Rethinking Evaluation in ASR: Are Our Models Robust Enough?

Tatiana Likhomanenko* Facebook, Menlo Park antares@fb.com

Qiantong Xu* Facebook, Menlo Park qiantong@fb.com

Vineel Pratap* Facebook, Menlo Park vineelkpratap@fb.com

Paden Tomasello Facebook, Menlo Park padentomasello@fb.com

Jacob Kahn Facebook, Menlo Park jacobkahn@fb.com

Gilad Avidov Facebook, Menlo Park avidov@fb.com

Ronan Collobert Facebook, Menlo Park locronan@fb.com

Gabriel Synnaeve Facebook, Paris gab@fb.com

April 9, 2021

Abstract

Is pushing numbers on a single benchmark valuable in automatic speech recognition? Research results in acoustic modeling are typically evaluated based on performance on a single dataset. While the research community has coalesced around various benchmarks, we set out to understand generalization performance in acoustic modeling across datasets – in particular, if models trained on a single dataset transfer to other (possibly out-of-domain) datasets. Further, we demonstrate that when a large enough set of benchmarks is used, average word error rate (WER) performance over them provides a good proxy for performance on real-world data. Finally, we show that training a single acoustic model on the most widely-used datasets – combined – reaches competitive performance on both research and real-world benchmarks.

1 Introduction

Progress in automatic speech recognition (ASR) is measured on the validation and test sets of standard datasets. However, most acoustic models (AMs) are often developed and tuned on a single dataset and transfer poorly to other datasets. Moreover, most large standard benchmarks have similar domains and recording conditions. These factors lead to siloed ASR research. A unified benchmark comprised of conversational, oratory, and read speech with varied recording conditions and noise would certainly serve the research community well; here, however, we study how the currently-popular public benchmarks can be used to gauge model generalization performance.

Our approach constructs a validation procedure – using only public datasets – that is a better predictor of overall and domain transfer performance than datasets taken in isolation. We train the same state-of-the-art model architecture on different benchmarks pushing for best performance on each benchmark separately. We also jointly train a model on all datasets. Given the transfer performance on test sets, we can ascertain which test sets are good proxies for transfer performance as well as which training sets can produce the best-performing models. This informs us on the robustness of various datasets in transfer and which test sets are the best predictors of ASR performance in others. Finally, we look at the performance, in transfer only, on our in-house ASR datasets. This informs us about which sets of test sets should be used if one wants to transfer to a wide range of conditions of speech.

2 Related Work

Previous works that study transfer in ASR include [Ghahremani et al., 2017] that studied transferring varying number of layers trained out-of-domain, from SwitchBoard to AMI-IHM or from LibriSpeech to AMI-IHM. In this paper

*Equal contribution.

as in ours, a joint model trained on multiple out-of-domain datasets exhibits better transfer. In the context of the Arabic MGB-3 challenge, Manohar et al. [2017] transfered AMs trained on broadcast TV to Youtube videos, with a different setting than here as the training transcriptions were noisily labeled. Distillation was used to improved transfer in [Asami et al., 2017], where the soft-target part of the distillation loss may help with regularization. For another kind of transfer in [Kunze et al., 2017], the authors transferred LibriSpeech trained wav2letter [Collobert et al., 2016] models to German by fine-tuning them on German, with better performance than training from scratch. Very recently, Szymański et al. [2020] point out some limitations of current ASR benchmarks, and propose guidelines to create multi-domain datasets. Finally, while DeepSpeech 2 [Amodei et al., 2016] did not focus their study on transfer, we train a single AM on multiple datasets at once, as they did.

