Kannada Speech Recognition System for Aphasic people

Jaya Aishwarya
Manipal Institute of Technology
Manipal Academy of Higher Eductaion
Manipal, India
jaya.ash16@gmail.com

Poornima Panduranga Kundapur
Manipal Institute of Technology
Manipal Academy of Higher Eductaion
Manipal, India
poornima.girish@manipal.edu

Sampath Kumar
Manipal Institute of Technology
Manipal Academy of Higher Eductaion
Manipal, India
kumar.sampath@manipal.edu

Hareesha K S
Manipal Institute of Technology
Manipal Academy of Higher Eductaion
Manipal, India
hareesh.ks@manipal.edu

***Abstract*— Aphasia is a loss or an impairment of language that is due to brain damage, which affects the production of speech and the ability to read or write. his lead to the necessity of large human resources in terms of a speech therapist and physicians, caretakers, etc. Having an automatic computer/mobile-based technology for the detection, evaluation and providing timely feedback would give a great benefit to the medical community immensely. It also speeds up the process with minimum efforts. This has motivated us to create a knowledge base to build machine learning model to understand and learn from the available data for an accurate prediction. The Automatic Speech Recognition (ASR) technology has the potential to enable individuals with aphasia using computer based conversion of spoken word to a text form. This would make it possible for language therapy software to provide feedback about the correctness of the user's spoken utterances or to engage the user in spoken dialog practice or other language therapy activities. ASR hence must be developed further in order to make it feasible for people with speech impairments.**

Keywords—Aphasia, ASR, Automatic Speech Recognition, speech therapy

I. INTRODUCTION

Aphasia refers to acquired language impairments resulting from focal brain damage. It can be caused due to a stroke, head injury or damage to a specific part of the brain that results in speech impairments [1]. The difficulties faced by patients dealing with this problem may range from slight troubles in finding the right words to express themselves to losing entirely the ability to speak, read, write and/or understand.

Automatic Speech Recognition (ASR) is the technology that allows us to use our voice to command the computer interface/ talk to it such that it resembles a human conversation. ASR techniques make it possible to process very large datasets, extract features and assess the same using various techniques. Speech Recognition systems' applications include voice dialing, call routing, data entry, direct voice input, etc. This has been found to be highly useful among people with hearing and speech impairments. Speech Recognition is also very useful for people who cannot use their hands effectively.

A. Aphasia and therapy

In order to treat Aphasic people, speech therapy is a common method employed [1]. This speech therapy is generally done face to face with the help of caretakers that train the patients also called as Speech Pathologists. Traditionally, speech therapy involves the patient going up to the speech rehabilitation center to sit with the speech pathologists and learn/ train certain words for a set period of time. It is very much human intensive. These speech pathologists have a tight schedule, and the patient must fit himself into that for a limited time period. Also, using a speech therapist for a long time will lead to high costs. It may also prone to human errors. Sometimes, the speech pathologists' system might not be very effective. All these drawbacks hinder the patients' improvement.

B. ASR

Therefore, ASR was developed such that speech therapy can be implemented at the comfort of their own home, with/without a speech pathologist or a caretaker. The entire process has been automated.

Most ASR's use different methods for phoneme recognition and word decoding. After extracting the features using methods like MFCC (Mel-Frequency Cepstral Coefficients) or PLP (Perceptual Linear Prediction) or LPC (Linear Predictive Coding) to attain phonemes for a desired frame length. To further decode these frames, HMM (Hidden Markov Models) with a previously trained language model is used. This is used to find the sequence of phonemes that the output is most likely to represent.

C. Convolutional Neural Networks (CNN)

CNN is a class of deep, feed-forward artificial neural networks. CNN's use multilayer perceptions for minimal pre-processing. CNN was inspired by biological processes i.e. patterns between neurons that resemble the organization of the cortex. A typical CNN consists of an input, output and multiple hidden layers. These hidden layers in the CNN consist of convolutional layers, pooling layers, fully connected layers and normalization layers [2]

The motivation for this study was the need to create a knowledge base of commonly used words (for people with aphasia), building a machine learning model that would understand and learn from the available data in order to accurately predict a match with the spoken word.

II. BACKGROUND RESEARCH

A. Previously implemented methods

ASR has been an active topic of research in the field of Machine Learning since the '70s. Most ASR systems developed use two different methods for word decoding and phoneme recognition. Earlier, researchers used other classification algorithms on highly specialized features like

MFCC (Mel Frequency Cepstral Coefficients) to distribute the phonemes for each frame. Hidden Markov Models (HMMs) [3] are further used for word decoding where a previously trained language model is sued to map the phonemes to a string, thus forming the output word. In the earlier stages of Deep Neural Networks (DNN) [4] also, the two tasks are separated out where DNN's discriminative power is sued for the phoneme recognition whereas, HMM was again adopted for word mapping.

