DOI: 10.17469/O2104AISV000012

CECILIA DI NARDI, ROSANNA TURRISI, ALBERTO INUGGI, NILO RIVA, ILARIA MAURI, LEONARDO BADINO

# An automatic speech recognition Android app for ALS patients

This paper describes AllSpeak, an Automatic Speech Recognition (ASR) Android Application developed for Italian-speaking patients with Amyotrophic Lateral Sclerosis (ALS). It allows to recognize a predefined and customizable set of basic utterances that are used by the patient in everyday life (e.g., "I'm thirsty", "I feel pain", etc...). The ASR engine is based on deep learning architectures and it uses a simple decoding strategy to allow offline (i.e., w/o any network connection) and fast decoding. Although deep learning approaches have achieved outstanding results on different speech recognition tasks, recognition of impaired speech is still quite challenging for an ASR system mainly due to a scarce availability of training data and a large variability of impairments. We have addressed these two problems by limiting recognition to a set of key phrases/words corresponding to the patient's primary needs and by strongly adapting the neural networks to the target speaker's voice. Results show that the type of network architecture and the training strategy have both a very significant impact on recognition accuracy of dysarthric speech. Although different architectures and training strategies perform similarly on healthy speakers, recurrent neural networks trained in sequence-to-sequence fashion significantly outperform any other method on most of ALS speakers.

*Keywords:* automatic speech recognition, amyotrophic lateral sclerosis, smartphone application, deep neural networks.

## 1. *Introduction*

Amyotrophic lateral sclerosis (ALS) is characterized by progressive muscle paralysis caused by degeneration of motor neurons in the primary motor cortex, corticospinal tracts, brainstem, and spinal cord (Van Es et al., 2017). ALS is relentlessly progressive – 50% of patients die within 30 months of symptom onset and about 20% of patients survive between 5 years and 10 years after symptom onset (Talbot, 2009). Modern technology has allowed people with ALS to compensate to some degree for almost every loss of function, making it possible even for those with almost no muscle function to continue to breathe, communicate, eat, travel and use a computer. In particular, for many people with ALS, the speaking ability may be lost as weakness increases in the muscles of the mouth and throat that control speech and in the muscles that help generate the pressure that moves air over the vocal folds. Dysarthria is indeed the presenting symptom in 30% of patients with ALS and is found in >80% of patients (Hardiman, 2017) and this loss of communication

prevents patients from participating in many activities and may lead to social isolation, reducing the quality of life (QoL).

The goal of management of dysarthria in ALS patients is to optimize communication effectiveness for as long as possible. Speech therapy can delay the progression of dysarthria, and augmentative and alternative communication techniques are the treatments of choice and can enhance QoL in the most advanced phases of ALS. Nevertheless, although there have been several attempts to improve speech recognition for dysarthric speakers as communication techniques based on brain–computer interfaces, these efforts have not until recently converged and their use in the clinical setting is still limited. Moreover, modern automatic speech recognition (ASR) is ineffective at understanding relatively unintelligible speech caused by dysarthria and traditional representations in ASR such as Hidden Markov models (HMMs) trained for speaker independence that achieve 84.8% word-level accuracy for non-dysarthric speakers might achieve less than 4.5% accuracy given severely dysarthric speech on short sentences (Rudzicz, 2010a; Rudzicz, 2010b; Rudzicz, 2012). Recently, more accurate dysarthric speech recognition system has been developed by using deep learning based approaches (España-Bonet, Fonollosa, 2016; Joy, Umesh, Abraham, 2017; Vachhani, Bhat, Das & Kopparapu, 2017). However, in case of severe disability, the ASR performance still remains poor. Causes of poor performance may include slurred speech, weak or imprecise articulatory contacts, weak respiratory support, low volume, incoordination of the respiratory stream, hypernasality, and reduced intelligibility (Kim, Kent & Weismer, 2011). Additionally, dysarthric speech is not sufficiently covered in the training datasets of state-of-the-art commercial ASR systems.