3 Domain Transfer

In order to study transfer across datasets and conditions, we do a systematic analysis. In all our experiments, we use a single Transformer-based AM architecture with 270M parameters, to make our results comparable across the board. We train multiple single-dataset baselines as well as one joint model trained on all datasets at once. We then evaluate this set of models on all the validation and test sets, to measure how each "in-domain" model transfers to "out-of-domain" datasets. From this, we analyze which datasets suffer more acutely from "domain overfitting." Evidently, it is difficult to separate the "in-domainness" and size of a dataset; e.g., we cannot directly compare results on WSJ (80h) to ones on LibriSpeech (960h). We also fine-tune our joint model on the transfer dataset with 1h, 10h, and 100h of in-domain data. Finally, we examine how our models transfer to real data and in the process observe that public validation and test sets performance is predictive of the transfer performance of a model to real data.

4 Experiments

4.1 Datasets

To measure domain transfer, we restrict experiments to use only datasets in English, for which there exist several commonly-used and publicly available datasets with hundreds hours of transcribed audio. Validation sets from each dataset are used to optimize model configurations and to perform all hyper-parameter tuning, while test sets are used for final evaluation only.

LibriSpeech (LS) [Panayotov et al., 2015] consists of read speech from audiobook recordings. We use standard split of train, validation (*dev-clean*, *dev-other*) and test sets (*test-clean*, *test-other*).

SwitchBoard & Fisher (SB+FSH) consists of conversational telephone speech. To create a training set, we combine Switchboard [Godfrey and Holliman, 1993] and Fisher [Cieri et al., 2004, 2005a,,]. We use RT-03S [Fiscus et al., 2007] as the validation set; test sets are the Hub5 Eval2000 [LDC et al., 2002] data with two subsets, SwitchBoard (SB) and CallHome (CH). For the data processing and evaluation, we follow the recipe provided by Kaldi [Povey and other, 2011].

Wall Street Journal (WSJ) [Garofolo et al., 1993, LDC and Group, 1994, Woodland et al., 1994]. We consider the standard subsets *si284*, *nov93dev* and *nov92* for training, validation and test, respectively. We remove any punctuation tokens from *si284* transcriptions when used for training.

Mozilla Common Voice (CV) project [Ardila et al., 2020]. The CV dataset consists of transcribed audio in various languages where speakers record text from Wikipedia. Anyone can submit recorded contributions; as a result, the dataset has a large variation in quality and speakers. We use the English dataset[2], where data splits are provided therein.

TED-LIUM v3 (TL) [Hernandez et al., 2018] is based on TED conference videos. We use the last edition of the training set from this dataset (v3), for which the validation and test sets are kept consistent (and thus numbers are comparable) with the earlier releases. We follow the Kaldi recipe [Povey and other, 2011] for data preparation.

Robust Video (RV) is our in-house English video dataset, which are sampled from public social media videos and aggregated and deidentified before transcription. These videos contain a diverse range of speakers, accents, topics, and acoustic conditions making ASR difficult. The test sets are composed of *clean*, *noisy* and *extreme* with *extreme* being the most acoustically challenging subset among them. The validation set comprises of data from *clean* and *noisy* subsets.

[2] June 22nd 2020's snapshot: `https://tinyurl.com/cvjune2020`. Transcriptions contain upper-case and non-English characters and punctuation. To have similar transcription normalization as in other datasets, we normalize the text for all splits: lower-casing, Unicode normalization, removing punctuation and non-English tokens, and mapping common abbreviations (e.g. "mr." to "mister").

Table 1: Statistics on datasets: sampling frequency, duration (in hours), and speech type.

Data	kHz	Train (h)	Valid (h)	Test (h)	Speech
WSJ	16	81.5	1.1	0.7	read
TL	16	452	1.6	2.6	oratory
CV	48	693	27.1	25.8	read
LS	16	960	5.1+5.4	5.4+5.4	read
SB+FSH	8	300+2k	6.3	1.7+2.1	convers.
RV	16	5k	14.4	18.8+19.5+37.2	diverse

Table 2: Statistics on datasets: mean sample duration (in seconds) and mean sample transcription length (in words).

