Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 7277–7283
Marseille, 20-25 June 2022

BembaSpeech: A Speech Recognition Corpus for the Bemba Language

Claytone Sikasote[1] **and Antonios Anastasopoulos**[2]
[1]Department of Computer Science, University of Zambia, Zambia
[2]Department of Computer Science, George Mason University, USA
claytone.sikasote@cs.unza.zm, antonis@gmu.edu

Abstract

We present a preprocessed, ready-to-use automatic speech recognition corpus, BembaSpeech, consisting over 24 hours of read speech in the Bemba language, a written but low-resourced language spoken by over 30% of the population in Zambia. To assess its usefulness for training and testing ASR systems for Bemba, we explored different approaches; supervised pre-training (training from scratch), cross-lingual transfer learning from a monolingual English pre-trained model using DeepSpeech on the portion of the dataset and fine-tuning large scale self-supervised Wav2Vec2.0 based multilingual pre-trained models on the complete BembaSpeech corpus. From our experiments, the 1 billion XLS-R parameter model gives the best results. The model achieves a word error rate (WER) of 32.91%, results demonstrating that model capacity significantly improves performance and that multilingual pre-trained models transfers cross-lingual acoustic representation better than monolingual pre-trained English model on the BembaSpeech for the Bemba ASR. Lastly, results also show that the corpus can be used for building ASR systems for Bemba language.

Keywords: Automatic Speech Recognition, ASR corpus, Low-resource language, Bemba language, Zambia

1. Introduction

Speech-to-Text, also known as Automatic Speech Recognition(ASR) or simply just Speech Recognition (SP), is the task of recognising and transcribing spoken utterances into text. In recent years, there has been a tremendous growth in popularity of speech-enabled applications. This can be attributed to their usability and integration across wide domain applications, such as voice over control systems. However, building well-performing ASR systems typically requires massive amounts of transcribed speech, as well as large text corpora. This is generally not an issue for well-resourced languages such as English and Chinese, where ASR applications have been successfully built with remarkable results (Amodei et al., 2016, et alia).

Unfortunately, this is not the case for Africa and its over 2000 languages (Heine and Nurse, 2000). The prevalence of speech recognition applications for African languages is very low. This can at least partially be attributed to the lack or unavailability of linguistic resources (speech and text) for most African languages (Martinus and Abbott, 2019). This is particularly the case with Zambian languages. There exist no general speech or textual datasets curated for building natural language processing systems, including ASR systems.

In this paper we present a speech corpus, BembaSpeech, consisting of over 24 hours of read speech in Bemba, a written but under-resourced language spoken by over 30% of the population in Zambia. We also present an end-to-end speech recognition model obtained by fine-tuning a large scale Wav2Vec2.0 based multilingual pre-trained model, XLS-R on BembaSpeech corpus. To our knowledge this is the first work carried out towards building ASR systems for any Zambian language.[1]

The rest of the paper is organized as follows. In section 2, we summarise similar works in ASR for under-resourced languages with a focus on Africa languages. In Section 3 we provide details on the Bemba language. In section 4, we outline the development process of the BembaSpeech corpus, and in section 5 we provide details of our experiments towards building a Bemba ASR model. Last, section 6 discusses our experimental results, before drawing conclusions and sketching out future research directions.

2. Related Work

In the recent past, despite the challenge of limited availability of linguistic resources, several works have been carried out to improve the prevalence of ASR applications in Africa. For example, Gauthier et al. (2016c) collected speech data and developed ASR systems for four languages: Wolof, Hausa, Swahili and Amharic. In South Africa, researchers (de Wet and Botha, 1999; Badenhorst et al., 2011; Henselmans et al., 2013; Van Heerden et al., 2016; De Wet et al., 2017) have investigated and built speech recognition systems for South African languages. Other languages that have seen development of linguistic resources for ASR applications include: Fongbe (Laleye et al., 2016) of Benin; Swahili (Gelas et al., 2012) predominantly spoken by people of East Africa; Amharic, Tigrigna, Oromo and Wolaytta of Ethiopia (Abate et al., 2005; Tachbelie and Besacier, 2014; Abate et al., 2020; Woldemariam, 2020); Hausa(Schlippe et al., 2012) of Nigeria and Somali (Abdillahi et al., 2006) of Somalia. In all the aforementioned works, Hidden Markov Models (Juang and Rabiner, 1991) and traditional statistical language

