Low-Resource ASR with an Augmented Language Model

Timofey Arkhangelskiy
Universität Hamburg
timarkh@gmail.com

Abstract

It is widely known that a good language model (LM) can dramatically improve the quality of automatic speech recognition (ASR). However, when dealing with a low-resource language, it is often the case that not only aligned audio data is scarce, but there are also not enough texts to train a good LM. This is the case of Beserman, an unwritten dialect of Udmurt (Uralic > Permic). With about 10 hours of aligned audio and about 164K words of texts available for training, the word error rate of a Deepspeech model with the best set of parameters equals 56.4%. However, there are other linguistic resources available for Beserman, namely a bilingual Beserman-Russian dictionary and a rule-based morphological analyzer. The goal of this paper is to explore whether and how these additional resources can be exploited to improve the ASR quality. Specifically, I attempt to use them in order to expand the existing LM by generating a large number of fake sentences that in some way look like genuine Beserman text. It turns out that a sophisticated enough augmented LM generator can indeed improve the ASR quality. Nevertheless, the improvement is far from dramatic, with about 5% decrease in word error rate (WER) and 2% decrease in character error rate (CER).

Abstract

Ваньзылы тодмо, умой лэсьтэм кыл модель вераськемез асэрказ тодманлэсь ӟечлыксэ тросльі будэтыны быгатэ шуыса. Озьы ке но, куке вераськон мынэ пичи кылъёс сярысь, кызьы ке распознавателез дышетон понна волятэм куара, озьы ик умой кыл моделез дышетон понна текстъёс ӵемысь туж ӧжыт луо. Ӵапак таӵе югдур удмурт кыллэн гожъяськеттэм бесерман вераськетэныз кылдэмын. Ки уламы вань 10 час пала волятэм но расшифровать карем куара но 164 сюрс пала уже кутэм кылъёсын текстъёс. Та тодэтъёсын дышетскыса, Deepspeech система возьматэ 56,4% WER (мыдлань распознать карем кылъёслэн процентсы). Озьы ке но бесерман вераськетъя вань на мукет кылтодон ванёсъёс: бесерман-ӟуч кыллюкам но шонеррадъян морфологи анализатор. Та ужлэн целез — валаны, луэ-а вераськемез асэрказ тодманлэсь ӟечлыксэ будэтон понна та ватсам ресурсъёсты уже кутыны. Кылсярысь, соос вылэ пыкъяськыса, турттэмын кыл моделез паськытатыны, со понна кылдытэмын вал трос зэмос луисьтэм шуосъёс, кудъёсыз куд-ог ласянь тупало зэмос бесерман текстлы. Шуосъёсты кылдытйись генератор тырмыт «визьмо» ке, распознаванилэн ӟечлыкез зэмзэ но будэ вылэм. Озьы ке но та умоян шӧдскымон луэ шуыса, вераны уг луы: WER возьматон усе 5%-лы пала, нош CER (мыдлань распознать карем букваослэн процентсы) — 2%-лы.

Abstract

Известно, что хорошая языковая модель может существенно повысить качество автоматического распознавания речи. Однако если речь идёт о некрупном языке, зачастую имеется не только слишком мало выровненного звука для

обучения распознавателя, но и слишком мало текстов для обучения хорошей языковой модели. Именно таков случай бесермянского – бесписьменного диалекта удмуртского языка. В нашем распоряжении имеются около 10 часов звука, выровненного с расшифровками, и тексты объёмом около 164 тыс. словоупотреблений. Обучившись на этих данных, система Deepspeech демонстрирует WER (процент неправильно распознанных слов), равный 56,4%. Однако для бесермянского существуют другие лингвистические ресурсы, а именно бесермянско-русский словарь и правиловый морфологический анализатор. Цель этой работы – выяснить, можно ли использовать эти дополнительные ресурсы для улучшения распознавания речи. В частности, предпринимается попытка расширить с их помощью языковую модель путём порождения большого количества ненастоящих предложений, которые в некоторых отношениях похожи на настоящий бесермянский текст. Оказывается, что если генератор предложений достаточно “умён”, качество распознавания после этого действительно возрастает. Однако это улучшение вряд ли можно назвать существенным: показатель WER падает примерно на 5%, а CER (процент неправильно распознанных букв) – на 2%.