Data	Train $\mu \pm \sigma$ (s)	Valid $\mu \pm \sigma$ (s)	Test $\mu \pm \sigma$ (s)	Train $\mu \pm \sigma$ (wrd)	Valid $\mu \pm \sigma$ (wrd)	Test $\mu \pm \sigma$ (wrd)
WSJ	7.8 ± 2.9	7.8 ± 2.9	7.6 ± 2.5	17 ± 7	16 ± 7	17 ± 6
TL	6 ± 3	11.3 ± 5.7	8.1 ± 4.3	17 ± 10	35 ± 20	24 ± 15
CV	5.7 ± 1.6	6.1 ± 1.8	5.8 ± 2.6	10 ± 3	10 ± 3	9 ± 3
LS	12.3 ± 3.8	6.8 ± 4.5	7 ± 4.8	33 ± 12	19 ± 13	19 ± 13
SB+FSH	3.7 ± 3.2	4 ± 3.1	2.1 ± 1.7	11 ± 12	12 ± 12	8 ± 8

Table 3: Perplexity (including out-of-vocabulary words) of word-level LMs. We use 4-gram LM for WSJ, LS, SB+FSH, and 5-gram for TL, CV.

Data/Vocab	in-dom. n-gram		in-dom. Transf.		CC 4-gram	
	Valid	Test	Valid	Test	Valid	Test
WSJ/162K	159	134	84	69	297	285
TL/200k	119	149	74	79	142	136
CV/168K	359	329	181	188	213	157
LS/200K	155/147	164/154	48/50	52/50	258/258	244/249
SB+FSH/64K	124	114/112	50	55/67	221	199/153

4.2 Unifying Audio

The datasets used in our work have different sample rate and varied input lengths as shown in Table 1 and 2. Since we require the same set of filterbanks for joint training across all datasets, we upsample/downsample each dataset to 16Khz and use this setup for training both baseline models on individual datasets as well as joint models. For all experiments we compute 80 log-mel spectrogram features for a 25ms sliding window, strided by 10ms. All features are normalized to have zero mean and unit variance per input sequence before feeding into the neural network.

On SB+FSH individual baseline, we span the log-mel filterbanks up to only 4kHz (unlike 8kHz for all other training setups) as any spectrogram features beyond 4kHz cannot be determined accurately [Shannon, 1949]. This can also be

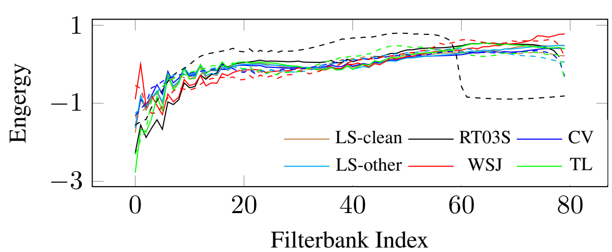

Figure 1: Distribution of the mean normalized energy of 80 filterbanks on all the public validation sets we used, for 16kHz audio (dashed) and 8kHz audio (solid).

Table 4: WER of models evaluated on all datasets (downsampled to 8kHz) with a greedy decoding and *no LM* (top), *with in-domain n-gram LM* beam-search decoding (middle) and with additional second-pass rescoring *by in-domain Transformer LM* (below), with beam-search decoding and 4-gram CC LM for joint model (joint CC). SOTA models are given from WSJ [Hadian et al., 2018], TED-LIUM [Zhou and other, 2020], LibriSpeech [Gulati et al., 2020], SwitchBoard & Fisher [Han et al., 2017]. The average is computed as average of averages for LibriSpeech's validations/tests, and SwitchBoard's tests (SB, CH) sets, so as not to weight them more heavily.