As a result, dysarthria can *have* dramatic consequences for speech intelligibility among artificial listeners – that is, speech recognition systems. In some preliminary experiments we have carried out on the TORGO dataset (Rudzicz, Namasivayam & Wolff, 2012), Google Speech API and IBM speech-to-text could misrecognize more than the 80% of words in single word utterances.

This paper describes AllSpeak, an Automatic Speech Recognition (ASR) Android Application specifically developed for patients with ALS. It allows patients to communicate, through their residual speech abilities, their basic needs to their families and caregivers.

## 2. *The AllSpeak App*

The AllSpeak App is a hybrid App developed with the Ionic 1.X framework for the applicationAndroid 6.0 platform. All the speech processing and recognition modules are implemented within a custom multi-threaded Cordova Plugin. The latter is composed by the following modules, each running on its own independent thread:

- audio acquisition (INPUT);
- voice activity detection (VAD);
- spectral features extraction (FE);
- speech command recognition, mainly based on Tensorflow neural networks (SR);

Once recognition is activated, these four processes run in parallel.

The INPUT module extracts speech from the smartphone's microphone and sends it to the VAD module.

When the VAD module recognizes speech activity, it sends the extracted speech segments to the FE module that calculates the spectral features and, once completed, sends concatenated feature vectors to the SR module.

The VAD module sends a detected speech segment to the FE module only if its duration is longer than a predefined threshold (500 ms in our case). The sent segment also contains a non-speech "tail", i.e., up to 400 ms long "active samples" after the last speech sample identified as speech. Then the resulting segment is considered as a command and after feature extraction its associated verbal command will be inferred by the SR module. The SR module consists in a simple speech decoder and runs preloaded Tensorflow deep neural networks.

This four-thread approach optimizes the recognition process, since the to-be-inferred features are already present in the SR module when the VAD module decides that a new command has been pronounced by the App user.

## 3. *The ASR engine*

The ASR engine (the SR module of the previous section) is based on deep neural networks. The spoken command decoding is simply the classification of the input speech segment and depends on the type of neural network used.

Neural networks training have been split in two steps: speaker-independent training on a control data set (i.e. healthy speakers) and speaker-adaptation to the patient of interest. Speaker adaptation has been applied to the deep feedforward neural networks (DNNs) to compensate the mismatch between clean speech-trained model and a small set of impaired speaker's data.

The ASR can use two different types of deep neural networks: deep feedforward neural networks (DNNs) and deep recurrent neural networks (RNNs).

### 3.1 Feature extraction technique

Feature extraction is the main part of the speech recognition system. The goal of feature extraction is to compute a sequence of feature vectors to have a compact representation of input signal. Because every speech and speaker has different individual characteristics embedded in their speech utterances, it is better to perform feature extraction in short term interval that would reduce these variabilities. Hence, the input voice signal is examined over a short period of time where the characteristics of speech signal become stationary. In general, a speech signal contains some acoustic information which can be represented by these features. There are several feature extraction techniques, however the use of Mel Frequency Cepstral Coefficients (MFCCs) can be considered as one of the standard methods for feature extraction (Motlíček, 2003) and it is also the technique employed in our algorithm. MFCCs are the result of a cosine transform of the real logarithm of the short-term energy spectrum expressed on a mel-frequency scale.

In our algorithm, the speech signal is first divided into time frames consisting of an arbitrary number of samples. Each time frame is then windowed with 25 ms length Hamming window shifted every 10ms and for each speech frame, a set of MFCCs is computed. The number of spectral features employed in the DNN is listed below.

