contained digits, because WER would be high for them even if transcriptions were correct (a couple of examples are given in Table 4), and also utterances for which Google API transcriptions were empty. The results are shown in Table 5. The numbers suggest that that Google API models may be optimized for certain speech domain and recording conditions that differ significantly from the ones tested by us.

Table 4 – Examples of some perfect IMS-Speech transcriptions and Google API transcriptions

Utterance	System	Transcription
LibriSpeech test-other, 2609-157645-0010	IMS-Speech	`then let them sing to the hundred and nineteenth replied the curate`
	Google API	`then let them sing the 119th repository`
Verbmobil 1 test, w007dxx0_001_BFG	IMS-Speech	`Ich würde Ihnen den einundzwanzigsten August bis zum vier fünfundzwanzigsten vorschlagen`
	Google API	`ich würde Ihnen den 21. August bis den 425 vorschlagen`

Table 5 – ASR performance comparison with Google API in term of WER (%)

Language	Dataset	IMS-Speech	Google API	Scored utterances
English	LibriSpeech test-clean	4.3	15.9	2444 of 2620 (93%)
	LibriSpeech test-other	12.5	28.0	2708 of 2939 (92%)
	Common Voice valid-test	4.5	19.2	3772 of 3995 (94%)
German	Tuda-De test	10.0	12.4	3481 of 4100 (85%)
	Verbmobil 1 test	8.7	19.5	334 of 631 (53%)

6 Conclusion

We presented IMS-Speech, a web based speech transcription tool for English and German languages that can be used by non-technical researchers in order to utilize the information from audio recordings in their studies. The comparison of the IMS-Speech results with the results of specialized systems in terms of WER showed that the described service can perform decently in a diverse set of tasks and conditions. In the future, we plan to allow the users to customize

the system for their needs as well as to constantly improve our ASR system.

References

[1] POVEY, D., A. GHOSHAL, G. BOULIANNE, L. BURGET, O. GLEMBEK, N. GOEL, M. HANNEMANN, P. MOTLICEK, Y. QIAN, P. SCHWARZ ET AL.: *The Kaldi speech recognition toolkit*. In *Proc. of ASRU*. 2011.

[2] WATANABE, S., T. HORI, S. KARITA, T. HAYASHI, J. NISHITOBA, Y. UNNO, N. E. Y. SOPLIN, J. HEYMANN, M. WIESNER, N. CHEN ET AL.: *ESPnet: End-to-End Speech Processing Toolkit*. *arXiv preprint arXiv:1804.00015*, 2018.

[3] WAIBEL, A., T. HANAZAWA, G. HINTON, K. SHIKANO, and K. J. LANG: *Phoneme recognition using time-delay neural networks*. In *Readings in speech recognition*. 1990.

[4] GHAHREMANI, P., V. MANOHAR, D. POVEY, and S. KHUDANPUR: *Acoustic Modelling from the Signal Domain Using CNNs*. In *Proc. of Interspeech*. 2016.

[5] VITERBI, A.: *Error bounds for convolutional codes and an asymptotically optimum decoding algorithm*. *IEEE Transactions on Information Theory*, 1967.

[6] BAHDANAU, D., K. CHO, and Y. BENGIO: *Neural machine translation by jointly learning to align and translate*. *arXiv preprint arXiv:1409.0473*, 2014.

[7] GRAVES, A., S. FERNÁNDEZ, F. GOMEZ, and J. SCHMIDHUBER: *Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks*. In *Proc. of ICML*. 2006.

[8] SENNRICH, R., B. HADDOW, and A. BIRCH: *Neural machine translation of rare words with subword units*. *arXiv preprint arXiv:1508.07909*, 2015.

[9] ZEYER, A., K. IRIE, R. SCHLÜTER, and H. NEY: *Improved training of end-to-end attention models for speech recognition*. *arXiv preprint arXiv:1805.03294*, 2018.

[10] KUDO, T.: *Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates*. *arXiv preprint arXiv:1804.10959*, 2018.

[11] CIERI, C., D. MILLER, and K. WALKER: *The Fisher Corpus: a Resource for the Next Generations of Speech-to-Text*. In *LREC*. 2004.

