# Towards an end-to-end speech recognizer for Portuguese using deep neural networks

Igor Macedo Quintanilha, Luiz Wagner Pereira Biscainho and Sergio Lima Netto

***Abstract*— This paper presents an open-source character-based end-to-end speech recognition system for Brazilian Portuguese (PT-BR). The first step of the work was the development of a PT-BR dataset—an ensemble of 4 previous datasets (of which 3 publicly available). The model trained on this dataset is a bidirectional long short-term memory network using connectionist temporal classification for end-to-end training. Several tests were conducted to find the best set of hyperparameters. Without a language model, the system achieves a label error rate of 31.53% on the test set, about 17% higher than commercial systems with a language model. This first effort shows that an all-neural high-performance speech recognition system for PT-BR is feasible.**

***Keywords*— deep learning; speech recognition; recurrent networks; connectionist temporal classification**

## I. Introduction

A big technological breakthrough in automatic speech recognition (ASR) occurred by the end of the 1960s with the development of hidden Markov models (HMM), which enabled the combination of acoustic, language, and lexicon models into one probabilistic algorithm. In the next decades, this became the core of every high-performance ASR system.

From the first commercial ASR solutions of the 1990s until 2012, continued research on HMM-based algorithms brought few significant advances. The increasing computational power (e.g. powerful GPUs), the availability of huge amounts of data, and the (re)discovery of deep learning [1] made possible the development of a new system [2] that improved the state-of-the-art performance in over $30\%$. After that, ASR got back to the spotlight, leading to hundreds of papers in this area.

An HMM-based ASR system involves a complex pipeline and is expertise-intensive: dictionaries, phonetic issues, segmented data, Gaussian mixture models to obtain initial frame-level labels, multiple stages with different feature processing techniques, and an expert to determine the optimal configurations of many hyperparameters. Furthermore, the objective function used to train the network is quite distant from the true performance measure (sequence-level transcription accuracy), and until recently this paradigm had not been broken.

Ultimately, there is a hype among researchers about applying only one neural-based system to perform speech recognition in an end-to-end fashion, avoiding the complex tasks listed above. Graves *et al.* [3] developed the first successful algorithm to that end: the connectionist temporal classification (CTC), whose main appeal is having been specifically designed for temporal classification tasks, *i.e.*, sequence labeling problems where the alignment between inputs and targets is unknown. It does not require pre-segmented training data, or external post-processing to extract the label sequence from the network outputs. Since then, CTC has been extensively used in end-to-end speech recognition systems, and was even adopted by Google as the default algorithm in Google Voice Search [4]. More recently, encoder-decoder attention-based recurrent neural networks (RNN) have been successfully applied to speech recognition [5]. Although promising, the latter are not investigated in this work.

The goal of this paper is to present the first open-source[1] character-based end-to-end speech recognition system for Brazilian Portuguese the authors have knowledge of. Although recent work [6] has been done on directly transcribing raw speech waveforms, this approach is not investigated here. Instead, an MFCC (Mel-Frequency Cepstrum Coefficients) feature extraction stage is the minimal preprocessing block.

The MFCC features are then processed by a bidirectional long short-term memory (BLSTM) block, followed by a softmax layer and then by the CTC method. Dense layers were avoided, because they add many learnable parameters, boosting the overfitting problem. The network is trained at a text-transcription level, avoiding the need for a pronunciation dictionary. It is worth mentioning that since the CTC is being used, no frame-level alignment is necessary.

This work combines elements of [3] (CTC), [7] (topology), and [8] (character-based recognition); however, it introduces and develops over a new Brazilian Portuguese dataset that is a pre-processed ensemble of 4 smaller datasets.

After this introduction, the paper is organized as follows. Sec. II describes the recurrent network architecture used in this work, and Sec. III describes the CTC method. Sec. IV introduces the Brazilian Portuguese speech dataset (BRSD). Sec. V details the proposed model. Sec. VI develops an experiment with the recurrent models and the BRSD, and analyzes the network's transcription results. Finally, in Sec. VII, conclusions and possible future directions are drawn.