1 Introduction

The key to reaching good ASR quality is having lots of data, i.e. thousands or at least hundreds of hours of text-aligned sound recordings. For most languages in the world, however, resources of that size are unavailable. With only a dozen hours of sound at hand, it is currently impossible to reach a WER low enough for the system to be usable in real-world applications. Nevertheless, a system with a WER, which is high, but lower than a certain threshold (e.g. 50%), could still be used in practice. Specifically, the primary motivation behind this research was the need to transcribe large amounts of spoken Beserman for subsequent linguistic research. If an ASR system, despite its high WER, could facilitate and accelerate manual transcription, that would be a useful practical application, even if limited in scope. Other possible applications of such under-trained noisy ASR systems have been proposed by Tyers and Meyer (2021). This is why it makes sense to experiment with datasets that small.

A number of techniques have been used to achieve better results in low-resource ASR systems. This includes pre-training the model on the data from another (possibly related or phonologically similar) language (Stoian et al., 2020), augmenting the sound data with label-preserving transformations (Tüske et al., 2014; Park et al., 2019), and training the LM on a larger set of texts taken e.g. from a written corpus (Leinonen et al., 2018). That a good language model can play an important role can be seen e.g. from the experiments on ASR for varieties of Komi, a language closely related to Udmurt, as described by (Hjortnaes et al., 2020b) and (Hjortnaes et al., 2020a). Replacing a LM with a larger and more suitable one (in terms of domain) can decrease WER significantly.

Beserman is traditionally classified as a dialect of Udmurt (Uralic > Permic) and is spoken by around 2200 people in NW Udmurtia, Russia. Unlike standard Udmurt, it lacks a codified orthography and is not used in the written form outside of scientific publications. This paper describes experiments with training Deepspeech (Hannun et al., 2014) on transcribed and elicited Beserman data. I am particularly interested in augmenting the LM with the help of linguistic resources that exist for Beserman: a Beserman-Russian dictionary and a morphological analyzer. The former is used, among other things, to transfer information from a model trained on Russian data. Same kinds of data augmentation could be relevant for many other under-resourced languages and dialects, since bilingual dictionaries and rule-based tools often exist for varieties, which are poor in raw data.

The paper is organized as follows. In Section 2, I describe the dataset and lay out the reasons why improving the LM could be challenging. In Section 3, the training setup is outlined. In Section 4, I describe how the artificially augmented LM was generated. In 5, the original results are compared to that of the augmented LM. This is followed by a conclusion.

2 The data

The Beserman dataset I have at hand consists of about 15,000 transcribed sound files with recordings from 10 speakers, both male and female, total-

ing about 10 hours (with almost no trailing silence). Most of them come from a sound-aligned Beserman corpus, whose recordings were made in 2012–2019 and have varying quality. Another 2,700 files, totaling 2.5 hours, come from a sound dictionary and contain three pronunciations of a headword each. The duration of most files lies between 1 and 5 seconds. In addition to the texts of the sound-aligned corpus, there are transcriptions of older recordings, which are not sound-aligned as of now, and a corpus of usage examples based on the Beserman-Russian dictionary[1] (Arkhangelskiy, 2019). All these sources combined contain about 27,400 written Beserman sentences (some very short, some occurring more than once), with a total of 164K words.

Such amount of textual data is insufficient for producing a well performing LM. Since Beserman is a morphologically rich language, most forms of most lexemes are absent from the sample and thus cannot be recognized, being out-of-vocabulary words. Unlike in some other studies mentioned above, it is hardly possible to find Beserman texts elsewhere. One way of doing that would be to use texts in literary Udmurt, which are available in larger quantities (tens of millions of words). Although I have not explored that option yet[2], I doubt it could have the desired effect because the available Udmurt texts belong to a completely different domain. While most Beserman texts are narratives about the past or the life in the village, or everyday dialogues, most Udmurt texts available in digital form come from mass media. There is a pronounced difference between the vocabularies and grammatical constructions used in these two domains.

Instead, I attempt to utilize linguistic resources available for Beserman: a Beserman-Russian dictionary comprising about 6,000 entries and a morphological analyzer. The latter is rule-based and is based on the dictionary itself. Apart from the information necessary for morphological analysis, it contains some grammatical tags, such as animacy for nouns and transitivity for verbs. The analyzer recognizes about 97% of words in the textual part of the Beserman dataset. A small set of Constraint Grammar rules (Karlsson, 1990; Bick and Didriksen, 2015) is applied after the analysis, which reduces the average ambiguity to 1.25 analyses per analyzed word.