[12] KO, T., V. PEDDINTI, D. POVEY, M. L. SELTZER, and S. KHUDANPUR: *A study on data augmentation of reverberant speech for robust speech recognition*. In *Proc. of IEEE ICASSP*. 2017.

[13] SIMONYAN, K. and A. ZISSERMAN: *Very deep convolutional networks for large-scale image recognition*. *arXiv preprint arXiv:1409.1556*, 2014.

[14] GRAVES, A., S. FERNÁNDEZ, and J. SCHMIDHUBER: *Bidirectional LSTM networks for improved phoneme classification and recognition*. In *International Conference on Artificial Neural Networks*, pp. 799–804. Springer, 2005.

[15] HOCHREITER, S. and J. SCHMIDHUBER: *Long short-term memory*. *Neural computation*, 9(8), pp. 1735–1780, 1997.

[16] ZEILER, M. D.: *ADADELTA: an adaptive learning rate method. arXiv preprint arXiv:1212.5701*, 2012.

[17] KINGMA, D. P. and J. BA: *Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980*, 2014.

[18] MILDE, B. and A. KÖHN: *Open Source Automatic Speech Recognition for German*. In *Proc. of ITG*. 2018.

[19] PANAYOTOV, V., G. CHEN, D. POVEY, and S. KHUDANPUR: *Librispeech: an ASR corpus based on public domain audio books*. In *Proc. of IEEE ICASSP*. 2015.

[20] GODFREY, J. J., E. C. HOLLIMAN, and J. MCDANIEL: *SWITCHBOARD: Telephone speech corpus for research and development*. In *Proc. of IEEE ICASSP*. 1992.

[21] HERNANDEZ, F., V. NGUYEN, S. GHANNAY, N. TOMASHENKO, and Y. ESTEVE: *TED-LIUM 3: twice as much data and corpus repartition for experiments on speaker adaptation. arXiv preprint arXiv:1805.04699*, 2018.

[22] CARLETTA, J.: *Unleashing the killer corpus: experiences in creating the multi-everything AMI Meeting Corpus. Language Resources and Evaluation*, 2007.

[23] PAUL, D. B. and J. M. BAKER: *The design for the Wall Street Journal-based CSR corpus*. In *Proc. of the workshop on Speech and Natural Language*. 1992.

[24] RADECK-ARNETH, S., B. MILDE, A. LANGE, E. GOUVÊA, S. RADOMSKI, M. MÜHLHÄUSER, and C. BIEMANN: *Open source german distant speech recognition: Corpus and acoustic model*. In *Text, Speech, and Dialogue*. 2015.

[25] KÖHN, A., F. STEGEN, and T. BAUMANN: *Mining the Spoken Wikipedia for Speech Data and Beyond*. In *Proc. of LREC*. 2016.

[26] WAHLSTER, W.: *Verbmobil: foundations of speech-to-speech translation*. Springer Science & Business Media, 2013.

[27] BURGER, S. and F. SCHIEL: *RVG 1 – A Database for Regional Variants of Contemporary German*. In *Proc. of LREC*. 1998.

[28] HESS, W. J., K. J. KOHLER, and H.-G. TILLMANN: *The Phondat-verbmobil speech corpus*. In *Fourth European Conference on Speech Communication and Technology*. 1995.

[29] CHAN, W. and I. LANE: *Deep recurrent neural networks for acoustic modelling. arXiv preprint arXiv:1504.01482*, 2015.

[30] HAN, K. J., A. CHANDRASHEKARAN, J. KIM, and I. LANE: *The CAPIO 2017 conversational speech recognition system. arXiv preprint arXiv:1801.00059*, 2017.

[31] GAIDA, C., P. LANGE, R. PETRICK, P. PROBA, A. MALATAWY, and D. SUENDERMANN-OEFT: *Comparing open-source speech recognition toolkits. Tech. Rep., DHBW Stuttgart*, 2014.

[32] DERNONCOURT, F., T. BUI, and W. CHANG: *A Framework for Speech Recognition Benchmarking. Proc. of Interspeech*, 2018.