## II. Network architecture

A simple multilayer perceptron (MLP) does not handle well temporal sequences at its input/output, since it is tailored to model a function that statically transforms input data into a desired output. This limits its applicability in cases where the inputs are part of a sequence whose ordering conveys some intercorrelation among them and, most important, among their corresponding outputs. One way to deal with these correlations

Igor Quintanilha, Luiz Biscainho and Sergio Netto are with the Federal University of Rio de Janeiro (UFRJ), Rio de Janeiro, Brazil. E-mails: {igor.quintanilha, wagner, sergioln}@smt.ufrj.br.

[1]Code available at `http://github.com/igormq/sbrt2017`

is using the long short-term memory [3] (LSTM), a special kind of RNN. The standard LSTM has two main parts: a function that at each time step $t$ maps the input $\boldsymbol{x}^{(t)} \in \mathbb{R}^D$ and the previous state $\boldsymbol{h}^{(t-1)}$ into the current state $\boldsymbol{h}^{(t)} \in \mathbb{R}^H$; and another that maps the current state into the output $\boldsymbol{z}^{(t)} \in \mathbb{R}^{K+1}$. The LSTM also includes a cell state $\boldsymbol{c}$, responsible for propagating the information through the time steps with minor modifications. Mathematically:

$$\boldsymbol{a} = \boldsymbol{W}_x \boldsymbol{x}^{(t)} + \boldsymbol{W}_h \boldsymbol{h}^{(t-1)} + \boldsymbol{b} = [\boldsymbol{a}_i; \boldsymbol{a}_f; \boldsymbol{a}_o; \boldsymbol{a}_g], \quad (1)$$

$$\boldsymbol{i} = \sigma(\boldsymbol{a}_i), \quad \boldsymbol{f} = \sigma(\boldsymbol{a}_f), \quad \boldsymbol{o} = \sigma(\boldsymbol{a}_o), \quad \boldsymbol{g} = \tanh(\boldsymbol{a}_g), \quad (2)$$

$$\boldsymbol{c}^{(t)} = \boldsymbol{f} \odot \boldsymbol{c}^{(t-1)} + \boldsymbol{i} \odot \boldsymbol{g}, \quad \boldsymbol{h}^{(t)} = \boldsymbol{o} \odot \tanh(\boldsymbol{c}^{(t)}), \quad (3)$$

where $\boldsymbol{W}_x \in \mathbb{R}^{4H \times D}$ and $\boldsymbol{W}_h \in \mathbb{R}^{4H \times 4H}$ contain the weight gates; $\boldsymbol{b}$ is the bias; $\boldsymbol{i} \in \mathbb{R}^H$, $\boldsymbol{f} \in \mathbb{R}^H$, $\boldsymbol{o} \in \mathbb{R}^H$, and $\boldsymbol{g} \in \mathbb{R}^H$ are the input, forget, output, and input block gates, respectively; $\sigma(\cdot)$ is the sigmoid function; both $\sigma(\cdot)$ and the hyperbolic tangent are applied element-wise; and $\odot$ is an element-wise product.

Recurrent networks only correlate samples up to time step $t$, *i.e.*, rely only on past information. In speech recognition, due to the co-articulation effect and even linguistics, the correct prediction of the current label (*e.g.* character, word, phoneme) may rely also on the next labels. Bidirectional RNNs (BRNNs) address that issue [9] by combining two RNNs, one moving forward and another backward through time. The output at each time step is the concatenation of their hidden states:

$$\boldsymbol{z}^{(t)} = \boldsymbol{W}_{hz}^T [\vec{\boldsymbol{h}}^{(t)}; \overleftarrow{\boldsymbol{h}}^{(t)}] + \boldsymbol{b}_z, \quad (4)$$

where $\boldsymbol{W}_{hz} \in \mathbb{R}^{K+1 \times 2H}$, $\boldsymbol{b}_z \in \mathbb{R}^{2H}$, and $\vec{\boldsymbol{h}}^{(t)}$ and $\overleftarrow{\boldsymbol{h}}^{(t)}$ are the forward and backward hidden state variables, respectively.