FBANKS (log mel-filter bank channel outputs):

- 24 FBANKS + temporal delta and acceleration coefficients (72 parameters per frame);
- 24 FBANKS + spectral delta and acceleration coefficients (72 parameters per frame);

MFCCs (mel-frequency cepstral coefficients):

- 13 MFCCs + temporal delta and acceleration coefficients (39 parameters per frame)
- 13 MFCCs + spectral delta and acceleration coefficients (39 parameters per frame)

Note that spectral deltas and acceleration coefficients are heuristic-based features we have proposed to account for the small training dataset. These features have an important impact for feedforward DNN as we will see in the result section.

### 3.2 ASR based on feedforward DNNs

Built on DNNs, the decoder simply averages the spoken command posterior probabilities that the DNN outputs at each speech frame and selects the command with the highest posterior.

The control dataset consisted of 23 commands spoken by eight healthy subjects, each command repeated from 8 to 10 times and the patient dataset comprised the same 23 commands spoken by eight ALS patients, each command repeated from 4 to 10 times – depending on patient's medical condition.

Regarding the DNN architecture and training, a three hidden layer DNN was implemented for the first training step on controls with an input layer containing 792 nodes (72 features x 11 context frames), the hidden layers with 500 nodes each and the output layer with 23 nodes, as many as the number of commands.

Once the first training step is completed, speaker adaptation comes in. We have experimented with a simple speaker-adapted layer insertion strategy consisting in adding input, output or hidden layers to the original net and then optimizing the parameters of that/those layer(s) only (see for example, Neto, Almeida, Hochberg, Martins, Nunes, Renals & T. Robinson, 1995; Gemello, Mana, Scanzio, Laface & De Mori, 2007; Li, Sim, 2010). For example, adding a first input layer should serve as "normalization" of the input, where the patient's input speech is transformed in order to closely match the input of the control training data.

As mentioned above, the DNN outputs are sentence/command posteriors:

$$(1) \quad y^* = \underset{s}{argmax} \frac{1}{T} \sum_{t=1}^{T} (p(s|x_t))$$

where $y^*$= selected sentence, $s$=sentence/command $T$= number of frames, $x_t$ = concatenated vectors at time $t$.

This very simple decoding strategy resembles the key phrase recognition strategy proposed for Google small footprint keyword spotting in Chen et al. (2014).

3.2.1 Results

Averaged results for both the acoustic features employed with 23 different recorded commands pronounced by eight ALS patients and by eight healthy controls are shown in Table 1.

Table 1 - *Average Performance (Command Error Rate)*

| | *Acoustic Features* | |
|---|---|---|
| | FBANKS | MFCC |
| *Spectral* | 17.8 % | 24.9 % |
| *Temporal* | 32.7 % | 32.7 % |

Previously showed results are primarily related to patient's vocabulary size and modality of speech (intelligible or degraded speech) depending on the extent of the disease in each patient at the time of this study. A more detailed per-speaker accuracy is displayed in Table 2, together with a Therapy Outcome Measuring (TOM) tool using a Rating Severity Scale from 0-5 to rate scores of dysarthria (0 = normal, 3 = moderate and 5 = severe).

Table 2 - *Per Speaker Command Error Rates*

| *Speaker* | *Temporal FBANKS* | *Spectral FBANKS* | *Temporal MFCC* | *Spectral MFCC* | *TOM* |
|---|---|---|---|---|---|
| BB | 60.9 % | 21.7 % | 47.8 % | 39.1 % | 1 |
| DAD | 4.3 % | 0.0 % | 0.0 % | 0.0 % | 4 |
| DG | 21.7 % | 8.7 % | 43.5 % | 4.3 % | 2 |
| PN | 36.4 % | 13.6 % | 36.4 % | 13.6 % | 3 |
| RS | 39.1 % | 26.1 % | 30.4 % | 34.8 % | 1 |
| SE | 30.4 % | 26.1 % | 47.8 % | 52.2 % | 3 |
| TE | 54.5 % | 31.8 % | 31.8 % | 36.4 % | 3 |
| VL | 14.3 % | 14.3 % | 23.8 % | 19.0 % | 3 |

3.3 ASR based on RNNs

This section refers to the decoding strategy based on recurrent neural networks (RNNs) trained using a sequence-to-sequence approach. The sequence-to-sequence approach is not the only one we have tested (e.g., we also experimented with connectionist temporal classification) but it is the one that turned out to be the most successful. In this approach, the entire variable-length sequence of feature vec-

tors representing the speech segment is fed into the RNN that returns one single vector of posterior probabilities with one element for each command. The decoder simply selects the command with the largest posterior probability.

