Sociedad de Ingeniería de Audio

Artículo de Congreso

Congreso Latinoamericano de la AES 2018
24 a 26 de Septiembre de 2018
Montevideo, Uruguay

A new automatic speech recognizer for Brazilian Portuguese based on deep neural networks and transfer learning

Igor M. Quintanilha,[1] Luiz W. P. Biscainho,[1] and Sergio L. Netto[1]

[1] Universidade Federal do Rio de Janeiro, DEL/Poli & PEE/COPPE
Rio de Janeiro, RJ, 21941-598, Brasil

igor.quintanilha@smt.ufrj.br, wagner@smt.ufrj.br, sergioln@smt.ufrj.br

ABSTRACT

This paper addresses the problem of training deep learning models for automatic speech recognition on languages with few resources available, such as Brazilian Portuguese, by employing transfer learning strategies. From a backbone model trained in English, the best fine-tuned network reduces the character error rate by 9%, outperforming previous related works.

0 INTRODUCTION

Deep neural networks [1] have changed the field of machine learning. The last couple of years have shown that deep learning is production-ready, embedded in almost every mobile system, from face detection to battery-saving adaptive software [2, 3].

Automatic speech recognition (ASR) systems have also benefited from deep neural networks — from almost 30 years of no significant accuracy improvement to an outstanding 30% improvement in 2012 [4], and to nearly human accuracy in 2017 [5], currently being deployed in several voice assistant products like Alexa and Google Home [6, 7].

However, ASR itself and deep learning-based systems are data-driven, requiring considerable amounts of data to produce reasonable results. Since the publication of TIMIT dataset [8], the offer of open-source English speech resources has grown, with the advent of the TED-LIUMv2 [9] (207 hours of speech) and the LibriSpeech [10] (1000 hours of speech) datasets, for example. Meanwhile, Baidu has shown [11] results in English ASR using their private dataset comprising more than 11,000 hours of speech, showing an improvement of ∼8% over the performance attained when solely using the LibriSpeech dataset.

High-accuracy ASR systems for some languages suffer from the lack of annotated speech or public corpora [12]. This work aims to address this issue by studying and showing how trained ASR systems for English can be beneficial to construct a more efficient

system for such languages by applying transfer learning techniques. From the pre-trained DeepSpeech 2 backbone model [11], this work investigates two setups for performing the transfer learning to Brazilian Portuguese. The two sets of experiments adapt a backbone model trained in English to transcribe Brazilian Portuguese speech, one using the same alphabet size and the other a broader alphabet.

The code developed for this work is open-source under MIT License and available at: http://github.com/igormq/aes-lac-2018.

The rest of the paper is organized as follows. Sec. 1 describes related work in the ASR area and in the transfer learning field. Sec. 2 details the backbone model based on the Deep Speech 2 network, whereas Sec. 3 depicts transfer knowledge methods in order to train the backbone model over a small dataset. Sec. 4 details the English and Brazilian Portuguese datasets used in this work, whereas Sec. 5 describes a set of experiments on training an ASR model for Brazilian Portuguese. Finally, in Sec. 6 conclusions are drawn.

1 RELATED WORK

Recently, deep learning has taken over the ASR field [13, 14, 5]. Since 2012, deep learning has evolved from being part of the ASR pipeline to be the entire model, mostly due to the connectionist temporal classification [15] and the sequence-to-sequence models [16]. Since then, ASR word error rate has dramatically improved, from 18.4% in 2013 [17] to 9.1% in 2017 [18] in the commonly used Hub5'00 evaluation [19]. The standard benchmarks, however, are mainly exclusive to English or Chinese, and not much effort has been made in benchmarking other languages, such as Brazilian Portuguese. This lack of effort is primarily due to the shortage of annotated speech in these languages, also termed as under-resourced languages. One way to overcome this reduced amount of available data is using alternative techniques such as transfer learning [20].

Transfer learning is an approach that uses knowledge learned from one specific task and applies this knowledge to another different, but related, task. The transfer learning technique is not a new idea [20]; although it is easy to deploy in computer vision systems [21], in ASR systems it is more challenging due to the recurrent models required. Transfer learning consists of two steps: pre-training, where the model[1] will be trained for the first task (e.g. classify objects in an image) with a lot of data; and fine-tuning, where the model will be adapted to the other task (e.g. detect different dog breeds), usually over a smaller dataset. This technique has several advantages in many machine learning areas, but it becomes more fruitful in the deep learning field.