The idea is to inflate the text corpus used to produce the LM by generating a large number of fake sentences, using real corpus sentences as the starting point and the source of lemma frequencies, and incorporating data from the linguistic resources in the process.

3 Deepspeech training

All Beserman texts were encoded in a version of the Uralic Phonetic Alphabet so that each Unicode character represents one phoneme. Although there are a couple of regular phonetic processes not reflected in the transcription, such as optional final devoicing or regressive voicing of certain consonants, the characters almost always correspond to actual sounds. Therefore, CER values reported below must closely resemble PER (phone error rates)[3]. All sound files were transformed into 16 KHz, single-channel format.

Deepspeech architecture (Hannun et al., 2014) (Mozilla implementation[4]) was used for training. This involves training a 5-layer neural network with one unidirectional LSTM layer. After each epoch, the quality is checked against a development dataset not used in training. After the training is complete, the evaluation is performed on the test dataset. The train/development/test split was randomly created once and did not change during the experiments. The development dataset contains 1737 sentences; the test dataset, 267 sentences. No sound dictionary examples were included in either development or test datasets, otherwise their unnaturally high quality would lead to overly optimistic WER and CER values. It has to be pointed out though that the training dataset contains data from all speakers of the test dataset. This is in line with the primary usage scenario I had in mind, i.e. pretranscription of field data, because most untranscribed recordings in my collection are generated by the same speakers. However, for a real-world scenario where the set of potential speakers is unlimited, this setting would

[1] Available for search at `http://beserman.ru`; a large part of it has been published as Usacheva et al. (2017).

[2] There are certain phonological, morphological and lexical differences between the standard language and the Beserman dialect. Before an Udmurt model can be used in Beserman ASR, the texts should be "translated" into Beserman. Although such attempts have been made (Miller, 2017), making the translations look Beserman enough would require quite a lot of effort.

[3] This property is the reason why UPA rather than Udmurt Cyrillic script was used for encoding. Otherwise, the choice of encoding is hardly important because UPA can be converted to Cyrillics and vice versa.

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

produce an overly optimistic estimate. No transfer learning was applied.

A number of hyperparameter values were tested: learning rate between 0.00005 and 0.001, dropout rate 0.3 or 0.4, training batch size between 16 and 36. These value ranges have been demonstrated to yield optimal results on a dataset of similar size by (Agarwal and Zesch, 2019). The results did not depend in any significant way on these values, except for almost immediate overfitting when the learning rate was close to 0.001. Under all these settings, the training ran for 8 or 9 epochs, after which the loss on the development dataset started rising due to overfitting. The model used for the evaluation was trained with the following parameters: learning rate 0.0002, droupout rate 0.4, training batch size 24.

The output of the trained Deepspeech model is filtered using kenlm, an n-gram language model (Heafield, 2011). 3-gram and 4-gram models were tried, with no substantial difference; the figures below refer to the 4-gram models. When using the model with Deepspeech, there are two adjustable parameters, α and β. α (between 0 and 1) is the LM weight: higher values make the filter pay more attention to the n-gram probabilities provided by the model. β defines the penalty for having too many words: higher values increase the average number of words in transcribed sentences and decrease the average length of a word. A number of α and β combinations were tested (see below).

4 Augmented language model

As could be immediately seen from the test results, at least one of the reasons why the automatic transcription was wrong in many cases is that the corpus used to train the LM simply lacked the forms. Since Beserman is morphologically rich, a corpus of 164K words will inevitably lack most forms of most lexemes. Thankfully, this gap can be filled relatively easily, since Beserman morphological analyzer and dictionary can be turned into a morphological generator. (Another option, not explored here, would be to use subwords instead of words (Leinonen et al., 2018; Egorova and Burget, 2018).) However, if one just generated all possible word forms and added them to the corpus packed into random sentences, that would completely skew the occurrence and co-occurrence probabilities of forms, which would lead to even worse performance. The real trick would be to add the lacking forms without losing too much information from the original model, i.e. without significantly distorting the probabilities. Specifically, one would need to make the following values as close to the original ones as possible:

- relative frequencies of lemmata;
- relative frequencies of affix combinations, such as “genitive plural”;
- constraints on co-occurrence of certain grammatical forms (e.g. “verb in the first person is not expected after a second-person pronoun”);
- lexical constraints on contexts (e.g. “mother eats apples“ should be fine, while “apple eat mother“ should not).

