Proceedings of the 1st Joint SLTU and CCURL Workshop (SLTU-CCURL 2020), pages 362–367
Language Resources and Evaluation Conference (LREC 2020), Marseille, 11–16 May 2020

Speech Transcription Challenges for Resource Constrained Indigenous Language Cree

Vishwa Gupta, Gilles Boulianne
Centre de recherche informatique de Montréal (CRIM)
{vishwa.gupta, gilles.boulianne}@crim.ca

Abstract

Cree is one of the most spoken Indigenous languages in Canada. From a speech recognition perspective, it is a low-resource language, since very little data is available for either acoustic or language modeling. This has prevented development of speech technology that could help revitalize the language. We describe our experiments with available Cree data to improve automatic transcription both in speaker- independent and dependent scenarios. While it was difficult to get low speaker-independent word error rates with only six speakers, we were able to get low word and phoneme error rates in the speaker-dependent scenario. We compare our phoneme recognition with two state-of-the-art open-source phoneme recognition toolkits, which use end-to-end training and sequence-to-sequence modeling. Our phoneme error rate (8.7%) is significantly lower than that achieved by the best of these systems (15.1%). With these systems and varying amounts of transcribed and text data, we show that pre-training on other languages is important for speaker-independent recognition, and even small amounts of additional text-only documents are useful. These results can guide practical language documentation work, when deciding how much transcribed and text data is needed to achieve useful phoneme accuracies.

Keywords: Speech recognition/understanding, Endangered Languages, Multimedia Document Processing

1. Introduction

As part of a governmental effort to preserve, revitalize, and promote the more than 70 Indigenous languages being spoken in Canada, we are exploring speech recognition technology for its potential to help transcription in language documentation and to facilitate access to recorded archives. Developing speech models for Indigenous languages is also important for applications valued by communities, such as creation of learning materials with synchronized text and speech, or dictionaries searchable by voice. We report here our work on Cree, one of the most spoken Indigenous languages in Canada, more specifically East Cree, spoken along James Bay coast. We hope to apply similar algorithm to help create speech recognition systems for other Indigenous languages.

From a speech recognition perspective, there is very little audio available in East Cree and even smaller amounts of audio are transcribed. Only a small amount of text (related mostly to the Bible) in Cree can be found for language modeling. We were able to get about 4.5 hours of transcribed audio from the internet: 9 videos from 5 speakers recalling stories, for a total of 1 hour (5939 words), and recordings related to readings of the Bible from one female speaker (3.5 hours, 21205 words). We also have 1247 CBC radio broadcasts in Cree which are not transcribed. Each CBC broadcast is about 1 hour long. We also had access to annual reports and Bible texts (differing from the read Bible audios) for training the language model. We outline here the experiments we have run in order to get good transcription accuracy for Cree with the available data described above.

The low resource transcription and keyword spotting effort received a great impetus from the IARPA Babel program[1]. In this program there were 10 languages with 10 hours (limited language pack or LLP) or 80 hours (full language pack or FLP) of transcribed audio. Many different DNN training algorithms have been experimented within the Babel program (Gales et al., 2014) (Knill et al., 2014) (Huang et al., 2013) (Trmal et al., 2014) (Chen et al., 2013) (Zhang et al., 2014). In (Gales et al., 2014) they experiment with both DNN and tandem systems and achieve token error rates (TER) between 60% and 77% with LLP, depending on the language and training algorithms. They also experiment with data augmentation by automatically labeling untranscribed data. In (Gales et al., 2014) and (Knill et al., 2014), they experiment with zero resource acoustic models where the acoustic models are trained using multiple languages to get good representation of phonemes. However, the language-independent TER is poor (over 80% TER). Another data augmentation method is to jointly train DNNs from multiple languages (Huang et al., 2013) (Trmal et al., 2014). Only the output layer is trained separately for each language.

All the languages in the Babel program are spoken by millions of people. Only the data to train the acoustic and language models was restricted. In contrast, the number of speakers speaking East Cree is estimated to be around 10,000 and very little transcribed audio data is available. The audio data that is available is spoken by only a few speakers. That is why we could only get 4.5 hours of transcribed audio from 6 speakers. For language modeling, written text in Cree was hard to get. There are no large volume publications in Cree. Most of the text is on Bible scriptures or annual community or town reports. Even in these reports, text in Cree is only a very small part of the report. We managed to extract about 260,000 words of text in Cree (written in syllabics) from all sources.