Throughout the years, a few papers have investigated the use of transfer learning techniques for under-resourced languages. In [22], the authors investigated several ways of adapting a model trained in a large dataset to a small one. Among them, it is worthwhile to cite the heterogeneous transfer learning, which trains the same base model in multiple tasks and languages, and model adaptation. Different from [22], this work presents recent findings in transfer learning for end-to-end models. Kunze et al. [23] explored transfer learning under the optics of GPU constraints from English to German using the Wav2Letter [24] model, which is an all-convolutional network. However, the German dataset has hundreds of hours of speech, while the Brazilian Portuguese dataset used in this paper has only 14 hours of speech.

In a previous work [12], a relative shallow model, with five layers, based solely on long short-term memories [25], has shown the effectiveness of applying end-to-end learning to a relatively small dataset. The authors also reported that more layers did not bring any improvement due to the small dataset. That work is continued here by increasing the capacity of the network and enabling bigger (and better) models using transfer learning methods.

2 BACKBONE MODEL

Fig. 1 illustrates the model architecture of this paper: it is based on the DeepSpeech 2 [11] model with two convolutional (CNN) and five bidirectional recurrent (RNN) layers.

Let us define a single utterance as $\boldsymbol{x} \in \mathbb{R}^{T \times D}$ and its respective transcription as $\boldsymbol{y} \in \mathbb{R}^{C}$ sampled from an arbitrary dataset. Each utterance $\boldsymbol{x}$ is a time-series of length T and dimension D, where each time slice is a vector $\boldsymbol{x}_t$, $t = 0, \ldots, T-1$. Here, as in [11], it is used a normalized power spectrogram calculated over the audio signal with D frequency bins. The goal of the model is to convert the audio sequence into the transcription $\boldsymbol{y}$.

The first main layers are spatial convolutions, usually found in image-related tasks. The convolution layer is used to increase the model capacity without exponentially increasing the number of parameters. In [11], the authors argued that the convolution in frequency models speakers' variability better than fully connected (FC) networks. Moreover, tuning the CNN parameters, such as strides and kernel sizes, help to release redundant information found in the spectrogram as well as reducing the number of outputs to be fed into the subsequent, more expansive, layers. Tab. 1 shows the CNN parameters used in this work, which yield the best performance, according to [11]. Each convolution layer is followed by a batch normalization layer [26] and a clipped ReLU non-linearity $(\min(\text{ReLU}(x), 20))$. This setup reduces the input dimension at least 8 times[2].

After two convolutions layers, the output $\boldsymbol{h}_t^l$ is fed to a stack of 5 bidirectional recurrent neural networks.

[1] also termed as backbone model.

[2] $\approx x/8 - 19$.

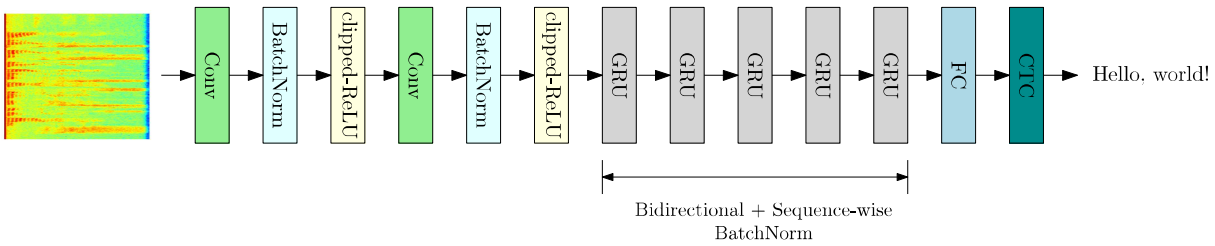

Figure 1: Deep Speech 2 like model. It consists of 2 convolutional layers, 5 GRU layers, and 1 FC with interleaved Batch Norm layers, totalizing over 30M parameters.

Table 1: Convolutional layers in the Deep Speech 2 model.