Of course, traditional word-based text generation models strive to achieve exactly that. However, they could hardly be applied here because the objective of correctly generating a lot of previously unseen forms would be missed. Instead, I developed a sentence generator that utilizes not only the texts, but also the linguistic resources available for Beserman.

After a series of sequential improvements, the resulting sentence generator works as follows.

First, the sentences from the Beserman corpora are morphologically analyzed and turned into sentence templates. In a template, content words (nouns, verbs, adjectives, adverbs and numerals) are replaced with “slots”, while the rest (pronouns, postpositions etc.) are left untouched. The idea is that the lemma in a slot can be replaced by another lemma with similar characteristics, while the remaining words should not be replaced with anything else. Certain high-frequency or irregular verbs or adverbs are also not turned into slots, e.g. negative verbs or discourse clitics. Templates where less than one-third of the elements were turned into slots, or that contain fewer than three words, are discarded.

A slot contains the inflectional affixes the word used to have, its tags (e.g. “N,anim” for “animate noun”), as well as the original lemma.

Second, the data from the grammatical dictionary of the analyzer is processed. For each item, its lemma, stem(s) and tags (part of speech among them) is loaded. A global frequency dictionary is created. If a lemma is present in the corpora, its total number of occurrences is stored as its frequency; for the remainder of the lemmata, the frequency is set to 1.

Third, semantic similarity matrices are created for nouns, verbs and adjectives separately. The semantic similarities are induced from the Russian translations the lemmata have in the Beserman-Russian dictionary. Each translation is stripped of usage notes in brackets and parentheses and of one-letter words. After that, the first word of the remaining string is taken as the Russian equivalent of the Beserman word. The similarities between Russian translations are then calculated with an embedding model `ruwikiruscorpora_upos_skipgram_300_2_2019`[5] trained on the data of Russian National Corpus and Russian Wikipedia (Kutuzov and Kuzmenko, 2017). The resulting pairwise similarities are then condensed into a JSON file where each Beserman lemma contains its closest semantic neighbors together with the corresponding similarity value. The similarity threshold of 0.39 was set to only keep lemmata which are sufficiently similar to the lemma in question in terms of their distribution. After that, an average lemma contains about 66 semantic neighbors.

After these preparatory steps, the sentence generation starts. A template is chosen at random, after which each slot is filled with a word. If a slot contains multiple ambiguous analyses, one of them is chosen at random with equal probability, apart from several manually defined cases where one of the analyses is much more probable than the others. The original lemma of the slot is looked up in the list of semantic neighbors. If found, its semantic neighbors are used as its possible substitutes. Neighbors whose tags differ from the slot tags (e.g. inanimate nouns instead of animate) are filtered out. A random similarity threshold is chosen, which can further narrow down the list of substitutes. This way, more similar lemmata have a higher chance of ending up on the list of potential substitutes. When the list is ready, a lemma is chosen at random, with probability of each lemma proportional to its frequency in the global frequency list. Its stem is combined with the inflectional affixes in the slot, taking certain morphophonological alternations into account. The resulting word is added to the sentence. Template elements that are not slots are generally used as is, but words from a certain manually defined list can be omitted with a probability of 0.2 (this mostly includes discourse particles).

[5] https://rusvectores.org/en/models/

The sentences generated this way do not always make sense, but many of them at least are not completely ungrammatical, and some actually sound quite acceptable.

5 Results and comparison

I did not check how the size of the training dataset affects the quality of the model. However, it is interesting to note that the addition of 2.5 hours of triple headword pronunciations from the sound dictionary apparently did not add to the quality. The results were almost the same when they were omitted from the training set.

As already mentioned in Section 3, the output of a trained Deepspeech model is filtered with an n-gram model trained on a text corpus, with parameters α and β. I evaluated the model on the test dataset with three kenlm models: based only on the real Beserman sentences (`base`), and two augmented models trained on real and generated sentences (`gen`). The first augmented model was trained on 2M additional sentences (about 170K word types), the second, on 10M additional sentences (about 300K word types). The difference between the two augmented models was almost nonexistent. The larger model performed slightly better than the smaller for most parameter values, except in the case of $\alpha = 1.0$; the difference in WER in most cases did not exceed 0.5%. The figures below are given for the larger model.