Speaker-dependent word or phoneme error rate (WER or PER) becomes an important issue for resource poor languages (Adams et al., 2018). Generally, we can find audio from a very few speakers, so getting good speaker-independent WER becomes difficult. The question is

[1]https://www.iarpa.gov/index.php/research-programs/babel

whether we can transcribe part of the audio from one speaker, train acoustic models, and then recognize the rest with reasonable accuracy. In this case, correcting a transcript with some errors might be much faster than transcribing from scratch.

This was shown in previous work on phoneme recognition for resource-poor languages (Adams et al., 2018). Time-aligned phoneme transcripts can be used to transcribe the audio in a low resource language more efficiently, reducing time and effort. The cited work used Persephone, an open source toolkit for phoneme recognition. We compare our system with Persephone and show that we get significantly lower phoneme error rates both with traditional DNN systems and also with sequence-to-sequence training.

We show here the word error rate and phoneme error rate we can achieve with just 4.5 hours of transcribed audio and 260k words of text for language modeling. We measure both speaker-dependent and speaker-independent WER. With the most advanced algorithms, we can achieve 24.6% WER for speaker-dependent recognition, and 70% WER for speaker-independent recognition. The 24.6% WER is probably good enough to significantly reduce transcription time for one speaker of Cree, for example, for the anchor of the CBC broadcasts in Cree.

2. Acoustic and Language Model Data

The complete data set for training acoustic and language models is outlined in Table 1.

data	amount	source
transcribed video stories	1 hour	beesum
read biblical text	3.5 hours	biblical website
CBC radio broadcasts	1343 hours	CBC radio archives
biblical text	160,000 words	biblical website
text from annual reports	110,000 words	annual reports of Cree organisations

Table 1: Amount of audio and text data in Cree from various sources. The CBC radio broadcasts in Cree are not transcribed. There are 9 transcribed video stories from 5 speakers. The read biblical text is from 1 female speaker.

One male and one female speaker from video data were used for testing (a total of 17 minutes of audio), and the remaining 3 video speakers were used in the training set (a total of 43.6 minutes of audio). The audio from one female speaker reading scriptures was divided into test (17.5 mins) and training (3.1 hours) sets. In order to increase the size of the training set for acoustic modeling, the training audio was speed perturbed, with factors of 0.9 and 1.1 times, before training the deep neural net (DNN) acoustic models.

The 4-gram language model was trained on a mix of annual reports and Bible texts downloaded from the internet and different from the read Bible audios described above. The texts were split into 253,245 words for training and 6,737 words for development. Only the words in the LM training set were used as the 4-gram LM vocabulary: a total of 27189 words.

The perplexity of the language model (LM) represents how well the LM represents the word sequences in the text. The lower the perplexity, the better the language model. The 4-gram LM has 82.9 perplexity on the LM dev set, 317 on video transcripts, and 159.7 on Bible transcripts. The weighted out-of- vocabulary (OOV) rate is 7% on LM dev set, 24.9% on video transcripts, and 9.1% on Bible transcripts. So we expect much lower WER on bible transcripts (lower OOV rate and lower perplexity) than on video transcripts using this LM.

We use the same pronunciation dictionary for acoustic model training and decoding: it contains all the words in the LM training set (27189 words) plus the words in the video transcripts for a total of 29598 words. However, during decoding, words not in the language model will be considered as out-of-vocabulary (OOV). All the texts use the Unified Canadian Aboriginal Syllabics character set[2], and all the words are transcribed in X-SAMPA[3] phoneme set, by directly mapping each syllabic character to its phonemic representation.

3. Experiments with Transcribed Data

Since the amount of training data for Cree is very small, we decided to run two separate experiments. In one experiment we train deep neural net (DNN) acoustic models from the training data for Cree only. In the second experiment, we train models from about 4000 hours of transcribed English, and then adapt the resulting models to the Cree training data. Currently, the state-of-the-art DNN models are lattice-free maximum mutual information (LF-MMI) trained factored time delay neural networks (TDNN-F) (Povey et al., 2018)(Povey et al., 2016) and bidirectional long short memory neural networks (BLSTM) (Graves et al., 2013) models. So in the two experiments, we train both BLSTM models and TDNN-F models. In both these experiments, we use the same i-vector extractor trained from a large English dataset with many speakers. I-vectors represent speaker characteristics and adding these i-vector features to the standard mel frequency cepstral coefficients (MFCC) results in significant reduction in word error rates (WER).