	# channels	kernel	stride	padding	# parameters
Conv 1	32	(41, 11)	(2, 2)	(0, 10)	14,464
Conv 2	32	(21, 11)	(2, 1)	(0, 0)	236,576
					251,040

Different from [11], this work uses gated recurrent units (GRU) instead of Elman's RNN. GRU is a simplified version of long short-term memories (LSTM) [25] where the forget and input gates are fused. For a unidirectional GRU layer l, at each timestep t, the activation vector $\boldsymbol{a}_t^l \in \mathbb{R}^{2H}$ is computed as follows

$$\boldsymbol{a}_t^l = \boldsymbol{W}_{\boldsymbol{h}^l,\boldsymbol{h}^{l-1}}^T \boldsymbol{h}_t^{l-1} + \boldsymbol{W}_{\boldsymbol{h}^l,\boldsymbol{h}^l}^T \boldsymbol{h}_{t-1}^l + \boldsymbol{b}^l, \quad (1)$$

where $\boldsymbol{h}_t^0 = \boldsymbol{x}_t$. We then divide $\boldsymbol{a}_t^l$ into two vectors $\boldsymbol{a}_t^l = [\boldsymbol{a}_{t,u}^l, \boldsymbol{a}_{t,r}^l]^T$, for $\boldsymbol{a}_{t,\star}^l \in \mathbb{R}^H$, and compute the reset gate $\boldsymbol{r}_t^l = \sigma(\boldsymbol{a}_{t,r}^l) \in \mathbb{R}^H$ and the update gate $\boldsymbol{u}_t^l = \sigma(\boldsymbol{a}_{l,u}^l) \in \mathbb{R}^H$, where σ is the sigmoid function applied element-wise. Finally, we compute the next hidden state $\boldsymbol{h}_t^l$ as

$$\tilde{\boldsymbol{h}}_t^l = \tanh\left[\boldsymbol{W}_{\boldsymbol{h}^l\tilde{\boldsymbol{h}}^l}^T \boldsymbol{h}_{t-1}^l + \boldsymbol{W}_{\boldsymbol{h}\tilde{\boldsymbol{h}}}^T (\boldsymbol{r} \odot \boldsymbol{h}^{(t-1)}) + \boldsymbol{b}_{\tilde{\boldsymbol{h}}}\right] \quad (2)$$

$$\boldsymbol{h}_t^l = (1 - \boldsymbol{u}_t^l) \odot \boldsymbol{h}_{t-1}^l + \boldsymbol{u} \odot \tilde{\boldsymbol{h}}_t^l, \quad (3)$$

where $\boldsymbol{W}_{\boldsymbol{h}^l\tilde{\boldsymbol{h}}^l} \in \mathbb{R}^{D\times H}$, $\boldsymbol{W}_{\boldsymbol{h}\tilde{\boldsymbol{h}}} \in \mathbb{R}^{h\times H}$, $\boldsymbol{b}_{\tilde{\boldsymbol{h}}} \in \mathbb{R}^H$ are learnable parameters of GRU as well, and $\odot$ is the element-wise product operator. A sequence-wise batch normalization [27] is also applied to the input-hidden connections of each recurrent layer for a faster convergence, i.e.,

$$\boldsymbol{a}_t^l = \mathrm{BN}(\boldsymbol{W}_{\boldsymbol{h}^l,\boldsymbol{h}^{l-1}}^T \boldsymbol{h}_t^{l-1}) + \boldsymbol{W}_{\boldsymbol{h}^l,\boldsymbol{h}^l}^T \boldsymbol{h}_{t-1}^l + \boldsymbol{b}^l. \quad (4)$$

In the bidirectional setting, there are two unidirectional GRUs for each layer, one proceeding forward and the other backward in time, which generate outputs $\overrightarrow{\boldsymbol{h}}_t^l$ and $\overleftarrow{\boldsymbol{h}}_t^l$. Then, the two outputs are summed into a single output $\boldsymbol{h}_t^l = \overrightarrow{\boldsymbol{h}}_t^l + \overleftarrow{\boldsymbol{h}}_t^l$ to be fed into next layer.