### 3.3.1 Architecture

RNNs have recently drawn the attention of researchers as they have proven to be a suitable tool to model temporal sequences. Indeed, it has been shown that RNNs can outperform feed-forward networks on large-scale speech recognition tasks (Sak et al., 2014). A recurrent neural network is a neural network that consists of a hidden state $h$ which operates on a variable-length sequence $x = (x_1, \ldots x_T)$ through a non-linear activation function $f$. In our system, the input x is a vector that represents the acoustic features and we aim at finding the most likely corresponding command $y$. At each time step t the hidden state $h_t^{\rightarrow}$ of the RNN is updated by $h_t^{\rightarrow} = f(h_{t-1}^{\rightarrow}, x_t)$. Finally, it estimates the label posterior $p(y_t \mid x_t, h_t^{\rightarrow})$. The power of RNN relies on taking into account temporal dependencies over the input sequence, either unidirectionally or bidirectionally. Unidirectional RNN estimates the label posteriors using only the left (past) context of the recurrent input, while bidirectional RNN uses separate layers for processing the input in the forward (i.e., from left to right) and backward (i.e., from right to left) directions. In the latter case, we will have $p(y_t \mid x_t, h_t^{\rightarrow}, h_t^{\leftarrow})$, where $h_t^{\leftarrow} = g(h_{t+1}^{\leftarrow}, x_t)$ for some nonlinear function $g$. The limit of RNNs is that they can capture only very short time dependencies. To overcome this problem, we looked at a particular type of recurrent neural networks: the long short-term memory (LSTM) (Hochreiter, Schmidhuber, 1997). In this work, we implemented the bidirectional LSTM (BLSTM) architecture.

Typically, in speech recognition, both recurrent and feed-forward networks are trained as frame-level classifiers. As a consequence, the alignment between audio and transcription sequences has to be determined in order to have a target for every frame. Typically, alignments are provided by a Gaussian Mixture Model – Hidden Markov Model (GMM-HMM) system trained with the Baum-Welch algorithm. However, a good alignment of impaired speech may not be feasible, and that can have catastrophic consequences on the (frame-level) training of neural networks (as labels would be very noisy). To address these issues, we trained the BLSTM as a sequence-to-sequence model (Sutskever, Vinyals & Le, 2014). This method allows to train the network by taking in input a sequence of length $T$ and giving as an output the correspondent sequence of length $T'$, where $T$ and $T'$ are not necessarily the same. In our case, the output sequence is a command and, therefore, $T' = 1$.

The underlying idea is very simple: an encoder (or reader) BLSTM processes the input sequence and emits a fixed-size context variable $C$, which represents a summary of the input sequence. A decoder (or writer) takes as input the context $C$ and generates the output sequence. Usually, the final hidden state of the encoder is used to compute $C$. In terms of probability, the sequence-to-sequence architecture maximizes the probability of the command, given the whole acoustic sequence, $p(\mathrm{y} \mid x_1, \ldots z_T)$.