[1]The corpus is publicly released at: `https://github.com/csikasote/BembaSpeech`.

models are adopted to develop ASR systems, typically using the Kaldi (Povey et al., 2011) or HTK (Young et al., 2009) frameworks. The disadvantage of such approaches is that they typically require separate training for all their pipeline components including the acoustic model, phonetic dictionary, and language model.

Recently, end-to-end deep neural network approaches have successfully been applied to speech recognition tasks (Amodei et al., 2016; Pratap et al., 2018, et alia) achieving remarkable results outperforming traditional HMM-GMM approaches. Such methods require only a speech dataset with speech utterances and their transcriptions for training. In this work, explore different approaches to building an end-to-end Bemba ASR model on the BembaSpeech corpus; supervised pre-training (traing from scratch) and finetuning the DeepSpeech English pre-trained model using an open source end-to-end neural network system, Mozilla's DeepSpeech (Hannun et al., 2014). We also explore the strategy of finetuning the Wav2Vec2.0 (Baevski et al., 2020) based multilingual models, XLS-R (Babu et al., 2021).

3. Bemba Language

The language we focus on is Bemba (also referred to as ChiBemba, Icibemba), a Bantu language principally spoken in Zambia, in the Northern, Copperbelt, and Luapula Provinces. It is also spoken in southern parts of the Democratic Republic of Congo and Tanzania. It is estimated to be spoken by over 30% of the population of Zambia (Kula and Marten, 2008; Kapambwe, 2018).

Bemba has 5 vowels and 19 consonants (Spitulnik and Kashoki, 2001). Its syllable structure is characteristically open and is of four main types: V, CV, NCV, and NCGV (where V = vowel (long or short), C = consonant, N = nasal, G = glide (w or y))(Spitulnik and Kashoki., 2014a). The writing system is based on Latin script (Mwansa, 2017).

Similar to other Bantu languages, Bemba is described to have a very elaborate noun class system which involves pluralization patterns, agreement marking, and patterns of pronominal reference. There are 20 different classes in Bemba: 15 basic classes, 2 subclasses, and 3 locative classes (Spitulnik and Kashoki, 2001; Spitulnik and Kashoki, 2014b). Each noun class is indicated by a class prefix (typically VCV-, VC-, or V-) and the co-occurring agreement markers on adjectives, numerals and verbs.

In terms of tone, Bemba is considered to be a tone language, with two basic tones, high (H) and low (L) (Kula and Hamann, 2016). A high tone is marked with an acute accent (e.g. á) while a low tone is typically unmarked. As with most other Bantu languages, tone can be phonemic and is an important functional marker in Bemba, signaling semantic distinctions between words (Spitulnik and Kashoki, 2001; Spitulnik and Kashoki, 2014b).

4. The BembaSpeech Corpus

Description The corpus has a size of 2.8 GigaBytes with a total duration of speech data of approximately over 24 hours. We provide fixed train, development, and test splits to facilitate future experimentation. The subsets have no speaker overlap among them. Table 1 summarises the characteristics of the corpus and its subsets. All audio files are encoded in Waveform Audio File Format (WAVE) with a single track (mono) and recording with a sample rate of 16kHz.

Data collection To build the BembaSpeech corpus we used the Lig-Aikuma app (Gauthier et al., 2016c) for recording speech. Speakers used the elicitation mode of the software to record audio from text scripts tokenized at sentence level. The Lig-Aikuma has been used by other researchers for similar works (Blachon et al., 2016; Gauthier et al., 2016a; Gauthier et al., 2016b).