After the bidirectional recurrent layers, one fully connected layer is employed to generate the unnormalized scores over the label set

$$\boldsymbol{h}_t^L = \boldsymbol{W}^L \boldsymbol{h}_t^{L-1} + \boldsymbol{b}^L. \quad (5)$$

Finally, we can calculate the probability distributions over the labels with a softmax layer

$$p(\boldsymbol{l}_t|\boldsymbol{x}) = \frac{\exp(\boldsymbol{h}_t^L)}{\sum \boldsymbol{1}^T \boldsymbol{h}_t^L}. \quad (6)$$

At each timestep t, the softmax layer predicts a label $p(l_t|\boldsymbol{x})$, where l_t is either a label from a set comprising the character set and the blank token. Finally, the predicted transcription is decoded from the sequences given according to the probability distributions. The loss function adopted is the connectionist temporal classification (CTC) [28], which accounts for each possible sequence of labels that can be translated into the same transcription, not being necessary a frame-level annotation. Given the input-output pair $(\boldsymbol{x}, \boldsymbol{y})$ and the network parameters $\boldsymbol{\theta}$, the loss $L(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta})$ and its gradient with respect to the network parameters $\nabla_{\boldsymbol{\theta}} L(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta})$ can be calculated over the batches. The gradient is then back-propagated through time in order to update the network parameters.

3 TRANSFERRING KNOWLEDGE

Training the Deep Speech 2 model with more than 30 million parameters with a few hours of Brazilian Portuguese speech is a challenge. This section investigates the transfer learning method to enable training bigger and deeper models with small datasets.

3.1 Transfer learning

One of the main challenges to deep learning practitioners is that most of the algorithms are data-driven, requiring a considerable amount of annotated data to generate reasonable results, which can be time-consuming and even prohibitively expensive. Profiting from the breakthrough of deep learning in a large number of small datasets is mainly possible due to the transfer learning approach.

Neural networks have greatly benefited from transfer learning due to the intrinsic characteristic of layered-pattern learning. The first layers usually learn more abstract and generic concepts (e.g., lines and circles in an image) while deeper layers usually learn more task-specific patterns (e.g., dogs or human faces). These abstract layers act as generic feature extractors, which may be used in other tasks that have not been

trained on. Successful approaches have been found in the field of computer vision [21], due to the open-available ImageNet dataset, a broad set of more than one million of images, organized in 1000 classes. Deep models trained with this dataset learn generic filters and features that can be used to adapt the model to a different, but related, task. In natural language processing, transfer learning has also been widely applied in the word embedding field [29].

Ordinarily, models with pre-trained weights have several favorable characteristics, leading to better generalization [20], better initial performances, steeper slopes, and higher asymptotes [30] than models initialized with random weights, as depicted in Fig. 2. Two factors are essential when using the transfer learning method: target dataset size and similarity between the datasets.

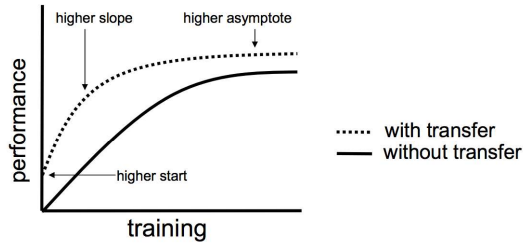

Figure 2: Transfer-learned models have better initial performance, steeper learning curve, and higher final accuracy in training than models without transfer learning. Adapted from [30].

Small and similar. Usually, it has a higher chance to overfit, then the best option is to freeze the first layers and only retrain the last ones.

Large and similar. Less chance to overfit, hence the entire model can be retrained.

Small and different. Deeper layers have less similarity with the new task. Thus, it is reasonable to remove the final layers, which are more task-specific for the original problem.

Large and different. We can afford training from scratch, but it has been shown that training a model from a pre-trained one is beneficial.

Nevertheless, transfer learning methods have a model constraint. In order to use a pre-trained model, machine learning practitioners must comply with the model topology to avoid a thorough change into the original model; otherwise, the co-dependency between layers will be lost and the pre-trained weights will not be worth much for transfer learning. Another common sense is that the learning rate should be usually lower in pre-trained models: since weights are better initialized, they are not expected to be modified so quickly during training.

4 DATASETS

This section describes the two datasets used in this work: a large one in English and freely available, and a small one, in Brazilian Portuguese and partially freely available.

4.1 LibriSpeech

The LibriSpeech [10] is a speech corpus derived from reading audiobooks from the LibriVox project, totalizing almost 1000 hours of reading speech at 16 kHz, one of the most significant public available English dataset. This dataset is composed of two test sets called `test-clean` and `test-other` which have the lowest and the highest word error rate (WER), respectively, evaluated by the authors in a state-of-the-art model at that time. Due to its massive amount of data, the LibriSpeech corpus is a perfect candidate to pre-train the Deep Speech 2 model.