Train	WSJ		TL		CV		LS				SB+FSH			average	
	nov93	nov92	valid	test	valid	test	dev-c	test-c	dev-o	test-o	RT03S	SB	CH	valid	test
SOTA		2.8	5.1	5.6				1.9		3.9	8.0	5.0	9.1		
WSJ	13.5	11.7	48.8	42.1	72.2	78.0	32.4	32.7	53.8	54.0	78.3	69.6	84.6	51.2	50.4
	7.3	5.3	35.2	28.7	55.0	62.0	17.7	18.2	37.4	38.8	68.0	58.2	78.3	38.6	38.6
	5.7	4.1	33.9	26.9	53.5	60.7	14.6	15.2	35.2	36.7	66.8	57.4	77.5	36.9	37.0
TL	12.0	9.9	10.2	7.9	32.2	36.7	11.8	12.4	21.6	22.6	32.1	23.6	32.4	20.6	20.0
	7.6	5.8	7.9	6.5	23.9	28.1	7.5	8.4	15.3	16.4	27.3	19.3	27.7	15.6	15.3
	6.3	5.1	7.4	6.2	22.6	27.3	5.9	6.9	13.1	14.1	26.8	19.0	27.3	14.5	14.5
CV	12.8	9.7	68.5	47.0	12.0	15.4	35.5	36.3	37.0	39.1	49.2	46.9	45.7	35.8	31.2
	6.0	3.5	56.6	34.2	10.1	13.0	23.8	25.6	26.6	29.1	39.2	36.2	36.6	27.4	22.9
	5.5	3.2	55.3	32.4	10.0	12.8	21.9	23.7	24.4	26.9	37.8	35.0	35.5	26.3	21.8
LS-960	10.0	7.9	13.6	13.0	25.8	29.9	2.6	2.7	7.0	6.8	36.6	27.6	35.0	18.2	17.4
	4.7	3.5	9.0	9.7	18.6	22.3	2.0	2.5	5.2	5.5	28.3	20.0	27.8	12.8	12.7
	3.9	2.9	8.5	8.8	17.6	21.6	1.5	2.0	4.2	4.5	27.7	20.0	27.1	12.1	12.0
SB+FSH	10.9	9.5	15.1	12.4	49.4	50.7	14.0	14.4	27.4	28.6	12.0	6.9	11.4	21.6	20.6
	5.1	4.0	9.3	8.9	40.4	41.8	7.5	8.2	18.8	20.4	10.4	6.5	10.3	15.7	15.4
	4.2	3.4	8.6	8.0	38.9	40.5	5.4	6.2	16.3	18.1	10.4	6.5	10.3	14.6	14.5
Joint	3.0	2.0	6.1	5.7	11.1	13.2	2.5	2.5	6.0	5.9	10.7	5.8	9.7	7.0	6.6
	2.0	1.4	5.4	5.5	9.4	11.1	1.8	2.3	4.5	4.8	9.4	5.4	8.6	5.9	5.7
	1.7	**1.3**	**5.0**	**4.7**	**9.2**	**10.9**	**1.4**	**2.0**	**3.7**	**4.1**	**9.4**	**5.4**	**8.6**	**5.6**	**5.4**
Joint CC	2.8	2.0	5.6	5.1	8.1	9.4	2.9	2.9	5.3	5.3	9.6	5.4	8.7	6.0	5.5

seen in Figure 1 which plots the distribution of mean normalized energy of filterbanks for different datasets with audio sampled at 16kHz and filterbanks span from 0-8kHz.

