SPEECH ASSISTANCE FOR THE DEAF

[1]Kalpana Deorukhkar, [2]Ms. Gauri Jare, [3]Ms. Aishwarya Sebin, [4]Ms, Wencita Rodrigues
[1]Assitant Professor, [2,3,4]Students,
[1]Computer Engineering Department,
[1]Fr. Conceicao Rodrigues College of Engineering, Mumbai, India.

Abstract: Communication is a fundamental for existence as well as it is a key element to progress. Unfortunately, there are many people born without this ability, therefore for effective communication between people many applications have been developed as a mediator for effective communication. Our project's main goal is to bridge the gap between the deaf and other people with no hearing disability by creating an application for converting Speech to text and Indian Sign Language to text. In speech recognition, an acoustic (sound) signal is divided into words or sub-words units which will make the recognition easy. Recurrent neural networks (RNNs) is a class of artificial neural networks (ANN) And are powerful sequence learners which is perfect for this job. RNN has the possibility of processing input of any length but their applications are limited because they require pre-segmented training data, post–processing to transform their output into label sequence and also cannot consider any future input for the current state. This paper presents a novel method for training RNNs to label un-segmented sequences directly and sign language recognizing hand gestures in Indian sign language.

Index Terms **- Attention RNN, Faster RCNN, Speech Recognition, Indian Sign Language.**

I. INTRODUCTION

Communication plays an essential role in our lives. Humans started with signs, symbols, and then made progress to a stage, where they began communicating with languages. Different ideas formed in the mind of the speaker are communicated by speech in the form of words, phrases, and sentences by applying some proper grammatical rules. Speech Recognition is one of the technologies that has now been gaining immense popularity. It was once the dream of Science fiction and has till date not been perfected. This technology first started taking shape in the 1950's with the invention of the "Audrey" system [1]. Of all the modern developments in technology, speech-to-text translation is the most exciting for people who are deaf or have hearing loss.

By classifying the speech with voiced, unvoiced and silence (VAS/S) an elementary acoustic segmentation of speech which is essential for speech can be considered [2]. Speech recognition technology is found in the form of ordinary interfaces so that the language barrier faced by diverse people would no longer be an issue, even assisting people with hearing disability [11]. For conversion of speech to text we make use of the End-to-End Automatic Speech Recognition model. Recurrent neural networks (RNNs) require no prior knowledge of the data, beyond the choice of input and output representation. They can be trained discriminatively, and their internal state provides a powerful, general mechanism for modelling time series. In addition, they tend to be robust to temporal and spatial noise.

There have been various mediums available to translate different languages, but sign language translation systems have been rarely developed, this is due to the scarcity of any sign language corpus. Sign Language is the natural and expressive way for the hearing-impaired people. People who are not physically challenged, never try to learn sign language to interact with the deaf people. The computer can be programmed in such a way that it can translate sign language to text format and the difference between the normal people and the deaf community can be minimized [7]. Indian sign language uses both hands to represent each alphabet and gesture.

Here this proposed system is able to recognize the various alphabets of Indian Sign Language; this will reduce the noise and give accurate results. This system introduces efficient and fast techniques for identification of the hand gesture representing an alphabet of the Sign Language. We designed and implemented a real time Sign Language Recognition system to recognize 36 gestures from the Indian Sign Language by hand gesture recognition system for text generation. These signs are captured by using a webcam. The extracted features are compared by using a pattern matching algorithm. In order to calculate the sign recognition, the features are compared with a test database. Finally, the recognized gesture is converted into text. This system provides an opportunity for a deaf-dumb person to communicate with non-signing people without the need of an interpreter.

II. Literature review

A. Speech Recognition for Out-of-vocabulary Words

Decadt et. al.[10] describes a method to develop the readability of the textual output in a large vocabulary continuous speech recognition system when out-of-vocabulary words occur. The basic idea is to replace uncertain words in the transcriptions with a phoneme recognition result that is post-processed using a phoneme-to-grapheme converter. This technique uses machine learning concepts.

B. Sign Language Recognition Using WiFi

SignFi to recognize sign language gestures using WiFi. SignFi uses Channel State Information (CSI) measured by WiFi packets as the input and a Convolutional Neural Network (CNN) as the classification algorithm [13]. Existing WiFi-based sign gesture recognition technologies are tested on no more than 25 gestures that only involve hand and/or finger gestures. SignFi is able to recognize 276 sign gestures, which involve the head, arm, hand, and finger gestures, with high accuracy. SignFi collects CSI measurements to capture wireless signal characteristics of sign gestures. Raw CSI measurements are pre-processed to remove noises and recover CSI changes over sub-carriers and sampling time. The average recognition accuracy of SignFi is 86.66% for 7,500 instances of 150 sign gestures performed by 5 different users.

C. Sign Language Recognition using 3D Convolutional Neural Networks

Sign Language Recognition (SLR) targets on interpreting the sign language into text or speech, so as to facilitate the communication between deaf-mute people and ordinary people. However, it is difficult to design reliable features to adapt to the large variations of hand gestures. To approach this problem, they propose a novel 3D convolutional neural network (CNN) which

extracts discriminative spatial-temporal features from raw video stream automatically without any prior knowledge, avoiding designing features. To boost the performance, multi-channels of video streams, including color information, depth clue, and body joint positions, are used as input to the 3D CNN in order to integrate color, depth and trajectory information [4].

D. Automatic Speech Recognition by Cuckoo Search Optimization based Artificial Neural Network Classifier

Bansal et. al. [12] presents an efficient automatic speech recognition (ASR) system using Cuckoo Search Optimization (CSO) based optimization technique for Artificial Neural Network (ANN). Here CSO is used in order to improve the classification performance of neural networks.