4.2 Brazilian Portuguese speech dataset

The Brazilian Portuguese speech dataset is an ensemble of three publicly available datasets and one paid [12]. It contains almost 14 hours of non-conversational speech, totalizing 425 different speakers and more than 12,000 utterances sampled at 16 kHz in a non-controlled environment.

5 EXPERIMENTS

In this section, we investigate two approaches to perform the transfer learning, as previously described in Sec. 3.

5.1 Backbone model pre-training

In all experiments, we have trained a backbone model using the LibriSpeech dataset. Table 2 summarizes the backbone model architecture and training hyperparameters. The network input is the normalized spectrogram, as described in Sec. 2, calculated using a Hamming window with 320 samples and a hop size of 160 samples, thus $D = 161$. Each recurrent layer has $H = 800$ hidden units and the alphabet contains $C = 29$ labels (A...Z, apostrophe, space, and blank label). Training was carried out using the stochastic gradient descent method with momentum, using a learning rate of 3e-4, a momentum of 0.9, an annealing rate of 0.9091, and a gradient norm clipping of 400 for over 15 epochs with a batch size of 10. In the first epoch, we use a curriculum learning called SortaGrad [11], which consists in sorting the utterances by its length, to accelerate training. After the first epoch, the batches are randomly organized. The predicted sequence is decoded using the greedy search [28]. Tab. 3 shows the results of the backbone model, which is close to the values found in the literature, without a proper language model for decoding and extra data. The Paddle Paddle [31] implementation differs from others by adopting larger recurrent layers (with 2048 hidden units each) and a different activation function.

Table 2: Deep Speech 2 backbone model: architecture and hyperparameters. View operation is a reshape over the inputs.

	Operation	Kernel size	Stride	Feature maps	Padding	Nonlinearity
Network - Input	$B \times 1 \times 161 \times T$					
	Convolution	41×11	2×2	32	0×10	BN-clippedReLU
	Convolution	21×11	2×1	32	0×0	BN-clippedReLu
	View	$B \times 32 \times 21 \times T_{\text{out}} \rightarrow T_{\text{out}} \times B \times 32 * 21$				
x 5	BatchRNN	*hidden size: 800*				
	BN					
	View	$T \times B \times 800 \rightarrow B \times T \times 800$				
	FC	*output size:* $B \times T \times 29$				softmax + CTC
BatchRNN Module						
	Sequence-wise BN					
	Bidirectional GRU					tanh
Sequence-wise BN Module	t, b, d					
	View	$t \times b \times d \rightarrow t * b \times d$				
	BN					
	View	$t * b \times d \rightarrow t \times b \times d$				
	Preprocessing	Normalized linear spectrogram (window size = 320, hop size = 160)				
	Optimizer	SGD with momentum (lr= $3e-4$, momentum= 0.9), SortaGrad enabled				
	Max gradient norm	400				
	Learning rate annealing	0.9091				
	Batch size	10				
	Epochs	15				
	Decoding	Greedy search decoder				

Table 3: Backbone model word error rate (WER) compared with Sean Naren [32] and Paddle Paddle [31] implementations.

	Ours	Sean Naren	Paddle Paddle
test-clean	11.66%	11.27%	6.85%
test-other	30.70%	30.74%	21.18%

5.2 Fine-tuning with the same label set

The first set of experiments investigates the transfer learning method, which uses the same character set as the pre-trained model. Tab. 4 summarizes the results and Fig 3 shows the character error rate over the epochs. Freezing the model and training only the RNNs and FC layers did not bring any improvement over the model trained from scratch (with the same setup as the backbone model) with the Brazilian Portuguese dataset. Fine-tuning the entire model shows an advantage, with the character error rate (CER) dropping from 22.19% to 16.17%, by far surpassing the previous work [12], as depicted in Tab. 5, where the fine-tuned model predicts better the word boundaries than the model trained from scratch. Due to the differences between the two languages, only training the recurrent and FC layers are not enough, thus fine-tuning the entire model makes the network adapt better to Brazilian Portuguese. Also, fine-tuning with lower learning rate (3e-5) has worse results than training with the backbone learning rate, which indicates that the pre-trained weights generalize better but the two datasets are more distinct than initially thought, requiring a higher learning rate.

Table 4: Transfer learning into the same label set.