4.3 Baselines and Joint Model

Acoustic Model (AM) All models are trained with Connectionist Temporal Classification [Graves et al., 2006] and the network architecture follows [Synnaeve et al., 2019]: the encoder of our AMs is composed of a convolutional frontend (1-D convolution with kernel-width 7 and stride 3 followed by GLU activation) followed by sinusoidal positional embedding and 36 4-heads Transformer blocks [Vaswani et al., 2017] (we don't use relative positional embedding inside Transformer blocks). The self-attention dimension is 768 and the feed-forward network (FFN) dimension is 3072 in each Transformer block. The output of the encoder is followed by a linear layer to the output classes. We use dropout after the convolution layer. For all Transformer layers, we use dropout on the self-attention and on the FFN, and layer drop [Fan et al., 2020], dropping entire layers at the FFN level. Dropout and layer dropout values are tuned for each model separately. Token set for all AMs consists of 26 English alphabet letters, augmented with the apostrophe and a word boundary token. The popular approach with word-pieces as tokens set we found to be not suited as intersection between word-pieces constructed on every training set less than 50%. Thus the question what word-pieces set should be used for the joint model is still open. SpecAugment [Park et al., 2019] is used for data augmentation in training: there are two frequency masks, and ten time masks with maximum time mask ratio of $p = 0.05$; frequency and time mask parameters are tuned separately for each model; time warping is not used. In the joint model, the maximum frequency bands masked by one frequency mask is 30, and the maximum frames masked by the time mask is 30, too. We use the Adagrad optimizer [Duchi et al., 2011] and decay learning rate by a factor of 2 each time the WER reaches a plateau on the validation sets. All experiments are implemented within flashlight[3] and wav2letter++ [Pratap et al., 2019]. All models are trained with dynamic batching (effective average batch size is 240s per GPU) and mixed-precision computations on 16 GPUs (Volta 32GB) for 1-3 days for single dataset baselines and 14 days for joint training.

Language Model (LM) and Beam-search Decoding For each dataset we train in-domain word-level n-gram LM

[3] https://github.com/facebookresearch/flashlight

using KenLM toolkit [Heafield, 2011] and Transformer LM as in [Synnaeve et al., 2019]. In-domain LM training uses: for WSJ and LS – their provided LM data; for CV, SB+FSH, and RV – train transcriptions only; for TL – both train transcriptions and provided LM data. We also train a 4-gram LM on Common Crawl (CC) data with 200k top words and pruning of all 3,4-grams appearing once. Perplexity of all LMs is shown in Table 3. We rely on the one-pass beam-search decoder from the wav2letter++ [Collobert et al., 2016] (lexicon-based with a n-gram LM) and second-pass rescoring with a Transformer LM following [Synnaeve et al., 2019].

Joint Model We adopt the same AM architecture described above but with less regularization when training on the combination of all the datasets. We weight each sample equally, i.e. each sample from each dataset is fed into the model once in each epoch.

State-of-the-Art Models In Table 4 for each dataset, we report known state-of-the-art models with in-domain LMs.

4.4 AM Transfer

In general, an AM trained in isolation on a single dataset performs poorly on other datasets, as shown in Table 4. The model trained on WSJ performs the worst (part of the reason could be the smaller amount of training data) for transfer, while other models transfer very well to WSJ. All models transfer poorly to CV and the CV model transfers poorly to other datasets, which may indicate that CV is very different from other benchmarks. From the results on LS, TL and SB+FSH there is a similarity between LS and TL (they transfer the best to each other). There is also a similarity in transfer between SB+FSH and TL benchmarks, however, LS and SB+FSH do not transfer well to each other. When training on all datasets at once, the joint model in Table 4 performs better or close to a single dataset training. This behaviour compared to results on a single dataset training indicates that i) datasets differ from each other and ii) a robust model scoring well on all these benchmarks exists.

In Table 5, we report results of transfer, of those same models trained on public datasets, to our in-house RV dataset. We also report numbers from a baseline system that is trained in-domain on a corresponding training set of 5000h. As for other benchmarks, single dataset training transfers poorly to in-house data, however, the transfer quality varies a lot, having the best results from the TL model. At the same time our joint model, which performs well on each benchmark, transfers really well, stating that i) public datasets could be the good proxy of training data for real-world ASR, ii) improving average performance on public benchmarks leads to improving performance on real-world noisy data. Moreover, fine-tuning of this joint model on 1h closes the gap with the RV baseline model and fine-tuning with 10h or 100h of data and decoding with CC LM surpasses WER compared to the RV baseline model decoded with in-domain LM.