The following α values were tested: 1.0, 0.9, 0.75, 0.6, 0.4. The values of β between 1.0 and 7.0 with the step of 1 were tested. The WER values for $\beta \geq 5.0$ were always worse than with lower β values and are not represented below.

One can see that the values obtained with the augmented model are better than the baseline across the board, so the sentence generation has had a positive effect on ASR quality. Also, the augmented model tolerates larger β values, whereas the baseline model starts producing too much short words in place of longer words absent from its vocabulary in that case. Nevertheless, the difference is not that large: the best `gen` value, 51.4, is lower than the best `base` value, 56.4, only by 5%. The difference in CER is even less pronounced:

A more in-depth analysis of the data reveals that the effect of LM augmentation is most visible on longer sound files. If only tested on sentences whose ground-truth transcription contained at least 6 words, the best WER value for `gen` equals 52.1, as

	$\beta = 1$	$\beta = 2$	$\beta = 3$	$\beta = 4$
$\alpha = 1.0$	57.3 / 55.6	56.5 / 54.1	57.7 / 52.7	58.8 / 52.4
$\alpha = 0.9$	57.0 / 53.3	56.8 / 52.7	58.4 / 51.6	59.6 / 53.5
$\alpha = 0.75$	56.4 / 52.9	57.2 / 51.9	58.7 / **51.4**	60.9 / 53.4
$\alpha = 0.6$	57.1 / 52.9	58.9 / 51.9	60.4 / 53.8	64.9 / 56.3
$\alpha = 0.4$	59.6 / 55.5	62.3 / 56.4	67.0 / 59.8	75.4 / 66.1

Table 1: WER for `base` (before slash) and `gen` (after slash) models with different α and β values.

	$\beta = 1$	$\beta = 2$	$\beta = 3$	$\beta = 4$
$\alpha = 1.0$	35.0 / 34.4	33.9 / 33.3	33.3 / 32.0	32.8 / 30.9
$\alpha = 0.9$	34.0 / 32.5	33.4 / 32.1	33.0 / 30.7	32.7 / 30.2
$\alpha = 0.75$	32.9 / 31.5	32.4 / 30.4	32.0 / 29.7	31.9 / **29.2**
$\alpha = 0.6$	32.2 / 30.1	31.5 / 29.9	31.5 / 29.6	31.7 / 29.3
$\alpha = 0.4$	31.6 / 30.1	31.3 / 29.6	31.3 / 30.3	31.8 / 30.8

Table 2: CER for `base` (before slash) and `gen` (after slash) models with different α and β values.

opposed to only 59.5 for `base`. On short files, however, the added benefit of having plausibly looking n-grams in the corpus stops playing any role. For sentences (or, rather, sentence fragments) that contained at most 3 words, the best WER value for `gen` equals 60.5, compared to 61.5 for `base`.

As we can see, the LM augmentation did improve the ASR quality, even if marginally. The most important takeaway from this experiment, however, was that using a bilingual dictionary and a Russian model for approximating semantic similarity was a crucial part of the LM augmentation. Without that step, the generated LM did not visibly differ from `base`, even when lemma frequencies and tags were taken into account.

Since, to the best of my knowledge, no Deepspeech (or any other) ASR models existed for standard Udmurt when the experiments were conducted, it was impossible to compare ASR quality for Beserman and standard Udmurt.

6 Conclusion

There is a famous statement by Frederick Jelinek, made exactly in the context of ASR development, "Whenever I fire a linguist our system performance improves". Indeed, contemporary ASR is largely an engineering enterprise and relies on algorithms and large amounts of data rather than on any linguistic insights. Still, if there is not enough data, can linguistic resources – resources created by linguists and for linguists – be of any help at all? The results of the experiments with the Beserman data are not conclusive. On the one hand, linguistic intervention did improve the ASR results, lowering WER by 5% and even more so in the case of longer sentences. Linguistic resources, such as the rule-based analyzer turned into a generator, and the Beserman-Russian dictionary, as well as the corpus of usage examples, seemed indispensable in the process. On the other hand, the result is yet another experimental model for a low-resource language with suboptimal performance, which might be not good enough even for auxiliary uses. In order to make it usable, one would still have to either add more data or change the algorithm (e.g. (Baevski et al., 2021) report results for comparable amounts of Tatar and Kyrgyz data that almost look like magic). It would be interesting to see if the "linguistic" LM augmentation adds anything in that case.