In a typical hybrid system, the neural network is trained to predict frame-level targets obtained from a forced alignment generated by an HMM/GMM system [5]. The temporal modelling and decoding operations are still handled by an HMM but the posterior state predictions are generated using the neural network. This hybrid approach is problematic in that training the different modules separately with different criteria may not be optimal for solving the final task. As a consequence, it often requires additional hyper-parameter tuning for each training stage which can be laborious and time consuming.

Recently, the phoneme recognition and word decoding have been consolidated, where the model is trained jointly using neural networks. The first idea to approach this problem statement with Convolutional Neural Networks Was done in [6] where the power spectrum was taken as an input.

Dynamic Time Warping (DTW) [7] is another algorithm that has been given a lot of importance before the classifiers started doing their magic with the datasets and training samples. This algorithm calculates the closeness between two-time series that might shift with respect to time and speed. The wrapping in two time arrangements can be utilized to focus on the closeness. DTW compares two time arranged patterns and measure the similarity between them by computing the normalized distance between them.

Further DTW can be applied to classifiers such as the K-nearest neighbor (KNN) algorithm [8] or the Linde–Buzo–Gray (LBG algorithm) and such [9]. Here, the datasets can get trained further and cluster the points and then find the normalized distance between the test sample and the dataset points.

B. *Summary of Background Work*

After going through all the algorithms that have been implemented previously including methods that are as simple as DTW to extremely complicated Machine Learning algorithms, it has become quite evident that, intelligent machines can certainly help the mankind to do such tasks repetitively much more efficiently with little human intervention. Further, using two different methods for phoneme recognition and word detection as in the case of HMM's and DNN has been ruled out.

Similarly, using methods like DTW results in poor accuracy levels. On further implementation of this method, it has been proven that this method fails to provide an accuracy level around 75%. Hence, using this method is also ruled out.

Therefore, a system like CNN can be used for this purpose where after building training model, testing and validation are further done to get very good accuracy results. Thus, end-to-end Neural Networks is implemented in such cases.

Neural Networks are made up of neurons that have learnable weights and biases. Each neuron receives some inputs, performs a dot product and optionally follows it with a non-linearity. Neural Networks receive an input (a single vector), and transform it through a series of *hidden layers*. Each hidden layer is made up of a set of neurons, where each neuron is fully connected to all neurons in the previous layer, and where neurons in a single layer function completely independently and do not share any connections. The last fully-connected layer is called the "output layer" and in classification settings it represents the class scores. The convolutional neural network is used in the proposed method.

III. METHODOLOGY

A. *Simplified ASR approach*

The methodology followed was straightforward. The input is the speech sample which is compared with a available lexicon and using speech to text conversion, the output is to predict a match.

In convolutional neural network, the input is represented as a matrix. Each row of the matrix corresponds to one word. These matrices are word embeddings such that for a 10 word sentence with 100 embeddings, the input would be characterised as an image of size 10x100. The width of our filters is usually the same as the width of the input matrix. The height may vary, but sliding windows over 2-5 words at a time is typical. Putting all the above together, a Convolutional Neural Network is developed that functions.

Here, Speech recognition is being implemented with the help of Tensorflow, an open source Machine Learning library in Python.

First, a database was created consisting of 6 unique Kannada words: "Bana", "Buss", "Mane", "Karadi", "Sebu" and "Vonte". Now, each of these words have been spoken 30 times each, hence making 180 training samples in all. Similarly, a testing dataset must also be created. There is no limit or a count on the number of samples that must be present in this dataset since it must be compared to the model created after training the CNN on the training dataset.

The model is trained with parameters like number of iterations, learning rate, accuracy and cross entropy values are shown in every iteration.

Learning Rate: Higher the learning rate lower the accuracy, whereas lower the learning rate, better the accuracy. Therefore, the learning rate can be fixed on the trial and error basis. Hence, an optimum learning rate like 0.001 is used. Fig. 1 depicts loss vs epoch where epoch refers to the number of times a particular example was used in training. To further increase the accuracy, after 15000 iterations, 3000 iterations are trained using a learning rate of 0.0001. This is the point where all the weights distributed are saved into one file. Further on, only these weights are used.