4.5 Transfer with LM

Single-dataset AMs get a boost in WER performance when decoding/rescoring with an in-domain LM, as shown in Table 4. These AMs perform however poorly in transfer domain conditions (see Tables 4 and 5). In contrast, the joint model transfers well to in-house RV data, when decoded with an in-domain LM (see Table 5). Decoding the joint model with the large generic CC LM leads to WER performance which is overall improved, on both public and in-house RV datasets.

4.6 Predictors of transfer

We performed single variable linear regressions using data from Table 4: lines as datapoints, test set score columns as features, and labels being the same models' performance in transfer on the average of test clean, noisy, and extreme, from Table 5. Across all datasets, and taken over all trained models, the best "single test set" predictor for out-of-domain performance on RV data is SB with an $r^2 = 0.8$ (rejecting the null hypothesis with $p < 0.001$), the worst single predictor being WSJ's nov92 with $r^2 = 0.5$ ($p < 0.001$). We also performed multivariate regressions using all the test results from Table 4 and only the results for the models decoded with n-gram LMs. This gives an overspecified problem (more variables: 7, than models: 6), so OLS gives a "perfect" (overfitted, $r^2 = 1$) solution which weights nov92, test-clean, test-other *negatively*. We repeat this regression with heavy L1 regularization (Lasso, as proxy for L0 norm regularization) and it yields a regression with $r^2 = 0.98$ (although only 6 datapoints) with only 3 test sets weighted non-zero, and positively: TL, CV, and SB. We can conclude that those test sets are the most predictive of the performance in transfer on RV of our Transformer-based AMs decoded with n-grams. A larger study across AMs and LMs variants should provide a more robust conclusion.

Table 5: WER comparison with *a greedy decoding* and with *a* 5-*gram in-domain LM* and/or *the 4-gram CC LM* beam-search decoding on RV validation and test data from videos. Except for the "RV" training and for models with "+finetune", all other models correspond to models in Table 4.

Train	LM	Valid	Test		
			clean	noisy	extreme
RV	-	18.4	17.1	22.4	31.8
	in-dom.	12.8	15.7	20.9	29.8
WSJ	-	69.6	67.7	74.3	84.8
	in-dom.	56	54.9	62.4	71.8
TL	-	29.5	26	34.4	46.5
	in-dom.	22.1	21.4	29.4	40.6
CV	-	42.2	34.7	45.7	58
	in-dom.	31.6	27.3	37.7	49.4
LS-960	-	36.9	32.7	42.7	58.3
	in-dom.	24.4	24.6	33.5	45
SB+FSH	-	35.7	31.6	37.0	45.3
	in-dom.	28.6	26.6	32.5	41.0
Joint	-	23.6	19.2	25.5	35.0
	in-dom.	17.9	16.1	21.9	31.4
	CC	20.6	15.8	21.7	31.2
Joint + finetune RV-1h	-	22.5	18.4	23.6	34.3
	in-dom.	16.7	15.2	21.2	30.3
	CC	19.5	15.0	20.9	30.1
Joint + finetune RV-10h	-	20.8	17.1	23.4	33.0
	in-dom.	15.7	14.6	20.5	29.8
	CC	18.5	14.1	20.2	29.5
Joint + finetune RV-100h	-	18.9	15.5	21.2	31.4
	in-dom.	14.3	13.3	18.7	28.2
	CC	16.8	12.9	18.2	27.7

5 Conclusion

We studied transfer across five public datasets, as well as transfer to out-of-domain, real-world audio data, for a single AM architecture based on Transformers and with n-gram and Transformer-based LMs for decoding. We showed that no single validation or test set from public datasets is sufficient to measure transfer to other public datasets or to real-world audio data. Our results suggests that ASR researchers interested in producing transferable AMs should, at the very least, report results on SwitchBoard, CommonVoice, and TED-LIUM (v3). Finally, we provided a recipe for a community-reproducible robust ASR model, which can be trained with a couple of public audio datasets, and an LM built on Common Crawl.