Acknowledgments

This research was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project ID 428175960.

References

Aashish Agarwal and Torsten Zesch. 2019. German end-to-end speech recognition based on DeepSpeech. In *KONVENS*.

Timofey Arkhangelskiy. 2019. Corpus of usage examples: What is it good for? In *Proceedings of the 3rd Workshop on the Use of Computational Methods in the Study of Endangered Languages Volume 1 (Papers)*, pages 56–63, Honolulu. Association for Computational Linguistics.

Alexei Baevski, Wei-Ning Hsu, Alexis Conneau, and Michael Auli. 2021. Unsupervised speech recognition.

Eckhard Bick and Tino Didriksen. 2015. Cg-3 – beyond classical constraint grammar. In *Proceedings of the 20th Nordic Conference of Computational Linguistics, NODALIDA*, pages 31–39. Linköping University Electronic Press, Linköpings universitet.

Ekaterina Egorova and Lukáš Burget. 2018. Out-of-vocabulary word recovery using fst-based subword unit clustering in a hybrid asr system. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5919–5923.

Awni 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. *arXiv e-prints*, page arXiv: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. Association for Computational Linguistics.

Nils Hjortnaes, Timofey Arkhangelskiy, Niko Partanen, Michael Rießler, and Francis Tyers. 2020a. Improving the language model for low-resource ASR with online text corpora. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, pages 336–341, Marseille, France. European Language Resources association.

Nils Hjortnaes, Niko Partanen, Michael Rießler, and Francis M. Tyers. 2020b. Towards a speech recognizer for Komi, an endangered and low-resource uralic language. In *Proceedings of the Sixth International Workshop on Computational Linguistics of Uralic Languages*, pages 31–37, Wien, Austria. Association for Computational Linguistics.

Fred Karlsson. 1990. Constraint grammar as a framework for parsing unrestricted text. In *Proceedings of the 13th International Conference of Computational Linguistics*, volume 3, pages 168–173, Helsinki.

Andrey Kutuzov and Elizaveta Kuzmenko. 2017. *WebVectors: A Toolkit for Building Web Interfaces for Vector Semantic Models*. Springer International Publishing, Cham.

Juho Leinonen, Peter Smit, Sami Virpioja, and Mikko Kurimo. 2018. New baseline in automatic speech recognition for Northern Sámi. In *Proceedings of the Fourth International Workshop on Computational Linguistics of Uralic Languages*, pages 87–97, Helsinki, Finland. Association for Computational Linguistics.

Eugenia Miller. 2017. Avtomaticheskoe vyravnivanie slovarej literaturnogo udmurtskogo i jazyka i besermjanskogo dialekta [Automatic alignment of literary Udmurt and Beserman dictionaries]. In *Elektronnaja pismennost narodov Rossijskoj Federacii: opyt, problemy i perspektivy [Electronic writing of the peoples of the Russian Federation: Experience, challenges and perspectives]*, Syktyvkar, Russia. KRAGSiU.

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. In *Proc. Interspeech 2019*, pages 2613–2617.

Mihaela C. Stoian, Sameer Bansal, and Sharon Goldwater. 2020. Analyzing ASR pretraining for low-resource speech-to-text translation. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7909–7913.

Zoltán Tüske, Pavel Golik, David Nolden, Ralf Schlüter, and Hermann Ney. 2014. Data augmentation, feature combination, and multilingual neural networks to improve asr and kws performance for low-resource languages. In *INTERSPEECH-2014*, pages 1420–1424.

Francis M. Tyers and Josh Meyer. 2021. What shall we do with an hour of data? Speech recognition for the un- and under-served languages of Common Voice.

Maria Usacheva, Timofey Arkhangelskiy, Olga Biryuk, Vladimir Ivanov, and Ruslan Idrisov, editors. 2017. *Тезаурус бесермянского наречия: Имена и служебные части речи (говор деревни Шамардан) [Thesaurus of the Beserman dialect: Nouns and auxiliary parts of speech (Shamardan village variety)]*. Izdatelskie resheniya, Moscow.