### 3.3.2 Experimental setup

We evaluated the sequence-to-sequence BLSTM on the AllSpeak dataset. In particular, we tested five patients and two control speakers in order to cover the whole range of dysarthric degrees (on the TOM scale). From the speech of these speakers we extracted the adaptation data and the testing data. We only considered the temporal MFCC feature vectors, as they are the most conventional choice. For all the experiments, we used the BLSTM network with 5 hidden layers and 250 units per layer. We set the initial learning rate to be 0.01, and we exponentially decayed the learning rate by a factor of 0.7, every 3000 steps. Our model was trained to minimize the cross entropy (within the sequence-to-1 paradigm), by using the momentum optimizer with momentum equal to 0.9. We also clipped the gradient to avoid the vanishing/exploding gradient problem. Cross-validation was employed to get the best number of training epochs. To reduce the mismatch between the acoustic model and the testing speaker, we performed the speaker adaptation. More precisely, after training the network, we added a feed-forward layer atop the input. We trained the new layer, freezing the other ones, on the adaptation data. Finally, we used the testing data to measure the level performance of the model.

### 3.3.3 Results

Table 3 shows the command error rate (CER). As expected, the error is lower on the control speakers testing. Surprisingly, the sequence-to-sequence BLSTM achieves a good performance even in presence of dysarthric speech, with a minimum error of 4% on the speaker SG. In every case, the error is reduced (or remains equal) after speaker adaptation. In the best case, adaptation provides an error reduction from 71.7% to 21.7%. Note that the averaged error rates are not referred to all speakers but only to BB, PN, RS and SE. This is to compare the CERs with the ones coming from Table 2. As we can see, we obtained a CER reduction from 36.0% to 16.6%.

Table 3 - *Sequence-to-sequence BLSTM results*

| *Speaker* | *Patient/Control* | *CER (without adaptation)* | *CER (after adaptation)* | *TOM* |
|---|---|---|---|---|
| AI | Control | 7.0 % | 7.0 % | 0 |
| CD | Control | 8.0 % | 1.3 % | 0 |
| BB | Patient | 25% | 18.2 % | 1 |
| PN | Patient | 34.8 % | 13 % | 3 |
| RS | Patient | 71.7 % | 21.7 % | 1 |
| SE | Patient | 4.0 % | 4.0 % | 3 |
| SG | Patient | 44.4 % | 25.9 % | NA |
| Average | Patients | 36.0 % | 16.6 % | – |

## 4. *Conclusion*

Despite their growing presence in home computer applications and various telephony services, commercial automatic speech recognition technologies are still not easily employed by everyone, especially individuals with speech disorders. ALLSpeak is an App designed for Android equipped smartphones and tablets that allow ALS patients to go on communicating with the rest of the world, both when speaking becomes an effortful task and when their voice intelligibility almost vanishes. The first version of our algorithm running on the App was based on a DNN trained on non-dysarthric speech. This recognizer had an averaged command error rate ranging from 32.7% to 17.8% using temporal and spectral FBANKS respectively and from 32.7% to 24.9% using temporal and spectral MFCC for dysarthric speech. With the aim of improving the recognizer performance, we explored a further method: the sequence-to-sequence LSTM model. We observed that the best performance is accomplished by applying the speaker adaptation, providing an averaged command error rate of 15.0% over all 7 speakers, and 16.6% over the 5 patients. In order to compare the DNN and LSTM models, we analyzed the results related to the common tested speakers. What we found is an averaged error rate difference of 19.4%, showing that the LSTM model trained in a sequence-to-sequence fashion is a more suitable tool to address the dysarthric speech recognition. Thus, the following step will be the integration of this method to the mobile application. Our belief is that, by using our AllSpeak application, people with speech disorders will have the opportunity to participate in the technology present and experience the benefits of smartphones which are powerful devices able to mitigate their disabilities.

## *Bibliography*

Chen, G., Parada, C., Heigold, G. (2014). *Small footprint keyword spotting using deep neural networks.* In IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014, Florence, Italy, May 4-9, 4087-4091.