	[12]	scratch	freeze	fine-tuning
CER	25.13%	22.19%	30.80%	**16.17%**

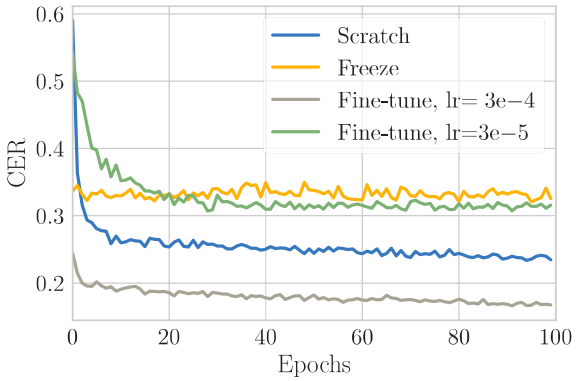

Figure 3: Character error rate over time in the validation set. A lower learning rate did not improve the results.

5.3 Fine-tuning with a broader label set

The second set of experiments, in contrast to the first one, performs the transfer learning using a broader set of characters, including the Brazilian Portuguese accents — expanding the number of characters C from 29 to 43. Since the number of characters is different between the backbone and fine-tuned model, the last fully connected pre-trained weights may be lost. These two sets, however, have some intersection (both sets have A...Z characters). One could either not take into account this intersection and initialize all FC weights from a random distribution, or initialize the subset of FC weights related to the same characters. Tab. 5 shows that the network was able to correctly predict the Brazilian Portuguese accents and Tab. 6 presents the results for both types of initialization. It is clear that seizing the better weights from the backbone model is advantageous, reducing the CER by 5.05% if the last weights are randomly initialized, and by 5.06% if the subset FC layer is not randomly initialized; this indicates that

Table 5: Comparison between original transcription and the ones predicted by different models and label sets.

model	accent?	transcription
reference	-	segundo ele a polícia não iria ceder a exigências
scratch	No	segundo elha a posnicie nao iria cebe aesxigencia
fine-tuning	No	segundo elhe a policia nao eria cede a exigencias
non-random FC weights	Yes	segundo ele apolícia não eria cde a eigências

the phonemes related to each character are different between the English and Brazilian Portuguese languages, and the better initialization of some weights in the last layer brings no visible improvements.

Table 6: Transfer learning into a broader label set.

	scratch	random FC weights	non-random FC weights
CER	22.78%	17.73%	**17.72%**

6 CONCLUSIONS

In this work, we have discussed transfer knowledge techniques applied to the ASR field for under-resourced languages as Brazilian Portuguese. From a pre-trained Deep Speech 2 backbone model on LibriSpeech, we have conducted several experiments on transfer learning from English to Brazilian Portuguese, showing an improvement of 8.96% over the previous work and a character error rate of 17.72% using a broader label set covering all Portuguese Brazilian accents. These results show that the end-to-end models for ASR can significantly benefit from transfer knowledge methods.

7 ACKNOWLEDGMENT

This research was partially supported by CNPq and CAPES. We are thankful to Sean Naren on whose implementation ours was inspired.

REFERENCES

[1] I. Goodfellow, Y. Bengio, A. Courville, *Deep learning*, Adaptive Computation and Machine Learning (MIT Press, Cambridge, England) (2016).

[2] Computer Vision Machine Learning Team, "An on-device deep neural network for face detection," https://apple.co/2s86Asy, accessed: 2018-05-22.

[3] J. Smith, S. Rosen, C. Gamble, "DeepMind, meet Android," http://bit.ly/2KNRf8f, accessed: 2018-05-22.

[4] G. E. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97 (2012 November).

[5] W. Xiong, L. Wu, F. Alleva, J. Droppo, X. Huang, A. Stolcke, "The Microsoft 2017 conversational speech recognition system," (2017 August), eprint arXiv:1708.06073v2.

[6] G. Anders, "Alexa, understand me," http://bit.ly/2x2Yx5F, accessed: 2018-05-22.

[7] Y. Leviathan, Y. Matias, "Google duplex: An AI system for accomplishing real-world tasks over the phone," http://bit.ly/2keb5Om, accessed: 2018-05-22.

[8] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, V. Zue, "Timit acoustic-phonetic continuous speech corpus LDC93S1," Philadelphia (1993), linguistic Data Consortium.

[9] A. Rousseau, P. Deléglise, Y. Estève, "Enhancing the TED-LIUM corpus with selected data for language modeling and more TED talks," presented at the *International Conference on Language Resources and Evaluation*, pp. 3935–3939 (2014 May).