Speakers The speakers involved in BembaSpeech recording were students of Computer Science in the School of Natural Science at the University of Zambia. The corpus consists of 14,438 audio files recorded by 17 speakers, 9 male and 8 female. Based on the information extracted from metadata as supplied by speakers, their range of age is between 22 and 28 years and all of them identified as black. All the speakers were selected based on their fluency to speak and read Bemba and are not necessarily native language speakers. There are 14 native Bemba speakers, 1 Lozi, 1 Lunda and 1 Nsenga. It is also important to note that the recordings in this corpus were conducted outside controlled conditions. Speakers recorded as per their comfort and have varied accents. Therefore, some utterances are expected to have some background noise. We consider this "more of a feature than a bug" for our corpus: it will allow us to train and, importantly, evaluate ASR systems that match *real-world* conditions, rather than a quiet studio setting.

Preprocessing The corpus was preprocessed and validated to ensure data accuracy by eliminating all corrupted audio files and, most importantly, to ensure that all utterances matched the transcripts. All the numbers, dates and times in the text were replaced with their text equivalent according to the utterances. We also sought to follow the LibriSpeech (Panayotov et al., 2015) file organization and nomenclature by grouping all the audio files according to the speaker, using speaker ID number. In addition, we renamed all the audio files by pre-pending the speaker ID number to the utterance ID numbers.

Text Sources The phrases and sentences recorded were extracted from diverse sources in Bemba language, mainly Bemba literature. In Table 2, we summarise the sources of text contained in BembaSpeech. The length of the phrases varies from a single word to as many as 20 words.

Availability The corpus is made available to the research community licensed under the Creative Com-

Subset	Duration	Utterances	Speakers	Male	Female
Complete Corpus:					
Train	20hrs	11906	8	5	3
Dev	2hrs, 30min	1555	7	3	4
Test	2hrs	977	2	1	1
Total	24hrs, 30min	14438	17	9	8
Used with DeepSpeech:					
Train	14hrs, 20min	10200	8	5	3
Dev	2hrs	1437	7	3	4
Test	1hr, 18min	756	2	1	1
Subset total	17hrs, 38min	12393	17	9	8

Table 1: General characteristics of the BembaSpeech ASR corpus. We use a subset (audio files shorter than 10 seconds) for our baseline experiments.

ID	Source Name	Size(%)
1	Bemba literature	70
2	Other online resources	15
3	Local Radio/TV shows	10
4	Youtube movie	5

Table 2: Sources of text contained in BembaSpeech corpus. The Bemba literature includes publicly available books, magazines and training materials written in Bemba. Other online resources includes various websites with Bemba content.

		No. of Tokens	
Language Model	Sentences	Unique	Total
LM1	13461	27K	123K
LM2	403452	189K	5.8M

Table 3: The token counts for the two sets of text sources used to create the language models.

mons BY-NC-ND 4.0 license and it can be found at our github project repository.

5. Experiments

In this section, we describe the experiments to ascertain the usefulness of the speech corpus for ASR applications.[2]

5.1. DeepSpeech Model

In our experiments, we use Mozilla's DeepSpeech - an open source implementation of a variation of Baidu's first DeepSpeech paper (Hannun et al., 2014). This architecture is an end-to-end sequence-to-sequence model trained via stochastic gradient descent (Bottou, 2012) with the Connectionist Temporal Classification (Graves et al., 2006, CTC) loss function. The model is six layers deep: three fully connected layers connected followed by a unidirectional LSTM (Hochreiter and Schmidhuber, 1997) layer followed by two more fully connected layers. All hidden layers have a dimensionality of 2048 and a clipped ReLU (Nair and Hinton, 2010) activation. The output layer has as many dimensions as characters in the alphabet of the target language (including desired punctuations and blank symbols used for CTC). The input layer accepts a vector of 19 spliced frames (9 past frames, 1 present frame and 9 future frames) with 26 MFCC features each. We use the DeepSpeech v0.8.2[3] release for all our experiments.