For the first experiment, we trained both the GMM/HMM and DNN models from just the Cree training data. We used 40-dimensional MFCC features, and 100 dimensional i-vectors (Gupta et al., 2014) (Saon et al., 2013) (Senior and Lopez-Moreno, 2014) to represent speaker characteristics. For the small TDNN-F models (768-dimensional system with a linear bottleneck dimension of 160), LF-MMI training was followed by two iterations of discriminative training with sMBR (scalable Minimum Bayes Risk) criteria. In each iteration, the alignment between the audio and the audio transcript was created from the previous models followed by 3 epochs of sMBR discriminative training. For the BLSTM models, we did two iterations of back propagation training. In the first iteration, the models were trained

[2]https://unicode.org/charts/nameslist/c_1400.html

[3]https://fr.wikipedia.org/wiki/X-SAMPA

by back propagation using 6 epochs of training. In the second iteration models from the first iteration were used for alignment and as the initial models. In second iteration also we did 6 epochs of training.

The various results are shown in Table 2, which shows that BLSTM models give lower WER (74.3%) for video (speaker-independent) test set than either GMM-HMM or TDNN-F models. The TDNN-F models give the lowest WER (25.9%) for the scriptures test set (speaker-dependent). So three hours of audio is enough to give *useful* speaker-dependent results. By *useful* we mean that new audio from the same speaker can be transcribed and time-aligned with these models, and the resulting transcription can be corrected in significantly less time than manually transcribing the whole audio from scratch. Such transcription of large amounts of archived audio from a single speaker at a reasonable cost is of interest for many indigenous languages.

Model architecture	**WER video**	**WER scriptures**
GMM-HMM	77.1%	35.0%
BLSTM 1st BP	75.1%	27.9%
(1) BLSTM 2nd BP	74.3%	27.1%
TDNN-F LF-MMI	96.0%	27.0%
TDNN-F LF-MMI + sMBR	89.7%	26.0%
(2) + 2nd sMBR	90.1%	25.9%

Table 2: Word error rates for video (speaker-independent) and scriptures (speaker-dependent) test sets with a total of 3.85 hours of Cree training data. BP refers to back propagation, while sMBR is discriminative training with sMBR criterion.

In the second scenario, we train initial models from a very large dataset, and then adapt these models to the Cree training set. The idea is to have a model with well-trained phoneme set, and then to adapt these phonemes to Cree with the small training set for Cree. Phonemes that occur in the Cree training audio will get trained by Cree data, while other phonemes will still have a somewhat decent representation in these models. For the very large dataset, we used about 4000 hours of transcribed English audio available to us. This audio included Hub4, RT03, RT04, Market, WSJ, Librispeech, Switchboard, and Fisher data. This data is available from LDC. We added Inuktitut data available to us from the same Indigenous languages project, in an effort to cover some of the Cree phonemes missing from English. We used the same X-SAMPA phoneme set for this scenario also.

We then adapt both BLSTM and TDNN-F models trained above to the Cree training set. For training the larger TDNN-F (1536-dimensional system with a linear bottleneck dimension of 160) acoustic models, we ran multiple iterations of discriminative training with sMBR criteria on Cree data starting with the above models. So in the first iteration, the alignments for the Cree data come from the TDNN-F models trained from the 4000 hours of audio, and is followed by three epochs of discriminative training with sMBR criteria using the Cree data only. In the subsequent iterations, the alignments between Cree training audio and its transcript come from the acoustic models generated in the previous iteration and is followed by three epochs of discriminative training with sMBR criteria using the Cree data only. We ran three such iterations of discriminative training using Cree data only.

