IberSPEECH 2018
21-23 November 2018, Barcelona, Spain

Baseline Acoustic Models for Brazilian Portuguese Using Kaldi Tools

Cassio T. Batista, Ana Larissa da S. Dias, Nelson C. Sampaio Neto

Federal University of Pará, Institute of Exact and Natural Sciences
Augusto Corrêa 1, Belém 660750-110, Brazil
{cassiotb,nelsonneto}@ufpa.br, ana.dias@itec.ufpa.br

Abstract

Kaldi has become a very popular toolkit for automatic speech recognition, showing considerable improvements through the combination of hidden Markov models (HMM) and deep neural networks (DNN). However, in spite of its great performance for some languages (e.g. English, Italian, Serbian, etc.), the resources for Brazilian Portuguese (BP) are still quite limited. This work describes what appears to be the first attempt to create Kaldi-based scripts and baseline acoustic models for BP using Kaldi tools. Experiments were carried out for dictation tasks and a comparison to CMU Sphinx toolkit in terms of word error rate (WER) was performed. Results seem promising, since Kaldi achieved the absolute lowest WER of 4.75% with HMM-DNN and outperformed CMU Sphinx even when using Gaussian mixture models only.

Index Terms: automatic speech recognition, Brazilian Portuguese, Kaldi

1. Introduction

The attempts to simplify the communication between humans and machines are not a novelty. Ever since the emergence of consumer electronic computers, researchers have been doing a lot of effort in order to design more convenient interfaces for controlling and interacting with electronic devices. However, despite the consolidation of keyboards and mice as main input methods for personal computers, as well as touch screens for mobile devices, alternative control interfaces such as speech, body gestures, or even thoughts never ceased to be investigated.

For years, the combination of hidden Markov models (HMM) and Gaussian mixture models (GMM) has been the state-of-the-art technique for acoustic modeling in the automatic speech recognition (ASR) field [1, 2]. Concerning the Brazilian Portuguese language in particular, robust speech recognition systems based on traditional HMM-GMMs have already been proposed [3, 4] using both HTK [5] and CMU Sphinx [6] tools. However, the deep learning approaches that emerged last decade seem to have outperformed HMM-GMM models when replacing the Gaussian mixtures by deep neural networks (DNN) combined with HMMs.

Most works that apply deep learning for speech recognition make use of Kaldi [7], an open-source software package that implements the hybrid HMM-DNN combination. To the best of our knowledge, no previous work has developed a Kaldi recipe for Brazilian Portuguese (BP) yet. Therefore, towards building freely available resources [8] for a large vocabulary continuous speech recognition (LVCSR) system in BP using the Kaldi toolkit, this work presents results for HMM-GMM triphone-based acoustic models in terms of word error rate (WER). A preliminar result for DNN-based models is also presented, but the main development of HMM-DNN hybrid models is still ongoing, given the huge amount of time taken to train them.

2. Related Work

A literature review was conducted on IEEE Xplore and ACM Digital Library. However, most works from ACM simply mention speech recognition as an application of DNNs. Therefore, only papers from IEEE Xplore were considered. After filtering by title and abstract, the most relevant ones were selected and will be shortly detailed below.

Sahu and Ganesh [9] performed a survey on HTK, CMU Sphinx and Kaldi toolkits for different languages regarding their performance in terms of WER. They found that Kaldi achieved the best WER value of 2.7% using the Wall Street Journal (WSJ) English corpus. In another work, Becerra *et al.* [10] presented a comparative case study for Spanish between the conventional HMM-GMM architecture and the recent HMM-DNN model using Kaldi. The audio corpus used includes 1,836 sentences from 87 speakers sampled at 16 kHz, which are a mixture of human voices and text-to-speech utterances. A 20.71% improvement was achieved by the HMM-DNN architecture over the HMM-GMM models: 3.33% against 4.20% of WER, respectively.