5.2. Data preprocessing

We preprocessed the data in conformity with the expectation of the DeepSpeech input pipeline. We converted all transcriptions to lower case. Since DeepSpeech only accepts audio files not exceeding 10 seconds, we considered only audio files with that duration for our training. This resized the corpus for training as can be seen in Table 1. We also generated an alphabet of characters and symbols which appear in the text, the length of which determines the size of the output layer of the DeepSpeech model. We note that, since Bemba uses the Latin alphabet, our alphabet was the same as that of the pre-trained DeepSpeech English model.

5.3. Training a Bemba Model from Scratch

Similar to (Hjortnaes et al., 2020; Meyer, 2020), we trained DeepSpeech from scratch using the default parameters[4] on the BembaSpeech dataset, providing a baseline model for our experiments.

[2]Code to reproduce experiments with DeepSpeech is available here: `https://github.com/csikasote/bembaspeech-exps`.

[3]`https://github.com/mozilla/DeepSpeech/tree/v0.8.2`

[4]With the exception of batch size: instead of using the default batch size of 1 for train, dev and test, we used 64, 32, 32 respectively for all our experiments.

Model	WER(%)	CER%
Baseline	100.00	85.67
DeepSpeech	71.21	16.68
DeepSpeech + LM1-5	**54.78**	**17.05**
DeepSpeech + LM2-3	55.65	19.69

Table 4: Results of experiments using DeepSpeech. Fine-tuning the DeepSpeech English pretrained model we obtained 71.21% WER before integrating the 5-gram language model and 54.78% afterwards.

5.4. Transfer Learning from English to Bemba

In our search for a better performing model, we applied and also experimented with cross-lingual transfer learning. We achieve this by fine-tuning a well performing DeepSpeech English pre-trained model on our Bemba dataset, using a learning rate of 0.00005, dropout at 0.4, and 50 training epochs with early stopping. We report the WER results in Table 4. By finetuning a DeepSpeech English pretrained model on the BembaSpeech we were able to obtain 28% relative WER improvement over the baseline model.

5.5. Finetuning Multilingual Pretrained Models

In addition to training from scratch and transfer learning using the DeepSpeech model, we explored the strategy of finetuning multilingual pretrained models [5] on the complete BembaSpeech dataset. In our experiment, we considered the wav2vec2.0 (Baevski et al., 2020) based, XLS-R (Babu et al., 2021). The XLS-R (Babu et al., 2021) models are trained on approximately half a million hours of publicly available speech audio data in 128 languages. In our experiments, we explored two models, the 300 million and 1 billion parameter models herein referred to as XLS-R-0.3B and XLS-R-1B models respectively[6]

We use the HuggingFace Transformer library (Wolf et al., 2020) to finetune both models on complete BembaSpeech dataset using Connectionist Temporal Classification (Graves et al., 2006, CTC). With exception of the batch sizes, learning rate and mask probability, most of the configurations are the default ones inherited from the library.[7] All the models are finetuned on a single NVIDIA Tesla P100 GPU using the Google Colab+ platform. We utilize gradient checkpoint and gradient accumulation to optimize GPU memory usage and utilization. In addition, we use the bitsandbytes (Dettmers et al., 2021) to replace the native-memory intensive Adam (Kingma and Ba, 2015) optimizer during training.

Model	WER%	CER%
XLS-R-0.3B	42.23	8.65
XLS-R-1B	40.23	7.61
+LM1-5		
XLS-R-0.3B	37.34	8.45
XLS-R-1B	**32.91**	**6.77**

Table 5: Results obtained from finetuning the wav2vec2.0 based large-scale multi-lingual pretrained models; XLS-R-0.3B and XLS-R-1B on the complete BembaSpeech dataset. The 5-gram LM boosted XLS-R with 1B paramater model gave the best performance with 32.91% WER.