In supervised training, where inputs $\boldsymbol{x}$ and desired outputs $\boldsymbol{y}$ are available, one must define a scalar loss function $\mathcal{L}$ evaluated over mini-batches of a training set $\mathbb{S}$ to quantify the error between network predictions $\hat{\boldsymbol{y}}$ and desired outputs. These errors are backpropagated through the network to recursively compute the gradients of the loss function w.r.t. each network parameter. Due to the nonlinearity of the model, a gradient-based (*e.g.* SGD [10], and Adam [11]) method is preferred to update the parameters. The iteration of gradient calculation and parameters update constitutes the training procedure. Training stops when there is no improvement over a validation dataset the network has not been trained on, which aids the adjustment of non-trainable parameters (*e.g.* number of hidden units $H$, learning rate) so as to ensure generalization. The resulting model is evaluated over the unseen inputs of a test set.

## III. Connectionist Temporal Classification

The label sequence of an utterance $\boldsymbol{X} = (\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(T)})$ is denoted as $\boldsymbol{l} = (l_1, \ldots, l_U)$, with blank label $\varnothing$ indexed as 0. Therefore, $l_u$ is an integer ranging from 1 to $K$. The length of $\boldsymbol{l}$ is constrained to be no greater than the length of the utterance, *i.e.*, $U \leq T$. CTC [3] aims at maximizing $\log \mathrm{P}(\boldsymbol{l}|\boldsymbol{X})$, the log-likelihood of the label sequence given the inputs, by optimizing the model parameters.

The last layer is softmax: $\hat{\boldsymbol{y}}^{(t)} = \exp(\boldsymbol{z}^{(t)}) / \sum_k e^{z_k^{(t)}}$, where exponentiation is taken element-wise and the $K+1$ elements in $\boldsymbol{z}^{(t)}$ correspond to each possible label (including $\varnothing$). At each frame $t$, the network outputs a vector $\hat{\boldsymbol{y}}^{(t)}$ whose $k$-th element $y_k^{(t)}$ is the posterior probability of label $k$.

If the output probabilities at each time step are assumed to be independent given $\boldsymbol{X}$, then $\mathrm{P}(\boldsymbol{p}|\boldsymbol{X}) = \prod_{t=1}^{T} \hat{y}_{p_t}^{(t)}$, where $\boldsymbol{p} = (p_1, \ldots, p_T)$ is the CTC path, a sequence of labels at frame level that differs from $\boldsymbol{l}$ for allowing blank label occurrences and repeated non-blank labels. Distinct CTC paths may correspond to the same label sequence, *e.g.*, both "A A$\varnothing$BC$\varnothing$" and "$\varnothing$AAB$\varnothing$CC" are mapped to "ABC". Considering the set of CTC paths for $\boldsymbol{l}$ as $\Phi(\boldsymbol{l})$, the likelihood of $\boldsymbol{l}$ is $\mathrm{P}(\boldsymbol{l}|\boldsymbol{X}) = \sum_{\boldsymbol{p} \in \Phi(\boldsymbol{l})} \mathrm{P}(\boldsymbol{p}|\boldsymbol{X})$. The CTC is able to use unsegmented data exactly for possibly mapping different paths into the same label sequence, which allows the network to predict the labels without knowing precisely when they occur.

Unfortunately, summing over all CTC paths is computationally impractical. A solution is to compactly represent the possible CTC paths as a trellis. To allow blanks in CTC paths, the blank label is added at the beginning and the end of $\boldsymbol{l}$, as well as between each two original labels in $\boldsymbol{l}$. The resulting augmented label sequence is input to a forward-backward algorithm [12] for efficient likelihood evaluation.