For BLSTM adaptation, in the first iteration, we do back propagation starting with alignments from the BLSTM models trained from 4000 hours of audio, followed by 6 epochs of back propagation training. The initial models for this back propagation training are the BLSTM models trained from the 4000 hours of audio. In the subsequent iterations of back propagation, the alignments on Cree training data come from the models generated in the previous iteration and is followed by 6 epochs of back propagation training. In this back propagation training also, the initial BLSTM acoustic models come from the previous iteration. Overall we do four such iterations. The results with this adaptation strategy are shown in Table 3. When we compare Table 2 with Table 3, we see that BLSTM models trained from only Cree training data (first scenario) is only slightly worse than BLSTM acoustic model trained through adaptation from a model trained with large amount of data (74.3% versus 74.1%). However, for TDNN-F models, the speaker-independent WER is much lower with adapted TDNN-F (89.7% versus 78.8%). The speaker-dependent WER is much better when trained using the first scenario for both TDNN-F and BLSTM models. We also tried 40-dimensional filter-bank features instead of 40-dimensional MFCC features, but the WER is a little bit worse. However, since each model is different, we can combine the outputs using ROVER (Fiscus, 1997) to get significantly lower WER for both speaker-independent and speaker-dependent recognition. ROVER combines multiple transcripts obtained from multiple recognizers by first aligning the transcripts, and then taking a majority vote. The last line in Table 3 shows WER after ROVER.

Model specification	**WER video**	**WER scriptures**
(3) BLSTM MFCC 4th BP	74.1%	29.5%
(4) BLSTM fbank 4th BP	79.0%	31.2%
(5) TDNN-F 3rd sMBR	78.8%	30.0%
(8) ROVER of 1,2,3,4,5	72.5%	25.1%

Table 3: Word error rates for video and scriptures test sets after adaptation to a total of 3.85 hours of Cree training data. BP refers to back propagation, while sMBR is discriminative training with sMBR criterion.

4. Experiments with Untranscribed Cree

In the MGB-3 challenge, we used closed-captioned audio files for training the acoustic models (Gupta and Boulianne, 2018). JHU also used closed-captioned audio for training the DNNs for Arabic audio (Manohar et al., 2017). The closed-captioned transcripts are not verbatim transcription of the audio. Using the transcripts from closed-captioning as is for training acoustic models will lead to poor acoustic models. So the transcript from the closed-captioning

is purified by generating another transcript through recognition, then comparing the two transcripts, and only using segments of the transcripts that match well. This results in acoustic models that give much lower WER. For Cree audio, we do not have a transcript from closed-captioning. So we can fake closed captioning by combining (using ROVER) decoded transcripts from many recognizers as the closed-captioned transcript. This transcript is different from the recognized transcript of the best recognizer, so we can use segments of the transcripts that match well for training new acoustic models.

As a first experiment, we used ROVER of 3 transcripts of 1247 Cree audio files (two best TDNN-F models and the LSTM fbank model). The Cree audio files from CBC radio broadcasts have music, singing, etc. So these portions were removed first by using a DNN-based voice activity detector (Alam et al., 2019). The remaining segments were decoded using the above three acoustic models. The transcripts were then ROVERed to get a final transcript. This transcript was then used as a closed-captioned transcript to find matching segments using the best TDNN-F acoustic model adapted to the Cree audio (item (5) in Table 3). Out of 1247 hours of audio, this process reduced the audio to 221 hours. The 3.85 hours of Cree training audio was then added to these segments. This 224 hours of transcribed audio was then used to adapt the best TDNN-F models to this data by another iteration of discriminative training. The results are shown in Table 4. We have reduced WER for the video test set from 78.8% to 75.1%, and for the scriptures test set from 30.0% to 28.7%. Training BLSTM models with this 224 hours of audio gave 72.5% (video) and 27.6% (scriptures) WER. Note that the WER differences between the TDNN-F and BLSTM models are much smaller after training with the 224 hours of audio. Combining with ROVER the ctm files from all the 7 models (item (9) in Table 4) results in 69.9% WER for the video test set (speaker-independent) and 24.6% WER for the scripture test set (speaker-dependent).

Training	video	scriptures
(6) BLSTM with 224 hours	72.5%	27.6%
(7) TDNN-F with 224 hours	75.1%	28.7%
(9) ROVER 1 thru 7	69.9%	24.6%

Table 4: Word error rates for the video and scriptures test sets after training with untranscribed Cree radio broadcasts.

5. Phoneme Recognition

The work in (Adams et al., 2018) uses a phoneme recognizer to generate time-aligned phoneme and tone sequences for two different low resource languages, Yongning Na and Eastern Chatino. Both training and test audio are from a single speaker. The reason is that recordings from very few speakers are available in a low resource language, and the primary task is to record audio from one speaker and to transcribe it in order to document and preserve the language. The authors claim that as long as phoneme error rate is low, the time-aligned phoneme sequence helps linguists and speeds up the transcription of the audio significantly.