Espana-Bonet, C., Fonollosa, J.A.R. (2016) *Automatic Speech Recognition with Deep Neural Networks for Impaired Speech.* A: International Conference on Advances in Speech and Language Technologies for Iberian Languages. "Advances in Speech and Language Technologies for Iberian Languages: Third International Conference, IberSPEECH 2016: Lisbon, Portugal, November 23-25, 2016: proceedings". Lisbon: Springer, 97-107.

Gemello, R., Mana, F., Scanzio, S., Laface, P. & De Mori, R. (2007). *Linear hidden transformations for adaptation of hybrid ANN/HMM models*. Speech Communication, Elsevier: North-Holland, 49 (10-11), 827.

Hardiman, O. (2017). *Amyotrophic lateral sclerosis*. Nature Reviews Disease Primers, 3, 17071.

Hochreiter, S., Schimidhuber, J. (1997). *Long short-term memory*. Neural computation, 9(8), 1735-1780.

Joy, N.M., Umesh, S., Abraham, B., (2017) *On Improving Acoustic Models for TORGO Dysarthric Speech Database*. Proceedings Interspeech 2017, 2695-2699.

Kim, Y., Kent, R.D., Weismer, G. (2011). *An acoustic study of the relationships among neurologic disease, dysarthria type, and severity of dysarthria*. In *Journal of Speech, Language, and Hearing Research*, 54(2), 417-429.

Li, B. & Sim, K.C. (2010). *Comparison of discriminative input and output transformations for speaker adaptation in the hybrid NN/HMM systems*. Proceedings of the 11th Annual Conference of the International Speech Communication Association, INTERSPEECH 2010, 526-529.

Motlicek, P. (2003). *Feature Extraction in Speech Coding and Recognition, Report, Portland, to research, data, and theory*. Technical Report of PhD research internship in ASP Group, OGI-OHSU, < http://www. fit. vutbr. cz/~motlicek/publi/2002/rep ogi. pdf.

Neto, J., Almeida, L., Hochberg, M., Martins, C., Nunes, L., Renals, S. & Robinson, T. (1995). *Speaker-adaptation for hybrid HMM-ANN continuous speech recognition system*. Proceedings from Eurospeech '95: The 4th European Conference on Speech Communication and Technology, 2171-2174.

Rudzicz, F. (2010a). *Toward a noisy-channel model of dysarthria in speech recognition*. Proceeding of the NAACL HTL 2010 Workshop on Speech and Language Processing for Assistive Technologies, Association for Computational Linguistics, Los Angeles, California, 80-88.

Rudzicz, F. (2010b). *Correcting errors in speech recognition with articulatory dynamics*. Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL '10), Association for Computational Linguistics, Stroudsburg, PA, USA, 60-68.

Rudzicz, F. (2012). *Using articulatory likelihoods in the recognition of dysarthric speech*. In Speech Communication, 54, 430-444.

Rudzicz, F., Namasivayam, A.K., Wolff, T. (2012). *The TORGO database of acoustic and articulatory speech from speakers with dysarthria*. In Lang Resources & Evaluation, 46: 523-541.

Sak, H., Senior, A., Beaufays, F. (2014). *Long short-term memory recurrent neural network architectures for large scale acoustic modeling*. In Fifteenth annual conference of the International Speech Communication Association.

Sutskever, I., Vinyals, O., LE, Q.V. (2014). Sequence to sequence learning with neural networks. In Advances in Neural Information Processing Systems, 3104-3112.

Talbot, K. (2009). *Motor neuron disease: the bare essentials*. Practical neurology, 9, 303-09.

Vachhani, B., Bhat, C., Das, B., Kopparapu, S.K. (2017). *Deep auto encoder based speech features for improved dysarthric speech recognition*. In Proceedings of Interspeech 2017, 1854-1858.

Van ES, M.A., Hardiman, O., Chio, A., Al-Chalabi, A., Pasterkamp, R.J., Veldink, J.H., Van Den Berg, L.H. (2017). Amyotrophic lateral sclerosis. The Lancet, 377(9769), 942-955.