The CTC loss function is the colog probability of correctly labeling all training examples in training set $\mathbb{S}$:

$$\mathcal{L}(\mathbb{S}) = -\log \prod_{(\boldsymbol{X}, \boldsymbol{l}) \in \mathbb{S}} \mathrm{P}(\boldsymbol{l}|\boldsymbol{X}). \quad (5)$$

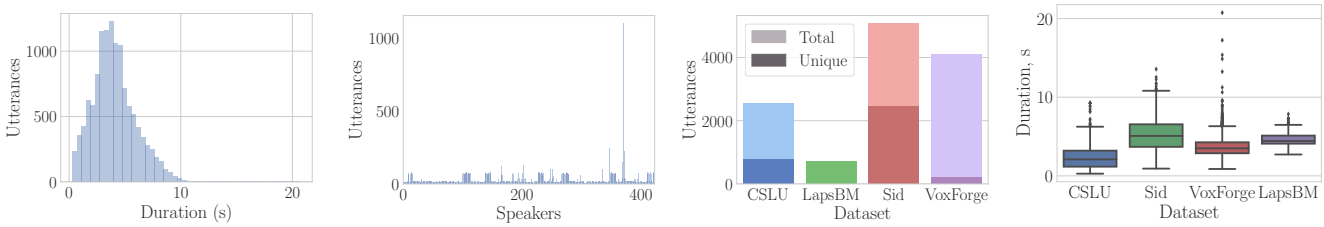
The loss $\mathcal{L}(\mathbb{S})$ is differentiable w.r.t. the network parameters and can be used by a gradient-based optimization method to find their best values. In the next section, the composed dataset from which the training set was extracted is described.

## IV. Brazilian Portuguese Speech Dataset

Building and end-to-end Portuguese speech recognition system using deep learning requires a large dataset, not readily available for free. For this reason, the Brazilian Portuguese speech dataset (BRSD) for long vocabulary continuous speech recognition (LVCSR) was built by combining 4 datasets from different origins, 3 of them freely distributed.

A) Sid dataset – Kindly provided by Dr. Sidney dos Santos for research purposes, it contains recordings by 72 speakers (20 women) from 17 to 59 years old with filed place of birth, age, genre, education, and occupation. Recorded at 22.05 kHz in non-controlled environment, its 5,777 utterances were transcribed at word level without time alignment. Contents span from spoken digits, single words, complex sequences, spelling of name and local of birth to phonetic covering and semantically unpredictable sentences. Some excerpts were discarded due to a systematic transcription error found.

B) Voxforge [13] dataset – Its intent is distributing transcribed speech audio under general public license to aid the development of acoustic models. Everyone can record and (anonymously or not) send specific utterances, which makes for the most heterogeneous *corpus*. Its Brazilian Portuguese section contains recordings by at least 111 speakers, not always with genre/age information, at varied sample rates from 16 kHz to 44.1 kHz, many with low signal-to-noise ratio (SNR). Its 4,130 utterances were transcribed at word level.




(a) Recording lengths. (b) Utterances per speaker. (c) Utterances per sub-dataset. (d) Rec. lengths per sub-dataset.

Fig. 1: Some statistics of BRSD.

TABLE I: Train, valid and test split of BRSD. "M/F" = male/female, "LL" = label length, "TD" = total duration.

| Dataset | Speakers (M/F) | Utterances (unique) | LL (min/max) | TD (hours) |
|---|---|---|---|---|
| Train | 390 (150/29) | 11,702 (3,437) | 2/149 | 13.01 |
| Valid | 21 (14/7) | 420 (420) | 36/95 | 0.55 |
| Test | 14 (11/3) | 280 (280) | 39/87 | 0.35 |

TABLE II: Commercial systems on BRSD (02/15/2016).

| System | LER | WER |
|---|---|---|
| Google API | 10.25% | 27.83% |
| IBM Watson | 11.38% | 35.61% |
| Microsoft Bing Speech | 14.77% | 40.84% |

C) LapsBM1.4 [14] dataset – The Fala Brasil group of Federal University of Pará uses this *corpus* to evaluate LVCSR systems in PT-BR. It contains recordings of 20 unique utterances by each of 35 speakers (10 women), totaling 700 excerpts, at 22.05 kHz sample rate without environment control.