Popović *et al.* [11] used Kaldi to develop an HMM-based ASR system for the Serbian language. The audio corpora used in the experiment contains 95 hours of speech sampled at 8 kHz. They obtained a word recognition accuracy of approximately 98%. For Italian, on the other hand, Cosi [12] adapted Kaldi's TIMIT recipe for the FBK ChildIt corpus, which contains approximately 10 hours of speech of children sampled at 16 kHz. The results only show that DNN configurations outperforms the non-DNN ones. Karan *et al.* [13] also used Kaldi to developed a speech recognition system, now for Hindi Odia language. The audio corpus consisted of 2,647 utterances collected from 104 speakers at 8 kHz using mobile phones. The experiment used the conventional HMM-GMM architecture only, and reports the best result of 1.74% WER in the triphone model.

Ali *et al.* [14] presented a complete Kaldi recipe for building Arabic speech recognition systems. The corpus used was the GALE Arabic Broadcast News data set, which consisted of 100,000 speech segments of nine different TV channels, a total of 203 hours of speech data recorded at 16 kHz. In the experiment, the DNN-based system achieved the best results with an overall WER of 26.95%, which is nearly a 10% relative improvement to the HMM-GMM model. Kipyatkova and Karpov [15], on the other hand, built an HMM-DNN acoustic model using Kaldi for Russian language. For training and testing, they used their own speech recorded at 44.1 kHz with 16 bits per sample. The data set was composed by 55 speakers and 16,850 utterances. Two different kinds of neuron activation functions were implemented on the neural network: tanh and p-norm. The results showed that the p-norm function obtained the best WER value of 20.30%.

Another search was performed on the previous IberSPEECH proceedings of 2014 and 2016, where two works stood out. On the first one, Guiroy *et al.* [16] im-

10.21437/IberSPEECH.2018-17

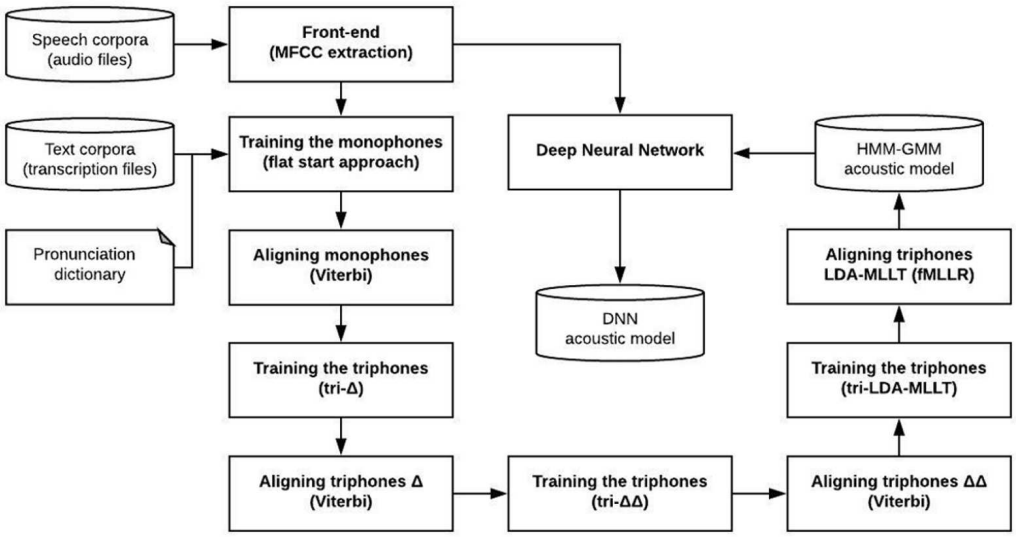

Figure 1: *Training stages of a hybrid HMM-DNN, triphone-based acoustic model on Kaldi.*

plemented an HMM-DNN ASR system in Kaldi and also conducted a comparative study between HMM-based models in both Kaldi and HTK. The Castillian Spanish SpeechDat(II) FDB-4000 audio corpus was used, which contains 43 hours of recordings from 4,000 speakers. The results indicated a 34.02% decrease in WER when comparing the most accurate DNN-based and HMM-based models from Kaldi. A decrease of 53.79% for the HMM-based model in Kaldi could also be observed over their most accurate model from HTK.