[10] V. Panayotov, G. Chen, D. Povey, S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," presented at the *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5206–5210 (2015 April).

[11] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, E. Elsen, J. Engel, L. Fan, C. Fougner, A. Y. Hannun, B. Jun, T. Han, P. LeGresley, X. Li, L. Lin, S. Narang, A. Y. Ng, S. Ozair, R. Prenger, S. Qian, J. Raiman, S. Satheesh, D. Seetapun, S. Sengupta, C. Wang, Y. Wang, Z. Wang, B. Xiao, Y. Xie, D. Yogatama, J. Zhan, Z. Zhu, "Deep speech 2: end-to-end speech recognition in English and Mandarin," presented at the *International Conference on Machine Learning*, vol. 48, pp. 1–10 (2016 June).

[12] I. M. Quintanilha, *End-to-end speech recognition applied to Brazilian Portuguese using deep learning*, M.Sc. dissertation, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil (2017).

[13] A. Graves, A.-r. Mohamed, G. E. Hinton, "Speech recognition with deep recurrent neural networks,"

presented at the *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6645–6649 (2013 May).

[14] Y. Miao, M. Gowayyed, F. Metze, "EESEN: end-to-end speech recognition using deep RNN models and WFST-based decoding," presented at the *IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 167–174 (2015 December).

[15] A. Graves, S. Fernández, F. J. Gomez, J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," presented at the *International Conference on Machine Learning*, pp. 369–376 (2006 June).

[16] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, Y. Bengio, "Attention-based models for speech recognition," presented at the *Advances in Neural Information Processing Systems*, pp. 577–585 (2015 December).

[17] K. Veselỳ, A. Ghoshal, L. Burget, D. Povey, "Sequence-discriminative training of deep neural networks," presented at the *Interspeech*, pp. 2345–2349 (2013 August).

[18] K. J. Han, A. Chandrashekaran, J. Kim, I. R. Lane, "The Capio 2017 conversational speech recognition system," (2018 April), eprint arXiv:1801.00059v2.

[19] NIST Multimodal Information Group, "1997 hub5 English evaluation LDC2002S23," Philadelphia (2002), linguistic Data Consortium.

[20] S. J. Pan, Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359 (2010 October).

[21] J. Yosinski, J. Clune, Y. Bengio, H. Lipson, "How transferable are features in deep neural networks?" presented at the *Advances in Neural Information Processing Systems*, pp. 3320–3328 (2014 December).

[22] D. Wang, T. F. Zheng, "Transfer learning for speech and language processing," presented at the *Asia-Pacific Signal and Information Processing Association*, pp. 1225–1237 (2015 December).

[23] J. Kunze, L. Kirsch, I. Kurenkov, A. Krug, J. Johannsmeier, S. Stober, "Transfer learning for speech recognition on a budget," presented at the *Workshop on Representation Learning for NLP*, pp. 1–10 (2018 August).

[24] R. Collobert, C. Puhrsch, G. Synnaeve, "Wav2Letter: An end-to-end convnet-based speech recognition system," (2016 September), eprint arXiv:1609.03193.

[25] S. Hochreiter, J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780 (1997 November).

[26] S. Ioffe, C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," presented at the *International Conference on Machine Learning*, vol. 37, pp. 1–9 (2015 July).

[27] C. Laurent, G. Pereyra, P. Brakel, Y. Zhang, Y. Bengio, "Batch normalized recurrent neural networks," presented at the *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2657–2661 (2016 March).

[28] A. Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*, Studies in Computational Intelligence (Springer Verlag, Heidelberg, Germany) (2012).

[29] T. Mikolov, K. Chen, G. Corrado, J. Dean, "Efficient estimation of word representations in vector space," (2013 September), eprint arXiv:1301.37813.

[30] E. S. Olivas, J. D. M. Guerrero, M. M. Sober, J. R. M. Benedito, A. J. S. Lopez, *Handbook of research nn machine learning applications and trends: Algorithms, methods and techniques - 2 volumes* (Information Science Reference, Hershey, EUA) (2009).

[31] "A PaddlePaddle implementation of DeepSpeech2 architecture for ASR," https://github.com/PaddlePaddle/DeepSpeech, accessed: 2017-05-23.

[32] S. Naren, "Speech recognition using DeepSpeech2," https://github.com/SeanNaren/deepspeech.pytorch, accessed: 2017-05-23.