References

D. Amodei et al. Deep Speech 2: End-to-End Speech Recognition in English and Mandarin. In *ICML*, 2016.

R. Ardila et al. Common voice: A massively-multilingual speech corpus. In *LREC*, 2020.

T. Asami et al. Domain adaptation of dnn acoustic models using knowledge distillation. In *ICASSP*, 2017.

C. Cieri, D. Graff, O. Kimball, D. Miller, and K. Walker. Fisher english training speech parts 1 and 2 transcripts LDC200{4,5}T19. *Philadelphia: LDC*, 2004, 2005a.

C. Cieri, D. Miller, and K. Walker. Fisher english training speech parts 1 and 2 LDC200{4,5}S13. *Philadelphia: LDC*, 2004, 2005b.

R. Collobert, C. Puhrsch, and G. Synnaeve. Wav2letter: an end-to-end convnet-based speech recognition system. *arXiv preprint arXiv:1609.03193*, 2016.

J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12, 2011.

A. Fan, E. Grave, and A. Joulin. Reducing transformer depth on demand with structured dropout. In *ICML*, 2020.

J. G. Fiscus et al. 2003 nist rich transcription evaluation data LDC2007S10. *Web Download. Philadelphia: LDC*, 2007.

J. Garofolo, D. Graff, D. Paul, and D. Pallett. CSR-I (WSJ0) complete LDC93S6A. *Web Download. Philadelphia: LDC*, 1993.

P. Ghahremani et al. Investigation of transfer learning for ASR using LF-MMI trained neural networks. In *ASRU*, 2017.

J. Godfrey and E. Holliman. Switchboard-1 release 2 LDC97S62. *Philadelphia: LDC*, 1993.

A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *ICML*, 2006.

A. Gulati et al. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*, 2020.

H. Hadian et al. End-to-end speech recognition using lattice-free MMI. In *Interspeech*, 2018.

K. J. Han, A. Chandrashekaran, J. Kim, and I. Lane. The CAPIO 2017 conversational speech recognition system. *arXiv preprint arXiv:1801.00059*, 2017.

K. Heafield. KenLM: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation*. Association for Computational Linguistics, 2011.

F. Hernandez et al. TED-LIUM 3: twice as much data and corpus repartition for experiments on speaker adaptation. In *SPECOM*, 2018.

J. Kunze et al. Transfer learning for speech recognition on a budget. In *ACL Workshop on Representation Learning for NLP*, 2017.

LDC and N. M. I. Group. CSR-II (WSJ1) complete LDC94S13A. *Web Download. Philadelphia: LDC*, 1994.

LDC et al. 2000 hub5 english evaluation speech LDC2002S09 and transcripts LDC2002T43. *Web Download. Philadelphia: LDC*, 2002.

V. Manohar, D. Povey, and S. Khudanpur. JHU Kaldi system for Arabic MGB-3 ASR challenge using diarization, audio-transcript alignment and transfer learning. In *ASRU*, 2017.

V. Panayotov et al. Librispeech: an ASR corpus based on public domain audio books. In *ICASSP*, 2015.

D. S. Park et al. SpecAugment: A simple data augmentation method for automatic speech recognition. In *Interspeech*, 2019.

D. Povey and other. The Kaldi speech recognition toolkit. In *ASRU*, 2011.

V. Pratap et al. wav2letter++: The fastest open-source speech recognition system. In *ICASSP*, 2019.

C. E. Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, 1949.

G. Synnaeve et al. End-to-end ASR: from supervised to semi-supervised learning with modern architectures. *arXiv preprint arXiv:1911.08460*, 2019.

Szymański et al. Wer we are and wer we think we are. *arXiv preprint arXiv:2010.03432*, 2020.

A. Vaswani et al. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

P. C. Woodland et al. Large vocabulary continuous speech recognition using HTK. In *ICASSP*, 1994.

W. Zhou and other. The RWTH ASR system for TED-LIUM release 2: Improving hybrid HMM with specaugment. In *ICASSP*, 2020.