On the second work, Zorrilla *et al.* [17] carried out several experiments using Kaldi in order to evaluate different deep-learning approaches for acoustic modeling on well-known Spanish data sets, namely Albayzin, Dihana, CORLEC-EHU and TC-STAR. In addition, the El País text corpus was used for language modeling. The authors found through experiments that all HMM-DNN hybrid acoustic models have outperformed the HMM-GMM ones and work well even with non task-specific language models.

During the research, we also found two works that tackle the ASR problem for BP using deep neural networks. Quintanilha *et al.* [18] presented an open-source, character-based, end-to-end bidirectional long short-term memory (BLSTM) neural network for LVCSR. Several experiments were conducted over a data set of approximately 14 hours of recorded audio and the best performance evaluated in terms of label error rate was 31.53% without the use of any language model. Bonilla *et al.* [19], on the other hand, proposed an end-to-end deep-learning system for recognizing digits, which is compared to a simple multilayer perceptron (MLP) network. It is not clear, however, if the system classifies characters, words or phonemes. The best result is reported as 97.5% of accuracy rate, against 82.8% achieved by the MLP.

According to the literature review, it appears no previous work has developed ASR resources with Kaldi for Brazilian Portuguese yet. Therefore, we believe this is the first attempt to build acoustic models for BP using the toolkit's deep learning approaches.

3. Tools and Resources for BP using Kaldi

In order to build a speech recognition system, one must be provided with a language model (LM), a phonetic dictionary and an acoustic model (AM). The resources and tools used to build each one of the three aforementioned components with Kaldi will be detailed below. It is worth mentioning that the LM and the dictionary are the very same used in CMU Sphinx as well. The steps to train the AMs in particular are similar for both toolkits, but some differences will be pointed out along the text. For further information about acoustic model training for BP using CMU Sphinx tools, the reader is referred to [4].

3.1. Audio Corpora

Speech recognition is a data-driven technology, which means it requires a relatively large amount of labeled data (transcribed audio) to work properly. The corpora used to train the acoustic models with Kaldi are composed by seven data sets, as summarized in Table 1. The data sets contain audio files in an uncompressed, linear, signed PCM (namely, WAVE) format, and are sampled at 16 kHz with 16 bits per sample. It is important to note that the actual number of speakers in West Point was rather reduced due to abundance of foreign words amidst the corpus. Besides, Constitution and Consumer Protection Code corpora share the same speaker.

3.2. Phonetic Dictionary and Language Model

The phonetic dictionary maps every grapheme in the lexicon (orthographic representation) to one or more phonetic transcriptions. The software described in [24] was used to include the pronunciation mapping of each of the 14,518 words into the dictionary. The trigram language model used in this work is described in [3]. It was trained with the SRILM [25] toolkit with 1.6 million phrases from the CETENFolha [26] corpus, yielding a perplexity value of 170. The LM is available in ARPA format, but in order to be used on the Kaldi environment, it was converted to the FST format using the provided `arpa2fst` script.

Table 1: *Audio corpora used to train acoustic models.*

Data set	Ref.	Hours	Words	Speakers
LapsStory	[3]	5h:18m	8,257	5
LapsBenchmark	[3]	0h:54m	2,731	35
Constitution	[20]	8h:58m	5,330	1
Consumer Protection Code	[20]	1h:25m	2,003	1
Spoltech LDC	[21]	4h:19m	1,145	475
West Point LDC	[22]	5h:22m	484	70
CETUC	[23]	144h:39m	3,528	101
Total		170h:51m	14,518	687

Table 2: *Kaldi DNN tools and parameters used for training.*

Tool or Parameter	Value
DNN codebase	`nnet2` ("Dan's DNN")
Script	`train_pnorm_fast.sh`
Hidden layers	2
Activation function	p_norm
`pnorm_output_dim`	3,000
`pnorm_input_dim`	300
`num_epochs`	8
`num_epochs_extra`	5
Minibatch size	512
Learning rate	0.02 down to 0.004

3.3. Acoustic Model

The scripts used on the Brazilian Portuguese audio corpora were based on Kaldi's recipe available for the WSJ corpus [27]. For the sake of comparison, HMM-GMM acoustic models were trained with both Kaldi and CMU Sphinx as well. On Kaldi, the deep learning approach actually uses the HMM-GMM training as a pre-processing stage. Figure 1 shows the steps followed to train HMM-DNN acoustic models on Kaldi based on HMM-GMM triphones. The audio signals are windowed at every 25 ms with 10 ms of overlap in the front-end, being encoded as a 39-dimension vector: 12 Mel frequency cepstral coefficients (MFCCs) [28] using C0 as the energy component, plus 13 delta (Δ, first derivative) and 13 acceleration ($\Delta\Delta$, second derivative) coefficients are extracted from each window.