We measure phoneme recognition accuracy on the Cree audio using three different systems: Persephone (Adams et al., 2018), wav2letter++ (Collobert et al., 2016) (Pratap et al., 2019), and the traditional speech recognition system using Kaldi (Povey and others, 2011) as described in the previous section 4. In both wav2letter++ and Kaldi systems, we decode word sequences and translate them to phoneme sequences in order to measure the phoneme error rate (PER). We show that by just training the language model with increasing amounts of text data, we can significantly reduce the PER, even though the language models are trained from a very small amount of additional text. These systems far outperform the Persephone system in this mode.

All three systems are tested with 17.5 minutes of speech from one female speaker reading scriptures. Some of the experiments use additional scripture texts downloaded from the Internet, from scripture books different from training and test sets.

Persephone is the phoneme recognizer used in (Adams et al., 2018), based on Tensorflow and made publicly available. The model is a bidirectional LSTM neural network architecture with the connectionist temporal classification (CTC) loss function. For this system, we tried training the bidirectional LSTM models with and without speed perturbed Cree training data. The Persephone system did not converge when we used both the video and scriptures training data. Long video segments caused training issues, and video speakers are different from the scripture test speaker. So we only used the scripture training set for training the bidirectional LSTM models. Table 5 gives the phoneme error rate for the scriptures test set using the Persephone system with and without speed perturbed training set. The PER goes down from 23.5% to 20.6% with the speed perturbed training set.

Training set	PER
scriptures training set	23.5%
scriptures training set with SP	20.6%

Table 5: Phoneme error rates (PER) with Persephone for the scripture test set after training with the scriptures training set with / without speed perturbation (SP).

Wav2letter++ is an open-source speech recognition toolkit recently released by Facebook for sequence-to-sequence training and decoding. It is entirely written in C++ and is very fast. Wav2letter++ uses convolutional network based acoustic models and a graph decoding. We have used the same architecture as provided in their documentation for Libri-speech data. Since we have much less data, we also tried an acoustic model with dimensions reduced by half. This smaller acoustic model gave slightly better results, so we give results for this acoustic model only. For wav2letter++, the lexicon contained words and their spellings in syllabics. So there were 141 distinct syllabic symbols used in the dictionary as the spelling alphabet (compared to 26 for English). For decoding, wav2letter++ is driven by a language model. For language model, we used the same 4-gram language model as used in the pre-

vious section. We also used two different training criteria: Connectionist Temporal Classification (CTC) and AutoSegCriterion (ASG). We ran wav2letter++ on a 32 GB Linux machine with an NVIDIA GPU card inside docker. For wav2letter++ also, training with video data included in the training set failed due to memory allocation failure. So we only trained with the scriptures training set with/without speed perturbation, and with CTC or ASG criterion. Table 6 gives the phoneme error rates for the various training conditions. The phoneme error rate is measured by converting words to phoneme sequences. For wav2letter++ the lowest PER was 15.1% on the scriptures test set when the training used ASG criterion. So wav2letter++ gives lower PER than the Persephone system.

Training set	criterion	PER
scriptures training set	CTC	20.9%
scriptures training set	ASG	15.1%
scriptures training set with SP	ASG	16.0%

Table 6: Phoneme error rates (PER) with wav2letter++ for the scripture test set after training with the scriptures training set with / without speed perturbation (SP).

We also translated the word error rates of different systems from sections 3. and 4. to phoneme error rates shown in Table 7. The system numbers correspond to the numbered systems in Tables 2, 3, 4. The lowest phoneme error rate corresponds to system 1: BLSTM system trained on Cree training data only with 2 iterations of back propagation. The 8.7% PER is significantly lower than 20.6% PER achieved by Persephone and 15.1% PER achieved by wav2letter++.

System	WER for scriptures	PER
(1)	27.1%	8.7%
(2)	25.9%	9.1%
(3)	29.5%	9.6%
(4)	31.2%	11.5%
(5)	30.0%	10.1%
(6)	27.6%	9.0%
(7)	28.7%	10.3%
(8)	25.1%	10.0%
(9)	24.6%	9.3%

Table 7: Phoneme error rates (PER) for the various systems in the previous section for the scripture test set (speaker-dependent error rate).