After every 400 iterations, the training set creates a confusion matrix to help the user understand the exact status of the training data. This matrix is a square matrix of the size of the number of words that are used to train the data. Each row and each column depicts a particular word. This matrix helps the user to understand how many times the classification is done right.

After several thousands of iterations, when the accuracy level reaches 80-90%, a confusion matrix as shown in Fig. 2 that will be formed.

This demarcation is done automatically. The Validation Accuracy must be close to the Training Accuracy. If the training accuracy increases, but the validation accuracy remains constant, this refers to a situation called as Overfitting. The Overfitting condition occurs where the model parameters aren't high enough so, the model keep suing only the training dataset without using the validation set. This is shown in Fig. 1.

When overfitting occurs, the dataset must be increased, number of parameters must be increased.

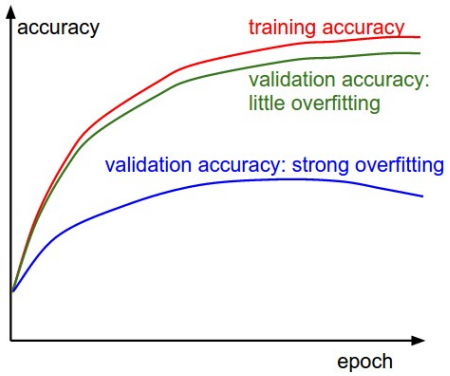

Fig. 1. Accuracy vs Epoch showing overfitting

```
[[258   0   0   0   0   0   0   0   0   0   0   0]
 [  0 170   0   0   0   0   0   0   0   0   0   0]
 [  0   0 164   0   0   0   0   0   0   0   0   0]
 [  0   0   0 185   0   0   0   0   0   0   0   0]
 [  0   0   0   0 155   0   0   0   0   0   0   0]
 [  0   0   0   0   0 200   2   0   0   0   0   0]
 [  0   0   0   0   0   0 189   0   0   0   0   0]
 [  0   0   0   3   0   0   0 136   0   0   0   0]
 [  0   0   0   0   0   0   0   0 172   0   0   0]
 [  0   0   0   0   0   0   0   0   0  94   0   0]
 [  0   0   0   0   0   0   0   0   0   0  80   0]
 [  0   0   0   0   0   0   0   0   0   0   0 210]]
```

Fig. 2. Confusion Matrix after ~ 16000 iterations

Further on, to run this model, a command line is used, where any wav file can be given as an input to give the following result with the top three wav files with the highest probabilities to be recognized as the input file given. The same can be used in real-time as well just by tweaking the original code.

sebu (score = 0.91477)
mane (score = 0.08139)
vonte (score = 0.003808)

TABLE I. TRANSLATION OF WORDS USED IN THIS PAPER (KANNADA TO ENGLISH)

Kannada Word	Pronunciation	English meaning
ಸೇಬು	Sebu	Apple
ಮನೆ	Mane	House
ಒಂಟೆ	Vonte	Camel
ಬಾಣ	Bana	Arrow
ಕರಡಿ	Karadi	Bear
ಬಸ್ಸು	Bass	Bus

IV. RESULT ANALYSIS

As explained previously, a training dataset of 6 words with 30 samples each i.e. 180 samples in all is being used and their MFCC coefficients are being extracted.

These MFCC coefficients are unique. For example, refer to Fig. 3 and Fig. 4. Thus, it is safe to conclude that the features that have been extracted are unique and distinguishable. These features that are so extracted can be further used to build a model that is being frozen onto a graph with an extension of x.pb. Another list of labels is created to map it to the training dataset. This list of labels contains all the words that have been used in the dataset. This list is shown in Fig. 5

The graph so made is used to do speech recognition in real time. The input given is compared to the model generated output and a result is generated that shows the probabilities of the three words that show the maximum probability of being the input word.

Each of the figures shown below depict the output that the system has generated after comparing it to the model that was computed by the system earlier. Fig. 6 shows the comparison of the input word *Bana*. Fig. 7 shows the comparison of the input word *Mane*

This method gives an accuracy level of ~50%. On application of a very large dataset, ~50,000 training files, a much higher accuracy is expected.

No significant deviations from the expected results were noted. Although, the deviations that can be seen are caused from known and expected factors such as:

- Random Noise
- Constrained dataset
- Disturbances during sample recordings
- Audio effects such as delay, echo, etc
- Too little time used in recording.

V. DISCUSSION AND CALCULATIONS

This system can also be used as a personal speech recognition system in our phones that detects our voice and can act as per requirement. This works very well in systems with high CPU power and GPU's. This tool is very fast paced. CNN as an algorithm itself is very fast in computation.