The AMs are iteratively refined. The flat-start approach models 39 phonemes (38 monophones plus one silence model) as context-independent HMMs. The standard 3-state left-to-right HMM topology with self-loops was used. At the flat-start, a single Gaussian mixture models each individual HMM with the global mean and variance of the entire training data. The transition matrices are also initialized with equal probabilities. The parameters used for extracting MFCCs and training the monophones are the same for both CMU Sphinx and Kaldi.

Nevertheless, Kaldi uses the Viterbi training algorithm [29] to re-estimate the models at each training step, rather than the Baum-Welch algorithm [30] used by CMU Sphinx. Furthermore, Viterbi alignment is applied after each training step in order to allow training algorithms to improve the model parameters, a feature that is not present on CMU Sphinx by default. Subsequently, the context-dependent HMMs are trained for each triphone, first with the delta and after with the acceleration coefficients. Each triphone is represented by a leaf on a decision tree, which is automatically created by both toolkits using statistical methods. Eventually, leaves with similar phonetic characteristics are then tied/clustered together.

The last two steps for training a HMM-GMM acoustic model with Kaldi are the linear discriminant analysis (LDA) [31] combined with the maximum likelihood linear transform (MLLT) [32], followed by the speaker adaptive training (SAT) [33]. Both are included in most tutorials for AM training with Kaldi. The latter, however, was not taken into account during our simulations in order to save time, so only LDA+MLLT was adopted. Moreover, these two steps are not enabled by default on CMU Sphinx and, since they were not used in [4] either, we decided not to include them in order to try to reproduce the results and to save time as well.

The LDA technique takes the feature vectors and splice them across several frames, building HMM states with a reduced feature space. Then, a unique transformation for each speaker is obtained by a diagonalizing MLLT transform. On top of the LDA+MLLT features, the fMLLR alignment algorithm, which is a speaker normalization that uses feature-space maximum likelihood linear regression (fMLLR), is applied [34].

Finally, the HMM-DNN acoustic model is obtained by using the neural network to model the state likelihood distributions as well as to input those likelihoods into the decision tree leaf nodes [16]. In short terms, the network input are groups of feature vectors and the output is given by the aligned state of the HMM-GMM system for the respective features of the input. The number of HMM states in the system also defines the DNN's output dimension [15].

Table 2 shows the most important Kaldi tools and parameters set used to train the deep neural network. Kaldi provides two distinct implementations for DNN training: `nnet1` [35], which is primarily maintained by Karel Veselý; and `nnet2` [36], by Daniel Povey. The former was chosen because it supports CPU training while the `nnet1` enables GPU training only, a resource that was not available for us. Regarding the activation functions of the DNN, we chose the p_norm nonlinearity because it presents a superior performance over the tanh in the literature review [15, 16, 17]. The remaining parameters of the DNN were set based in the Kaldi's documentation as well as in the related works. Since there is actually no parameter to define the number of neurons in the hidden layers for p-norm networks, `pnorm_output_dim` and `pnorm_input_dim` parameters must be set instead, being the latter an integer multiple of the former usually with a ratio of 5 or 10 [37]. The number of epochs is given by the sum of the `num_epochs` and `num_epochs_extra` parameters. The first one was supposed to be 15, but it is recommended to reduce it when the computational environment is not very high powered [37], so we choose 13 (8+5) to be the total number of epochs. The learning rate was set to vary from 0.02 down to 0.004 during the default number of epochs; and to stay constant at 0.004 for the next extra epochs [15].