Both wav2letter++ and Kaldi (TDNN-F and BLSTM models) decoding is driven by a language model, while Persephone does not use any language model. The first two systems have access to additional information from the language modeling text. Are they benefiting from this information, and if so, how much additional text is needed? To answer this question, we ran decoding with several different language models, starting from scripture training transcriptions only, and incrementally added text to the language model training. We only trained 3-gram LMs for this comparison, with the vocabulary found in transcriptions only, to make comparison with Persephone as fair as possible.

Table 8 shows the phoneme error rate for the various LMs, for the best wav2letter++ system (from Table 6) and for the BLSTM system (1). The first entry in the table is when language model training data is limited to transcriptions in the Cree training data only. Then all three systems have access to the same language model information for the Cree language. The phoneme error rate is similar for wav2letter++ (20.7%) and Persephone (20.6%), but is significantly lower for the BLSTM system (1) (14.4%).
As more text data is made available for the LM, in the following lines of the table, PER continues to go down both for wav2letter++ and the BLSTM system (1). This shows that for the best phoneme recognition, we should use all the available text for language modeling.

LM Training set	wav2letter	(1)
scriptures training (20k words)	20.7%	14.4%
scriptures training + 50k words	16.9%	10.1%
scriptures training + 100k words	15.8%	9.4%
scriptures training + 158k words	15.4%	8.7%

Table 8: Phoneme error rates (PER) for wav2letter++ and for the BLSTM system (1) for the scripture test set with increasing LM training set.

6. Implications for Language Documentation and Revitalization

How can speech recognition help in language documentation? There are many aspects to language documentation. One aspect is transcription of audio archives and of audio collected from the elders in the community in order to transcribe and preserve the language. For many native languages, a significant portion of the transcription may be done by linguists and second language learners. For them, displaying time-aligned phoneme sequences and word sequences can be a big help. For speaker independent recognizer for Cree described above, the displays will have too many errors. However, fortunately, significant portion of the audio in general is from a few speakers. So a speaker dependent recognizer can be trained from a few hours of transcribed audio, and the transcription of the remaining audio can be speeded up by displaying time-aligned phoneme and word sequences to the transcriber. As we have shown before, the error rates for the phonemes in speaker-dependent scenario can be well below 10%, and for words, below 30%.
Another issue in language documentation is to have content search capability in native audio archives. Since most of the archives in general are spoken by a few speakers, speaker dependent acoustic models can be used for such a search. Usually, the search looks for a sequence of matching phonemes, and a speech recognizer with less than 10% phoneme error rate can provide reasonable search capability with minimal false alarms.
Automatic transcription in words or phonemes, even with relatively large error rates, opens up new avenues for revitalization that simply bypass the transcription bottleneck. The ability to easily search in an approximate automatic transcription can be used to identify specific phrases in a

large audio archive and catalog it by contents. Time-aligned word or phoneme transcriptions make it easy to extract didactic material for language learning, or produce read-along audio books. As our work on East Cree confirms and improves upon previous work on Yongning Na and Eastern Chatino (Adams et al., 2018), we can hope that these methods will apply to many other Indigenous languages.

7. Conclusion

Cree is an Indigenous language spoken in Canada. It is a low-resource language as very little printed text or spoken transcribed audio is available. We could get at most 4.5 hours of transcribed audio, over 1200 hours of untranscribed audio (through CBC radio broadcast archives in Cree) and 260k words of written Cree text. So with this limited data, we estimate word and phoneme error rates in both speaker-independent and speaker-dependent scenario for the best possible speech-to-text systems.

The lowest WER (word error rate) for speaker-independent scenario we achieve is 69.9%. This error rate is too high to accurately transcribe audio from an arbitrary speaker of Cree. However, in the speaker-dependent scenario, we achieved a WER of 24.6% and a PER (phoneme error rate) of 8.7%. These error rates are small enough to help speed up transcription significantly.

We also compare our system with two state-of-the-art end-to-end toolkits. We show that training acoustic deep neural network models in a traditional way still gives significantly lower phoneme error rates, and training language models from additional text (without audio) results in even lower rates. Our experiments also provide quantitative information about minimal amounts of transcription and text documents that lead to useful phoneme recognition accuracies.

8. Acknowledgements

This work was funded in part by National Research Council of Canada (NRC) and the Ministère de l'économie, innovation et exportation (MEIE) of Gouvernement du Québec.