The results of these experiments are presented in Table 5, both without and with incorporating a Bemba language model. Using the pre-trained model leads to significant improvements over our previous best results, leading to a 40% reduction in WER (from 54.8 to 32.9) and a 60% reduction for CER (from 17.1 to 6.8), when using the language model. Improvements even without the language model are still significant, in the range of 26% reduction in WER. We note again that these results are without any serious hyperparameter tuning, which implies that there is room for further improvement, providing a viable path towards usable speech recognition in Bemba.

5.6. Impact of Language Model

We also investigated the impact of including a language model to the acoustic models to improve performance. We use the N-gram language model for our experiments. In order to identify the language model that give the most improved model performance, we explored two sets of language models each consisting, 3-gram, 4-gram and 5-gram. The first set of language models, denoted LM1, were generated from text sourced from train and development transcripts. The second set, denoted LM2, were sourced from a combination of text from train and development transcripts and additional Bemba text from the JW300 dataset (Agic and Vulic, 2020). In Table 3 we give the token count for LM1 and LM2. All the language models were generated using the KenLM (Heafield, 2011) language model library.

For the DeepSpeech model, we used the DeepSpeech native library to create the trie based models with default parameter values. The same speech recognition model obtained from section 5.4 was used changing only the language model. For the XLS-R models, we use only the 5-gram language model from LM1 set. We used a python-based CTC beam search decoder

[5] HuggingFace: https://huggingface.co/models?other=xls_r

[6] The fine-tuned XLS-R Bemba models are publicly available at https://huggingface.co/csikasote.

[7] Models are trained for 30 epochs with early-stopping. The XLS-R-0.3B is finetuned with learning rate of 3e-4 and batch size of 8 for training and evaluation whereas the XLS-R-1B model is finetuned using the learning rate of 5e-5 with batch size of 4 and 8 for training and evaluation respectively

Model	Training Data Splits		
	15hrs	10hrs	5hrs
- LM			
XLS-R-0.3B	45.41	47.35	47.33
XLS-R-1B	35.21	36.60	51.60
+ LM			
XLS-R-0.3B	39.39	40.05	40.54
XLS-R-1B	39.33	34.96	43.36

Table 6: Results of the ablation study on training data size of the BembaSpeech dataset. We report BembaSpeech ASR results in terms of WER. Models are fine-tuned with 15hrs, 10hrs, 5hrs of training data splits. We compare the performance of XLS-R-0.3B to XLS-R-1B with and without language model integration. Cross-lingual training with higher capacity such as for XLS-R (1B) obtains competitive performance.

library, `pyctcdecode`[8] to connect the external language model. The best performing model is a 5-gram LM boosted 1B parameter model with the word error rate of 32.91.

5.7. Effect of Training Data Size

Similar to (Krishna et al., 2021) we investigated the impact of training data size on model performance. To achieve this, we randomly subsampled and split the training data of the BembaSpeech into three separate sets; 15hrs, 10hrs and 5hrs. The splits do not overlap. For this experiment, we finetune the XSL-R-0.3B and XLS-R-1B models. With exception of the number of epochs, we use the same training configuration described in section 5.5 for each model. All models are trained for 10 epochs. We use the 5-gram language model from LM1 set to boost model performance. We report the WER result in Table 6. Surprisingly, for the 15hrs split, the model suffers a slight performance degradation after adding the LM. We will investigate this peculiar behaviour in future work.

6. Results and Discussion

Table 7 summarises the results obtained from our experiments. The best performing model was XLS-R-1B + LM1-5, obtained from fine-tuning a wav2vec2.0-based 1 billion parameter XLS-R model with a 5-gram language model generated from text sourced from transcripts. The model achieves a word error rate (WER) of 32.91%. We note also that XLS-R-1B model consistently outperforming both the XLS-R-0.3B and DeepSpeech models before and after inclusion of a language model. Before integrating the language model, the XLS-R-1B model improves by 5% and 44% relative WER over the XLS-R-0.3B and DeepSpeech models respectively. Similarly,