4. Experimental Tests and Results

Tests were executed on an HP EliteDesk 800 G1 desktop computer equipped with a Intel® Core™ i5-4570 3.20 GHz CPU, 8 GB of RAM and 1 TB of hard disk storage. During the experiments, the LapsBenchmark corpus was held exclusively for

Table 3: *WER (%) achieved by CMU Sphinx and Kaldi toolkits.*

Toolkit	# Gauss.	Number of tied-states or senones					
		500	1,000	2,000	4,000	6,000	8,000
CMU Sphinx (standard)	2	21.70	18.60	17.30	16.20	15.40	15.10
	4	17.50	15.50	14.60	13.10	13.10	12.70
	8	15.50	13.40	12.80	11.90	11.80	12.20
	16	14.20	12.80	11.90	**11.10**	11.20	11.30
Kaldi (tri-Δ)	2	26.54	21.31	18.25	15.69	14.32	14.04
	4	21.61	18.07	15.81	13.46	12.47	11.91
	8	19.91	15.80	13.67	11.66	10.85	10.31
	16	17.25	14.10	12.09	10.64	9.68	**9.31**
Kaldi (tri-Δ+ΔΔ)	2	25.47	20.63	17.28	15.18	13.64	13.42
	4	21.13	17.79	14.65	13.10	11.93	11.39
	8	18.72	15.26	13.02	11.15	11.03	10.20
	16	17.17	13.61	12.06	10.47	9.31	**9.23**
Kaldi (tri-LDA-MLLT)	2	19.91	15.36	12.12	9.99	9.60	9.13
	4	16.84	13.43	11.15	9.12	8.65	8.04
	8	14.42	11.93	9.63	8.15	7.58	6.99
	16	12.96	10.75	8.85	7.60	6.79	**6.50**

testing and the six other corpora were used for training. Unfortunately, no clusters or graphic cards could be used for training the models. Therefore, due to the computational burden and the lack of hardware resources, it was not possible to develop DNN-based AMs for all combinations of HMM-GMM acoustic models with Kaldi.

Table 3 shows the results obtained with both CMU Sphinx and Kaldi. For Kaldi, by the way, the WER was evaluated across all triphone training steps in order to perform a more complete comparison to CMU Sphinx results, since neither the LDA+MLLT stage or the fMLLR alignment were included for this toolkit. For Sphinx, as expected, the WER decreases as we increase both the number of Gaussians and the number of tied-states of the model. However, the values seem to converge after 4,000 senones and 8 Gaussians. The lowest WER value achieved was approximately 11.1% with 4,000 senones and 16 Gaussian densities.

For Kaldi, however, we found that the previous convergence shown on CMU Sphinx results does not occur. As we increase the number of senones and the number of Gaussians, the WER values linearly drop. Besides, it can be seen that the lowest WER values for the first two triphone training steps (tri-Δ and tri-ΔΔ) are already lower than the best one achieved by CMU Sphinx: 9.31% and 9.23%, respectively. The global, lowest WER value obtained with Kaldi was 6.5% with 8,000 tied-states and 16 Gaussians at the tri-LDA-MLLT step, which is equivalent to 128,000 leaves on the decision tree, according to Kaldi's parameter settings (which is basically the result of the product $8{,}000 \times 16$).

As proof of concept, we trained a DNN-based acoustic model on the best HMM-GMM model produced with Kaldi. The WER value dropped from 6.5% to 4.75%, an improvement of 26.92%. When compared to the lowest WER value obtained with CMU Sphinx, the improvement increases to 57.21%, which is a huge difference for dictation tasks.

5. Conclusions and Future Works

This paper addressed the first attempt to develop a speech recognition system for large vocabulary (LVCSR) in Brazilian Portuguese using the Kaldi toolkit. Triphone-based, HMM-GMM acoustic models with different values of Gaussians and tied-states were trained with Kaldi and CMU Sphinx tools in order to establish a comparison in terms of word error rate (WER). The evaluation results showed that the systems perform better as we increase the number of Gaussian densities per mixture and the number of tied-states. For CMU Sphinx, the results obtained are in accordance to [4], in spite of the current WER achieved being lower, possibly due the larger corpora used for training the models.