D) CSLU: Spoltech Brazilian Portuguese dataset version 1.0 [15] – Distributed by Linguistic Data Consortium (LDC) under catalog number LDC2006S16, the *corpus* includes recordings by 477 speakers from several regions in Brazil in either reading speech (for phonetic coverage) or (spontaneous) responses to questions. Of its 8,080 utterances recorded at 44.1 kHz sample rate in non-controlled environment, 2,540 have been transcribed at word level alignment, and 5,479 at phoneme level with time alignment. As pointed in [16], some audio recordings have lacking or erroneous transcriptions.

Three distinct sets must be defined: train, validation, and test. Since the LapsBM has been created to evaluate LVCSR systems, it is the natural choice to provide the test set. On the other hand, a composition of Sid, VoxForge and CSLU datasets provides a sufficiently rich train set. However, randomly separating part of the train set for validation could bias the results, since one has no control over repeated speakers/utterances. Due to its particular organization (unique utterances per speaker), the LapsBM dataset provided the best solution: it was randomly partitioned into a validation set with 21 speakers (7 women) and a test set with the remaining 14.

All datasets were pruned of samples with no/wrong transcription, too short recordings, and other defects that could produce misleading results. All audio files were resampled at 16 kHz. Recording lengths concentrate around 3 s but can reach 25 s, as seen in Fig. 1a. The number of utterances per speaker is shown in Fig. 1b; the highest peak refers to the anonymous contributions in the VoxForge dataset. In Figs. 1c and 1d, one sees the relation unique/total utterances and the utterance duration distribution per sub-dataset, respectively. A summary of the dataset partitions is shown in Tab. I.

In Tab. II, the test set of BRSD was evaluated against the three major commercial systems with public API. Google API has the best performance on both label error rate (LER) and word error rate (WER) [17]. The LER span is $4.5\%$, while the WER span is $13.0\%$—indicating that using a proper language model can make a huge difference in real ASR systems.

The most important datasets employed in the ASR literature are the 5.4-h TIMIT [18], the 73-h Wall Street Journal [19], [20](WSJ), and the 300-h Switchboard [21], all in English. For its non-controlled environmental conditions, multiplicity of acquiring hardware, and distinct speaker dialects, the 14-h BRSD is far more stringent than TIMIT and (in spite of its moderate length) even WSJ. On the other hand, Switchboard contains conversational speech, not found in BRSD.

## V. Proposed Model

The proposed architecture consists of a BLSTM layer with $H$ (determined in Sec. VI-B) hidden units, followed by a softmax layer and by the CTC loss function [22].

The input was preprocessed in window frames of 25 ms and hops of 10 ms. From a bank of 40 log-filters in mel-scale, 12 MFCCs and the log energy were computed; concatenated with their first and second differences (*i.e.* delta / double delta coefficients), they form a vector of 39 features.

The model outputs probabilities of character emission instead of phonemes. Thus, the label set is $\{\mathrm{a}, \ldots, \mathrm{z}, \mathrm{space}, \varnothing\}$, where "space" delimits word boundaries. The target sequences in the dataset were simplified to rely only on this reduced label set, i.e. punctuation, hyphens, and stress marks were removed/mapped to standard labels (*e.g.* á→a, ç→c). Finally, the output sequence was provided by a greedy decoder [3] for a fast evaluation in the train and validation sets. For the test set, a width-400 beam search decoder [8] was employed.

The recurrent weights were initialized with orthogonal matrices sampled from a normal distribution [23], which helps to alleviate the vanishing gradient problem [24] over long time steps. The non-recurrent weights were initialized from a uniform distribution using Xavier method [25], default in deep learning architectures. Also, the forget gate bias was set to one, which improves the LSTM performance [26].