[8] `https://github.com/kensho-technologies/pyctcdecode`

Model	WER%	CER%
- LM		
Baseline	100.00	85.67
DeepSpeech	71.21	16.68
XLS-R-0.3B	42.23	8.65
XLS-R-1B	40.23	7.61
+ LM1-5		
DeepSpeech	**54.78**	**17.05**
XLS-R-0.3B	37.34	8.45
XLS-R-1B	**32.91**	**6.77**

Table 7: Summary results of the experiments. We report only the best WER result for each model considered.

after language model integration, XLS-R-1B improves performance by 12% and 40% relative WER over the XLS-R-0.3B and DeepSpeech models respectively. This shows that model capacity significantly improves performance of a model because of possessing much richer cross-lingual acoustic representations. The results also show that both XLS-R-1B and XLS-R-0.3B outperform the DeepSpeech model both before and after integration of language model. This demonstrates the effectiveness of multilingual pre-trained models in cross-lingual transfers over monolingual pre-trained models. The improvement in performance by the DeepSeech model over the baseline shows that cross-lingual transfer ability of monolingually pre-trained models over supervised pre-training (training from scratch), in this case from English to Bemba.

The results also show that language model integration consistently gives a significant improvement to the performance of the models in recognising the Bemba language. By including the language model we were able to improve the model performance by 15%, 9% and 23% relative WER for XLS-R-1B, XLS-R-0.3B and DeepSpeech model. Interestingly, no significant change in performance was observed by the inclusion of the additional 389,991 sentences from the JW300 Bemba data.

7. Conclusion

In this paper, we presented an ASR corpus for the Bemba language of Zambia, BembaSpeech. We also explored different strategies of building an End-to-End Bemba ASR model; using supervised learning (from scratch), cross-lingual transfer learning using monolingual pretrained English model and fine-tuning multilingual pre-trained models. In our experiments, we used DeepSpeech for supervised learning and cross-lingual transfer learning from the mono-lingually English pre-trained model. For the multilingual pre-trained model, we used the Wav2vec2.0 based large scale, XLSR models. We found that fine-tuning a 1 billion parameter XLS-R model gave the best performance of 32.91% WER outperforming other models on the BembaSpeech.

For the future, there are many things we can do to improve the results of the model. Firstly, we are interested to carry in-depth error analysis to understand the typical errors being made by our model similar to the works of (Salimbajevs and Strigins, 2015, et alia). We are interested to investigate why additional Bemba text from JW300 did not make any significant change in model performance for the DeepSpeech model. We are also interested to investigate the impact of using the Transformer based language model instead of the N-gram language model we used in our experiments. In addition, we are also interested in further improving our corpus both in size and number of speakers involved. Lastly, we plan to (a) collect even more data in Bemba, (b) collect data in the different Bemba varieties as spoken throughout Zambia, as well as (c) other Zambian languages.

8. Acknowledgments

We express our gratitude to all the speakers who were involved in the creation of our corpus. Antonios Anastasopoulos is generously funded by the U.S. National Science Foundation under grants IIS-2125466 and BCS-2109578 and by a Google award for Inclusion Research.

9. Bibliographical References

Abate, S. T., Menzel, W., and Tafila, B. (2005). An amharic speech corpus for large vocabulary continuous speech recognition. In *9th European Conference on Speech Communication and Technology*.

Abate, S. T., Tachbelie, M. Y., Melese, M., Abera, H., Abebe, T., Mulugeta, W., Assabie, Y., Meshesha, M., Atinafu, S., and Ephrem, B. (2020). Large vocabulary read speech corpora for four ethiopian languages: Amharic, Tigrigna, Oromo and Wolaytta. In *LREC 2020 - 12th International Conference on Language Resources and Evaluation, Conference Proceedings*.

Abdillahi, N., Pascal, N., and Jean-François, B. (2006). Towards automatic transcription of Somali language. In *Proceedings of the 5th International Conference on Language Resources and Evaluation, LREC 2006*.

Agic, Ž. and Vulic, I. (2020). JW300: A wide-coverage parallel corpus for low-resource languages. In *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*.

Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G., Chen, J., Chen, J., Chen, Z., Chrzanowski, M., Coates, A., Diamos, G., Ding, K., Du, N., Elsen, E., Engel, J., Fang, W., Fan, L., Fougner, C., Gao, L., Gong, C., Hannun, A. N., Han, T., Johannes, L. V., Jiang, B., Ju, C., Jun, B., Legresley, P., Lin, L., Liu, J., Liu, Y., Li, W., Li, X., Ma, D., Narang, S., Ng, A., Ozair, S., Peng, Y., Prenger, R., Qian, S., Quan, Z., Raiman, J., Rao, V., Satheesh, S., Seetapun, D., Sengupta, S., Srinet, K., Sriram, A., Tang, H., Tang, L., Wang, C., Wang, J., Wang, K., Wang, Y., Wang, Z., Wang, Z., Wu, S., Wei, L., Xiao, B., Xie, W., Xie, Y., Yogatama, D., Yuan, B., Zhan, J., and Zhu, Z. (2016). Deep speech 2: End-to-end speech recognition in English and Mandarin. In *33rd International Conference on Machine Learning, ICML 2016*.

Babu, A., Wang, C., Tjandra, A., Lakhotia, K., Xu, Q., Goyal, N., Singh, K., von Platen, P., Saraf, Y., Pino, J., Baevski, A., Conneau, A., and Auli, M. (2021). Xls-r: Self-supervised cross-lingual speech representation learning at scale. *arXiv*, abs/2111.09296.

Badenhorst, J., van Heerden, C., Davel, M., and Barnard, E. (2011). Collecting and evaluating speech recognition corpora for 11 South African languages. *Language Resources and Evaluation*.

Baevski, A., Zhou, H., Mohamed, A., and Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. volume 2020-December.

Blachon, D., Gauthier, E., Besacier, L., Kouarata, G. N., Adda-Decker, M., and Rialland, A. (2016). Parallel Speech Collection for Under-resourced Language Studies Using the Lig-Aikuma Mobile Device App. In *Procedia Computer Science*.

Bottou, L. (2012). Stochastic Gradient Descent Tricks.

de Wet, F. and Botha, E. C. (1999). Towards speech technology for south african languages: Automatic speech recognition in xhosa. *South African Journal of African Languages*.

De Wet, F., Kleynhans, N., Van Compernolle, D., and Sahraeian, R. (2017). Speech recognition for under-resourced languages: Data sharing in hidden Markov model systems. *South African Journal of Science*.

Dettmers, T., Lewis, M., Shleifer, S., and Zettlemoyer, L. (2021). 8-bit optimizers via block-wise quantization.

Gauthier, E., Besacier, L., and Voisin, S. (2016a). Automatic Speech Recognition for African Languages with Vowel Length Contrast. In *Procedia Computer Science*.

Gauthier, E., Besacier, L., Voisin, S., Melese, M., and Elingui, U. P. (2016b). Collecting resources in sub-Saharan African languages for automatic speech recognition: A case study of Wolof. In *Proceedings of the 10th International Conference on Language Resources and Evaluation, LREC 2016*.

Gauthier, E., Blachon, D., Besacier, L., Kouarata, G. N., Adda-Decker, M., Rialland, A., Adda, G., and Bachman, G. (2016c). LIG-AIKUMA: A mobile app to collect parallel speech for under-resourced language studies. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*.

Gelas, H., Besacier, L., and Pellegrino, F. (2012). Developments of Swahili resources for an automatic speech recognition system. *SLTU-Workshop on Spoken Language Technologies for Under-Resourced Languages*.

Graves, A., Fernández, S., Gomez, F., and Schmidhu-

ber, J. (2006). Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *ACM International Conference Proceeding Series*.

Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., and Ng, A. Y. (2014). Deep Speech: Scaling up end-to-end speech recognition.

Heafield, K. (2011). KenLM : Faster and Smaller Language Model Queries. *Proceedings of the Sixth Workshop on Statistical Machine Translation*.

Heine, B. and Nurse, D. (2000). *African Languages: An Introduction*. Cambridge University Press.