Results also showed that Kaldi definitely outperformed CMU Sphinx even without the use of its deep learning tools. An explanation might be the use of Viterbi algorithm for training (rather than Baum-Welch), as well as the use of Viterbi alignments in between each training stage, which is said to improve or refine the parameters of the model [38]. With the use of DNNs, Kaldi presents an improvement of 57.21% over the best HMM-GMM-based acoustic model built with CMU Sphinx.

As future work, we plan to finish training the HMM-DNN triphone-based AMs with Kaldi and consequently make them publicly available (together with the recipe) [8] to the community. We also expect to test with 32 and 64 densities per mixture, now evaluating the decoding time too in terms of the real-time factor (xRT) as the WER possibly decreases. Furthermore HTK's latest release also has an implementation of deep learning algorithms, which may join the next comparisons.

6. Acknowledgements

The authors would like to thank Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) and Federal University of Pará (UFPA) under Edital n° 06/2017 – PIBIC/PROPESP for granting scholarships.

7. References

[1] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb 1989.

[2] X. Huang, A. Acero, and H.-W. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.

[3] N. Neto, C. Patrick, A. Klautau, and I. Trancoso, "Free tools and resources for brazilian portuguese speech recognition," *Journal of the Brazilian Computer Society*, vol. 17, no. 1, pp. 53–68, Mar 2011. [Online]. Available: https://doi.org/10.1007/s13173-010-0023-1

[4] R. Oliveira, P. Batista, N. Neto, and A. Klautau, "Baseline acoustic models for brazilian portuguese using cmu sphinx tools," in *Computational Processing of the Portuguese Language*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 375–380.

[5] S. Young, D. Ollason, V. Valtchev, and P. Woodland, *The HTK Book*. Cambridge University Engineering Department, version 3.4, 2006.

[6] D. Huggins-Daines, M. Kumar, A. Chan, A. W. Black, M. Ravishankar, and A. I. Rudnicky, "Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 1, May 2006, pp. I–I.

[7] D. Povey, A. Ghoshal, G. Boulianne, N. Goel, M. Hannemann, Y. Qian, P. Schwarz, and G. Stemmer, "The kaldi speech recognition toolkit," in *In IEEE 2011 workshop*, 2011.

[8] GitLab. (2018) Tutorial para treino de modelo acústico com kaldi. [Online]. Available: https://gitlab.com/fb-asr/fb-am-tutorial/kaldi-am-train

[9] P. K. Sahu and D. S. Ganesh, "A study on automatic speech recognition toolkits," in *2015 International Conference on Microwave, Optical and Communication Engineering (ICMOCE)*, Dec 2015, pp. 365–368.

[10] A. Becerra, J. I. de la Rosa, and E. González, "A case study of speech recognition in spanish: From conventional to deep approach," in *2016 IEEE ANDESCON*, Oct 2016, pp. 1–4.

[11] B. Popović, S. Ostrogonac, E. Pakoci, N. Jakovljević, and V. Delić, "Deep neural network based continuous speech recognition for serbian using the kaldi toolkit," in *Speech and Computer*. Cham: Springer International Publishing, 2015, pp. 186–192.

[12] P. Cosi, "A kaldi-dnn-based asr system for italian," in *2015 International Joint Conference on Neural Networks (IJCNN)*, July 2015, pp. 1–5.

[13] B. Karan, J. Sahoo, and P. K. Sahu, "Automatic speech recognition based odia system," in *2015 International Conference on Microwave, Optical and Communication Engineering (ICMOCE)*, Dec 2015, pp. 353–356.

[14] A. Ali, Y. Zhang, P. Cardinal, N. Dahak, S. Vogel, and J. Glass, "A complete kaldi recipe for building arabic speech recognition systems," in *2014 IEEE Spoken Language Technology Workshop (SLT)*, Dec 2014, pp. 525–529.

[15] I. Kipyatkova and A. Karpov, "Dnn-based acoustic modeling for russian speech recognition using kaldi," in *Speech and Computer*. Cham: Springer International Publishing, 2016, pp. 246–253.

[16] S. Guiroy, R. de Cordoba, and A. Villegas, "Application of the kaldi toolkit for continuous speech recognition using hidden-markov models and deep neural networks," in *Advances in Speech and Language Technologies for Iberian Languages. IberSPEECH 2016*, ser. LNCS, vol. 10077. Springer, November 2016.