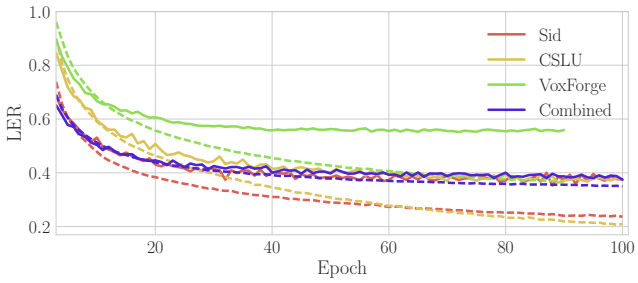



Fig. 2: Performance×train set (dashed=train, solid=validation).

TABLE III: Model performance *versus* network capacity.

| $H$ | # parameters | validation | test |
|---|---|---|---|
| 128 | 179k | 43.17% | 39.67% |
| 256 | 620k | 41.50% | 39.34% |
| 512 | 2.3M | 39.68% | 37.28% |
| 1024 | 8.8M | **38.63%** | **36.08%** |

The training was carried out with Adam [11] and learning rate of $10^{-3}$ (carefully chosen), unless stated otherwise over 30 epochs (due to time constraints), with a batch size of 32. A weight decay [1] of $10^{-4}$ was applied to all network parameters as a default regularizer. Dropout and variational dropout regularizers were also investigated (in Sec. VI-C).

## VI. Experiments and Analysis

In this section, we report three experiments: '1' investigates whether the BRSD ensemble can generalize better than its subsets; '2' searches for the best value of the number of hidden units $H$; '3' investigates the use of more advanced regularization methods as complements to weight decay. Finally, the overall network's transcription results are analyzed.

### A. Experiment 1

The effects of using the overall train set or each of its subsets were compared in a network with $H = 128$ hidden units. Fig. 2 shows train/validation evolution for each case. One sees that each subset exhibits a large bias between validation and training values—indicating ill generalization. Yet the entire dataset yields the best validation behavior and the lowest bias w.r.t training; moreover, training LER does not lower towards zero, suggesting that the model capacity should be increased.

### B. Experiment 2

The model capacity was expanded by increasing the number of hidden units. From Tab. III, one clearly infers that more hidden units are advantageous: the label error rate (LER) drops from $43.17\%$ to $38.62\%$ in the validation set as the number of units increases from 128 to 1024. Their number was not further increased due to computational power limitations.

### C. Experiment 3

Dropout [27] is an efficient method for preventing overfitting and regularizing the network, and it has been widely used in convolutional neural networks [28], [29]. Applying dropout to RNN, however, is not straightforward [30]. Proposed by Gal *et al.* [31], variational dropout is a new mathematical approach developed to this end. While previous methods only applied dropout to the non-recurrent weights, Gal *et al.* also exploited their application to the recurrent weights.

TABLE IV: Dropout-based regularizers: LER (dropout rate).

| H | validation set | | test set | |
|---|---|---|---|---|
| | dropout | var dropout | dropout | var dropout |
| 128 | 42.07% (20%) | **40.70% (10%)** | 38.43% | **37.64%** |
| 1024 | | 35.16% (20%) | | 32.66 |

Tab. IV shows the results of dropout and variational dropout with $H = 128$ hidden units. For $H = 1024$, only the results with variational dropout are available due to memory restrictions—traditional dropout calculates a different mask for each time step, consuming a lot of memory. Using dropout has its advantages, reducing absolute LER in 3% of when enabled. Variational dropout outperformed dropout in all setups.