Henselmans, D., Niesler, T., and Leeuwen, D. V. (2013). Baseline Speech Recognition of South African Languages using Lwazi and AST. In *Proceedings of the twenty-fourth annual symposium of the Pattern Recognition Association of South Africa (PRASA)*.

Hjortnaes, N., Arkhangelskiy, T., Partanen, N., Rießler, M., and Tyers, F. (2020). 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)*.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*.

Juang, B. H. and Rabiner, L. R. (1991). Hidden markov models for speech recognition. *Technometrics*.

Kapambwe, M. (2018). An Introduction to Zambia's Bemba Tribe.

Kingma, D. P. and Ba, J. L. (2015). Adam: A method for stochastic optimization.

Krishna, D. N., Wang, P., and Bozza, B. (2021). Using large self-supervised models for low-resource speech recognition. volume 1.

Kula, N. C. and Hamann, S. (2016). *Intonation in Bemba*.

Kula, N. C. and Marten, L. (2008). *Language and National Identity in Africa*. Oxford University.

Laleye, F. A., Besacier, L., Ezin, E. C., and Motamed, C. (2016). First automatic fongbe continuous speech recognition system: Development of acoustic models and language models. In *Proceedings of the 2016 Federated Conference on Computer Science and Information Systems, FedCSIS 2016*.

Martinus, L. and Abbott, J. Z. (2019). Benchmarking Neural Machine Translation for Southern African Languages.

Meyer, J. (2020). *Multi-Task and Transfer Learning in Low-Resource Speech Recognition*. Ph.D. thesis, The University of Arizona.

Mwansa, J. M. (2017). Theorectical Reflections on the Teaching of Literacy in Zambian Bantu Languages. *International Journal of Humanities, Social Sciences and Education*, 4(10):116–129.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve Restricted Boltzmann machines. In *ICML 2010 - Proceedings, 27th International Conference on Machine Learning*.

Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. (2015). Librispeech: An ASR corpus based on public domain audio books. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlıcek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., and Vesely, K. (2011). The Kaldi speech recognition toolkit. In *Proc. ASRU*.

Pratap, V., Hannun, A., Xu, Q., Cai, J., Kahn, J., Synnaeve, G., Liptchinsky, V., and Collobert, R. (2018). WAV2LETTER++: THE FASTEST OPEN-SOURCE SPEECH RECOGNITION SYSTEM.

Salimbajevs, A. and Strigins, J. (2015). Error analysis and improving speech recognition for latvian language. volume 2015-January.

Schlippe, T., Djomgang, E. G. K., Vu, N. T., Ochs, S., and Schultz, T. (2012). Hausa large vocabulary continuous speech recognition. In *Proceedings of the Workshop on Spoken Languages Technologies for Under-Resourced Languages (SLTU)*.

Spitulnik, D. and Kashoki, M. E. (2001). *Facts About the World's Languages: An Encyclopedia of the Worlds's Major Languages, Past and Present*. H.W. Wilson, New York.

Spitulnik, V. D. and Kashoki., M. E. (2014a). Bemba Morphology.

Spitulnik, V. D. and Kashoki, M. E. (2014b). Bemba Phonology.

Tachbelie, M. Y. and Besacier, L. (2014). Using different acoustic, lexical and language modeling units for ASR of an under-resourced language - Amharic. *Speech Communication*.

Van Heerden, C., Kleynhans, N., and Davel, M. (2016). Improving the Lwazi ASR baseline. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*.

Woldemariam, Y. (2020). Transfer Learning for Less-Resourced {S}emitic Languages Speech Recognition: the Case of {A}mharic. 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)*.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. (2020). Transformers: State-of-the-Art Natural Language Processing. pages 38–45. Association for Computational Linguistics, 10.

Young, S. J., Evermann, G., Gales, M. J. F., Hain, T., Kershaw, D., Liu, X., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., and Woodland, P. C. (2009). The HTK Book (for HTK Version 3.4). *Construction*.