Building a large dataset is the most important part of building a speech recognition system. The training set must consist of different male and female voices saying the same words at least 20-30 times each in different ways. This ensures a very general and usable ASR that is very vivacious.

Using only CNN as the classifier may not always work. For a much more efficient system, it is better to use Recurrent Neural Networks along with Convolutional Neural Networks for classifying the input sample. The model can be generated using CNN's but while testing, RNN's are used so that system memory can learn in case of wrong output so that the weights can be re arranged as per the requirement

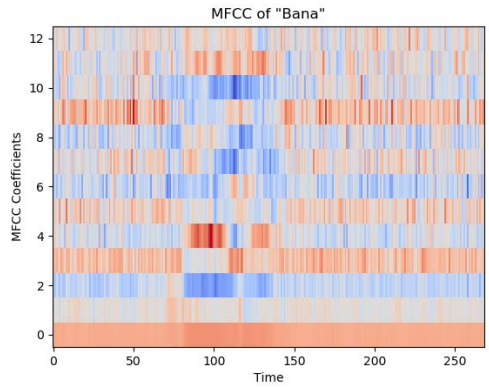

Fig. 3. Distingushed MFCC coefficients of “Bana”

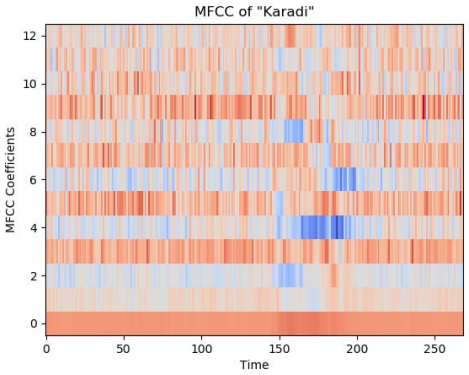

Fig. 4. Distingushed MFCC coefficients of “Karadi”

Fig. 5. Labels that are mapped to the training dataset

Fig. 6. Output results when the input speech is given as “Bana”

Fig. 7. Output results when the input speech is given as “Mane”

ACKNOWLEDGMENT

The authors would like to acknowledge and express gratitude to the Department of Speech and Hearing, School of Allied Health Sciences, Manipal Academy of Higher Education, Manipal, in particular, Rajath Shenoy for the use of the speech samples as a part of this study.

REFERENCES

[1] Bragoni, M., Altieri, M., Di Piero, V. et al. Bromocriptine and speech therapy in non-fluent chronic aphasia after stroke, Neurol Sci (2000) 21: 19.

[2] Abdel-Hamid, Ossama, Li Deng, and Dong Yu. "Exploring convolutional neural network structures and optimization techniques for speech recognition." In Interspeech, vol. 2013, pp. 1173-5. 2013.

[3] Longfei Li, Yong Zhao, Dongmei Jiang and Yanning Zhang. Hybrid Deep Neural Network - Hidden Markov Model Based Speech Emotion Recognition. Humaine Association Conference on Affective Computing and Intelligent Interaction IEEE 2013.

[4] I. Patel and Y. S. Rao. Speech Recognition Using Hidden Markov Model with MFCC-Subband Technique. 2010 International Conference on Recent Trends in Information, Telecommunication and Computing, Kochi, Kerala, 2010, pp. 168-172.

[5] P. Swietojanski, A. Ghoshal and S. Renals, Revisiting hybrid and GMM-HMM system combination techniques. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, 2013, pp. 6744-6748.

[6] O. Abdel-Hamid, A. r. Mohamed, H. Jiang, L. Deng, G. Penn and D. Yu. Convolutional Neural Networks for Speech Recognition. IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 22, no. 10, pp. 1533-1545, Oct. 2014.

[7] Keogh, Eamonn and Ratanamahatana, Chotirat Ann. Exact indexing of dynamic time warping. Knowledge and Information Systems, 2005,7:3, pages=358-386.

[8] J. M. Keller, M. R. Gray and J. A. Givens. A fuzzy K-nearest neighbor algorithm. IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-15, no. 4, pp. 580-585, July-Aug. 1985.

[9] Suman, M & Khan, Habibulla & Latha, Madhavi & Devarakonda, Dr Aruna. (2012). Speech Enhancement and Recognition of Compressed Speech Signal in Noisy Reverberant Conditions. Advances in Intelligent and Soft Computing. 132. 379-386. 10.1007/978-3-642-27443-5-43.