### D. Complete model

After several design choices, the best model has a BLSTM with $H = 1024$ hidden units and employs as regularizers variational dropout with dropout probability of $20\%$ and weight decay of $10^{-4}$. It was trained for more than 30 epochs until it overfitted. The best validation result occurred in epoch 41, with an LER of $34.15\%$, while in the test set an LER of $31.53\%$ was found. The difference between the rates is due to the beam search decoder applied to the test set. These results are close to that obtained by Graves *et al.* [7] with almost the same number of parameters, but roughly $17\%$ higher when compared to commercial systems (Tab. II), which are far more complex and take advantage of lexicon and language models.

### E. Transcription analysis

Some errors that occur in the transcription seem phonetically justifiable, as pointed in [7]. In Tab. V (a), the network changed "flexa" (Lucia's mid name) into "flecha" (arrow), which have the same sound. This misspelling might have occurred because the character sequence "cha" has a higher number of occurrences in the training set than "xa".

The network also made some "mistakes" for transcribing some speakers' peculiarities, depicted in Tab. V (b). While the ground truth is "tem se" the network outputs "ten ci", which sounds equal. It seems that the network transcribed the sentence the way it was spoken.

At first sight, the rationales behind network mistakes in Tab.V (a) and (b), both associated to phonetically equivalent sounds, may seem contradictory. Nonetheless, the sequence "nci", ignoring the space character, has a higher occurrence rate than "mse" in Portuguese, which can facilitate this wrong transcription. One way to overcome the language's multiplicity is increasing the training set, thus allowing the network to internally learn how to spell correctly.

Finally, an interesting fact is demonstrated in Tab. V (c). The model changed one letter in the word "e**x**plicacoes"

TABLE V: Examples of network mispellings – T = ground truth sequence, M = sequence transcribed by the best model.

| | | |
|---|---|---|
| a | T | esta instalado na casa do avo de lucia fle**x**a de lima |
| | M | esta estalado na casa do arode duscia fle**ch**a dima |
| b | T | **tem se** uma receita mensal de trezentos e quarenta mil dolares |
| | M | **ten ci** uma receira mensalbe trezentos e quarenta mil bolarte |
| c | T | ele podia dar e**x**plicacoes praticas para sua preferencia por faroestes |
| | M | ele putiadar e**s**plicacoes cratifos para soubre ferencia por faraeste |

(explanation), replacing "x" by "s". This substitution could be explained due to regional dialects. People from Rio de Janeiro, for example, pronounce the "x" in "explicacoes" like "sh" as in /leash/, while people from other regions tend to pronounce it as a sibilant "s" as in /juice/. Indeed, listening to the dataset recordings, one can notice that the majority of the speakers are not from Rio de Janeiro, which explains the behavior of the network. All in all, misspellings, like in Tab. V (b) and (c), could be easily corrected with the use of a language model.

## VII. Conclusions and future works

After enough training of the best model, an LER of $34.15\%$ was achieved in the validation set. In the test set, the sequences were decoded by a beam search decoder with a beam width of 400 and no language model, achieving an LER of $31.53\%$, which is comparable to the results of Graves *et al.* [7] and Maas *et al.* [8]; however, much improvement is needed to achieve results similar to those of commercial systems.

Stacking more layers is crucial to improve model's accuracy, as shown in [32]; unfortunately, this path could not be taken due to insufficient computational power. Using a language model is an essential tool for any reliable ASR system, and none has been applied yet. Further investigation of stacking layers and different language models (*e.g.* RNN-based [33], WFST [34]) are the next steps in this work.

## Acknowledgements

This work was partially funded by FAPERJ and CNPq Brazilian agencies.

## References

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[2] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. Audio, Speech, Language Process.*, vol. 20, no. 1, pp. 30–42, 2012.

[3] A. Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*. Springer Verlag, 2012.

[4] "Google voice search: Faster and more accurate." `https://research.googleblog.com/2015/09/google-voice-search-faster-and-more.html`. Accessed: 2017-03-02.

[5] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. NIPS*, (Montreal), pp. 577–585, 2015.

[6] D. Palaz, M. Magimai-Doss, and R. Collobert, "Analysis of CNN-based speech recognition system using raw speech as input," in *Annual Conf. of the Int. Speech Communication Association*, (Dresden), pp. 11–15, 2015.