[17] A. L. Zorrilla, N. Dugan, M. I. Torres, C. Glackin, G. Chollet, and N. Cannings, "Some asr experiments using deep neural networks on spanish databases," in *Advances in Speech and Language Technologies for Iberian Languages. IberSPEECH 2016*, ser. LNCS, vol. 10077. Springer, November 2016.

[18] I. M. Quintanilha, L. W. P. Biscainho, and S. L. Netto, "Towards an end-to-end speech recognizer for portuguese using deep neural networks," in *XXXV Simpósio Brasileiro de Telecomunicações e Processamento de Sinais*, September 2017, pp. 709–714. [Online]. Available: http://www.sbrt.org.br/sbrt2017/anais/1570360756.pdf

[19] D. A. Bonilla, N. Nedjah, and L. de Macedo Mourelle, "Reconhecimento automático de fala em português usando redes neurais artificiais profundas," in *Anais do 12 Congresso Brasileiro de Inteligência Computacional*, C. J. A. Bastos Filho, A. R. Pozo, and H. S. Lopes, Eds. Curitiba, PR: ABRICOM, 2015, pp. 1–6.

[20] PCD Legal. (2018) PCD legal: Acessível para todos. [Online]. Available: http://www.pcdlegal.com.br/

[21] LDC. (2018) Cslu: Spoltech brazilian portuguese version 1.0. [Online]. Available: https://catalog.ldc.upenn.edu/LDC2006S16

[22] LDC. (2018) West point brazilian portuguese speech. [Online]. Available: https://catalog.ldc.upenn.edu/LDC2008S04

[23] PUC-Rio. (2018) Centro de estudos em telecomunicações (CETUC). [Online]. Available: http://www.cetuc.puc-rio.br/

[24] A. Siravenha, N. Neto, V. Macedo, and A. Klautau, "Uso de regras fonológicas com determinação de vogal tônica para conversão grafema-fone em Português Brasileiro," *7th International Information and Telecommunication Technologies Symposium*, 2008.

[25] A. Stolcke, "SRILM - an extensible language modeling toolkit," *International Conference on Spoken Language Processing*, 2002. [Online]. Available: http://www.speech.sri.com/projects/srilm/

[26] Linguateca. (2018) Corpus de extractos de textos electrónicos nilc/folha de s. paulo (CETENFolha). [Online]. Available: https://www.linguateca.pt/cetenfolha/

[27] GitHub. (2018) Kaldi speech recognition toolkit. [Online]. Available: https://github.com/kaldi-asr/kaldi

[28] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, Aug 1980.

[29] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, April 1967.

[30] L. R. Welch, "Hidden markov models and the baum-welch algorithm," in *IEEE Information Theory Society Newsletter*, vol. 53, 2003, pp. 10–12.

[31] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Wiley Interscience, 2000.

[32] R. A. Gopinath, "Maximum likelihood modeling with gaussian distributions for classification," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, vol. 2, May 1998, pp. 661–664 vol.2.

[33] S. Matsoukas, R. Schwartz, H. Jin, and L. Nguyen, "Practical implementations of speaker-adaptive training," in *DARPA Speech Recognition Workshop*, 1997.

[34] M. J. F. Gales, "Maximum likelihood linear transformations for hmm-based speech recognition," *Computer Speech and Language*, vol. 12, no. 2, pp. 75–98, April 1998. [Online]. Available: https://doi.org/10.1006/csla.1998.0043

[35] K. Veselý, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *INTERSPEECH 2013*, 2013, pp. 2345–2349.

[36] D. Povey, X. Zhang, and S. Khudanpur, "Parallel training of dnns with natural gradient and parameter averaging," http://arxiv.org/pdf/1410.7455v8, Tech. Rep., 2014.

[37] Kaldi. (2018) Dan's dnn implementation. [Online]. Available: http://kaldi-asr.org/doc/dnn2.html

[38] E. Chodroff. (2018) Kaldi tutorial: Training overview. [Online]. Available: https://www.eleanorchodroff.com/tutorial/kaldi/training-overview.html