9. Bibliographical References

Adams, O., Cohn, T., Neubig, G., Bird, S., and Michaud, A. (2018). Evaluating Phonemic Transcription of Low-Resource Tonal Languages for Language Documentation. In *Proc. LREC*, pages 3356–3365.

Alam, J., Gupta, V., and Boulianne, G. (2019). Supervised and Unsupervised SAD Algorithms for the 2019 edition of NIST Open Speech Analytic Technologies Evaluation. In *OpenSAT2019 Workshop*, pages 1–21.

Chen, G., Khudanpur, S., Povey, D., Trmal, J., Yarowsky, D., and Yilmaz, O. (2013). Quantifying the value of pronunciation lexicons for keyword search in low resource languages. In *Proc. ICASSP*, pages 8560–8564.

Collobert, R., Puhrsch, C., and Synnaeve, G. (2016). Wav2Letter: an End-to-End ConvNet-based Speech Recognition System. http://arxiv.org/abs/1609.03193.

Fiscus, J. G. (1997). A post-processing system to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER). In *Proc. ASRU*, pages 347–352.

Gales, M. J. F., Knill, K. M., Ragni, A., and Rath, S. P. (2014). Speech Recognition and Keyword Spotting for Low Resource Languages: BABEL project research at CUED. In *Proc. SLTU*, pages 14–16.

Graves, A., Jaitly, N., and Mohamed, A. R. (2013). Hybrid speech recognition with Deep Bidirectional LSTM. In *Proc. ASRU*, pages 273–278.

Gupta, V. and Boulianne, G. (2018). CRIM's system for the MGB-3 English multi-genre broadcast media transcription. In *Proc. Interspeech*, pages 2653–2657.

Gupta, V., Kenny, P., Ouellet, P., and Stafylakis, T. (2014). I-vector-based speaker adaptation of deep neural networks for French broadcast audio transcription. In *Proc. ICASSP*, pages 6334–6338.

Huang, J.-T., Li, J., Yu, D., Deng, L., and Gong, Y. (2013). Cross-Language Knowledge Transfer Using Multilingual Deep Neural. In *Proc. ICASSP*, pages 7304–7308.

Knill, K. M., Gales, M. J., Ragni, A., and Rath, S. P. (2014). Language independent and unsupervised acoustic models for speech recognition and keyword spotting. In *Proc. Interspeech*, pages 16–20.

Manohar, V., Povey, D., and Khudanpur, S. (2017). JHU Kaldi system for Arabic MGB-3 ASR challenge using diarization, audio-transcript alignment and transfer learning. In *Proc. ASRU*, pages 346–352.

Povey, D. et al. (2011). The Kaldi speech recognition toolkit. In *Proc. ASRU*.

Povey, D., Peddinti, V., Galvez, D., Ghahremani, P., Manohar, V., Na, X., Wang, Y., and Khudanpur, S. (2016). Purely sequence-trained neural networks for ASR based on lattice-free MMI. In *Proc. Interspeech*, pages 2751–2755.

Povey, D., Cheng, G., Wang, Y., Li, K., Xu, H., Yarmohamadi, M., and Khudanpur, S. (2018). Semi-orthogonal low-rank matrix factorization for deep neural networks. In *Proc. Interspeech*, pages 3743–3747.

Pratap, V., Hannun, A., Xu, Q., Cai, J., Kahn, J., Synnaeve, G., Liptchinsky, V., and Collobert, R. (2019). Wav2letter++: a Fast Open-Source Speech Recognition System. In *Proc. ICASSP*, pages 6460–6464.

Saon, G., Soltau, H., Nahamoo, D., and Picheny, M. (2013). Speaker adaptation of neural network acoustic models using i-vectors. In *Proc. ASRU*, pages 55–59.

Senior, A. and Lopez-Moreno, I. (2014). Improving DNN speaker independence with I-vector inputs. In *Proc. ICASSP*, pages 225–229.

Trmal, J., Chen, G., Povey, D., Khudanpur, S., Ghahremani, P., Zhang, X., Manohar, V., Liu, C., Jansen, A., Klakow, D., Yarowsky, D., and Metze, F. (2014). A keyword search system using open source software. In *Proc. SLT Workshop*, pages 530–535.

Zhang, X., Trmal, J., Povey, D., and Khudanpur, S. (2014). Improving deep neural network acoustic models using generalized maxout networks. In *Proc. ICASSP*, pages 215–219.