[7] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. ICML*, vol. 32, (Beijing), pp. 1–9, 2014.

[8] A. L. Maas, Z. Xie, D. Jurafsky, and A. Y. Ng, "Lexicon-free conversational speech recognition with neural networks," in *Annual Conf. NAACL-HLT*, (Denver), pp. 345–354, 2015.

[9] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, 1997.

[10] C. M. Bishop, *Pattern Recognition and Machine Learning*. Information Science and Statistics, Springer Verlag, 2006.

[11] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, (San Diego), pp. 1–15, 2015.

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

[13] "Voxforge." `https://http://www.voxforge.org`. Accessed: 2017-03-06.

[14] "Falabrasil - ufpa." `http://www.laps.ufpa.br/falabrasil/`. Accessed: 2017-03-06.

[15] M. Schramm, L. F. Freitas, A. Zanuz, and D. Barone, "CSLU: Spoltech Brazilian Portuguese version 1.0 LDC2006S16." Philadelphia, 2006. Linguistic Data Consortium.

[16] N. Neto, P. Silva, A. Klautau, and A. Adami, "Spoltech and OGI-22 baseline systems for speech recognition in Brazilian Portuguese," in *Int. Conf. on Computational Process. Portuguese Language*, vol. 5190, (Aveiro), pp. 256–259, 2008.

[17] D. Jurafsky and J. H. Martin, *Speech and Language Processing*. Prentice Hall, 2nd ed., 2014.

[18] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, and V. Zue, "Timit acoustic-phonetic continuous speech corpus ldc93s1." Philadelphia, 1993. Linguistic Data Consortium.

[19] J. Garofolo, D. Graff, D. Paul, and D. Pallett, "CSR-I (WSJ0) Sennheiser LDC93S6B." Philadelphia, 1993. Linguistic Data Consortium.

[20] "CSR-II (WSJ1) Sennheiser LDC94S13B." Philadelphia, 1994. Linguistic Data Consortium.

[21] J. Godfrey and E. Holliman, "Switchboard-1 release 2 LDC97S62." Philadelphia, 1993. Linguistic Data Consortium.

[22] I. M. Quintanilha, "End-to-End Speech Recognition Applied to Brazilian Portuguese Using Deep Learning," MSc dissertation, PEE/COPPE, Federal University of Rio de Janeiro, Rio de Janeiro, Brazil, March 2017.

[23] A. M. Saxe, J. L. McClelland, and S. Ganguli, "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks," in *Proc. ICLR*, (San Diego), pp. 1–22, 2014.

[24] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. ICML*, vol. 28, (Atlanta), pp. 1310–1318, 2013.

[25] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Int. Conf. AISTATS*, vol. 9, (Sardinia), pp. 249–256, 2010.

[26] R. Józefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *Proc. ICML*, vol. 37, (Lille), pp. 1–9, 2015.

[27] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *JMLR*, vol. 15, pp. 1929–1958, 2014.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. NIPS*, (Lake Tahoe), pp. 1097–1105, 2012.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE ICCV*, (Santiago), pp. 1026–1034, 2015.

[30] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," 2014. eprint arXiv:1409.2329v5.

[31] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," in *Proc. NIPS*, (Long Beach), pp. 1019–1027, 2016.

[32] A. Graves, A.-R. Mohamed, and G. E. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE ICASSP*, (Vancouver), pp. 6645–6649, 2013.

[33] T. Mikolov, S. Kombrink, L. Burget, J. Cernocký, and S. Khudanpur, "Extensions of recurrent neural network language model," in *Proc. IEEE ICASSP*, (Prague), pp. 5528–5531, 2011.

[34] Y. Miao, M. Gowayyed, and F. Metze, "EESEN: end-to-end speech recognition using deep RNN models and WFST-based decoding," in *Proc. IEEE Workshop ASRU*, (Scottsdale), pp. 167–174, 2015.