Both acoustic modeling and language modeling are important parts of modern statistically-based speech recognition algorithms. There are different models that have been developed over the years.

- Hidden Markov Model Approach

HMMs are used in speech recognition because a speech signal can be viewed as a piecewise stationary signal or a short-time stationary signal. In a short time-scale (e.g., 10 milliseconds), speech can be approximated as a stationary process and HMM would output a sequence of n-dimensional real-valued vectors (with n being a small integer, such as 10). Each word, or (for more general speech recognition systems), each phoneme, will have a different output distribution; a hidden Markov model for a sequence of words or phonemes is made by concatenating the individual trained hidden Markov models for the separate words and phonemes [8].

- Dynamic Time warping (DTW)-based speech recognition approach

Dynamic time warping is an algorithm for measuring similarity between two sequences that may vary in time or speed. Dynamic time warping is an approach that was historically used for speech recognition but has now largely been displaced by the more successful HMM-based approach [14].

- Neural Networks

Neural networks have been used in many aspects of speech recognition such as phoneme classification, isolated word recognition, etc. Neural networks make fewer explicit assumptions about feature statistical properties than HMMs and have several qualities making them attractive recognition models for speech recognition. When used to estimate the probabilities of a speech feature segment, neural networks allow discriminative training in a natural and efficient manner. However, in spite of their effectiveness in classifying short-time units such as individual phonemes and isolated words, early neural networks were rarely successful for continuous recognition tasks because of their limited ability to model temporal dependencies. However, more recently, LSTM and related recurrent neural networks (RNNs) and Time Delay Neural Networks (TDNN's) [15,16] have demonstrated improved performance in this area.

- End-to-end Automatic System Recognition model

End-to-end models jointly learn all the components of the speech recognizer. This is valuable since it simplifies the training process and deployment process [2]. The first attempt at end-to-end ASR was with Connectionist Temporal Classification (CTC)-based systems. Jointly, the RNN-CTC model learns the pronunciation and acoustic model together. CTC models can directly learn to map speech acoustics to English characters, but the models make many common spelling mistakes and must rely on a separate language model to clean up the transcripts. An alternative approach to CTC-based models are attention-based models. Unlike CTC-based models, attention-based models do not have conditional-independence assumptions and can learn all the components of a speech recognizer including the pronunciation, acoustic and language model directly [17].

III. PROPOSED METHODOLOGY

Unlike CTC-based models, attention-based models do not have conditional-independence assumptions and can learn all the components of a speech recognizer including the pronunciation, acoustic and language model directly. The RNN encoder has an input sequence x1, x2, x3, x4. We denote the encoder states by c1, c2, c3. The encoder outputs a single output vector c which is passed as input to the decoder. Like the encoder, the decoder is also a single-layered RNN, we denote the decoder states by s1, s2, s3 and the network's output by y1, y2, y3, y4.

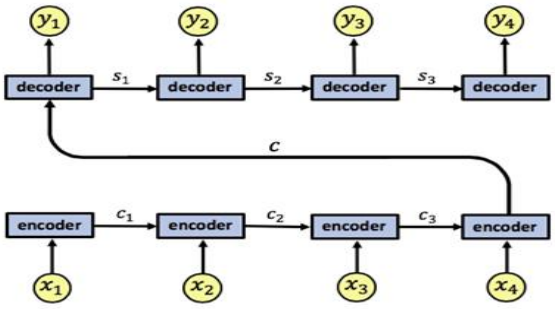

Figure 1: Encode-Decoder Architecture

A problem with this architecture lies in the fact that the decoder needs to represent the entire input sequence x1, x2, x3, x4 as a single vector c, which can cause information loss. Moreover, the decoder needs to decipher the past information from this single vector, a complex task in itself.

This attention model has a single layer RNN encoder, again with 4-time steps. We denote the encoder's input vectors by x1, x2, x3, x4 and the output vectors by h1, h2, h3, h4.

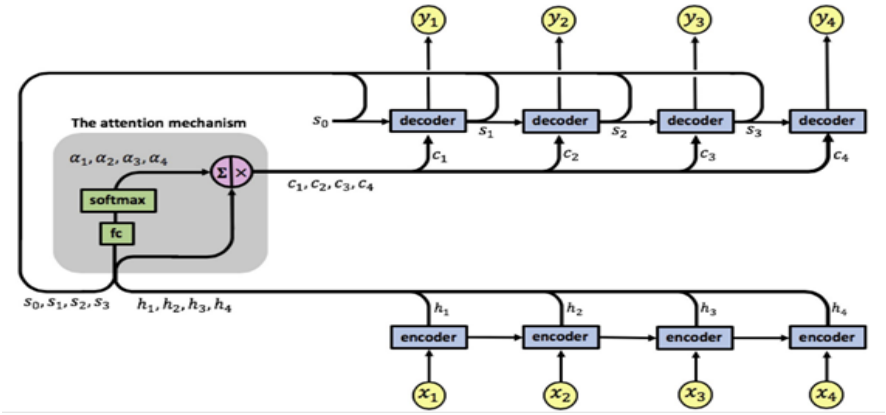

Figure 2: Encoder-Decoder with Attention

The attention mechanism is located between the encoder and the decoder, its input is composed of the encoder's output vectors h1, h2, h3, h4 and the states of the decoder s0, s1, s2, s3, the attention's output is a sequence of vectors called context vectors denoted by c1, c2, c3, c4. The attention model is applied over an LSTM model. All recurrent neural networks have the form of a chain of repeating modules of neural networks. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer. LSTMs also have this chain-like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four, interacting in a very special way.

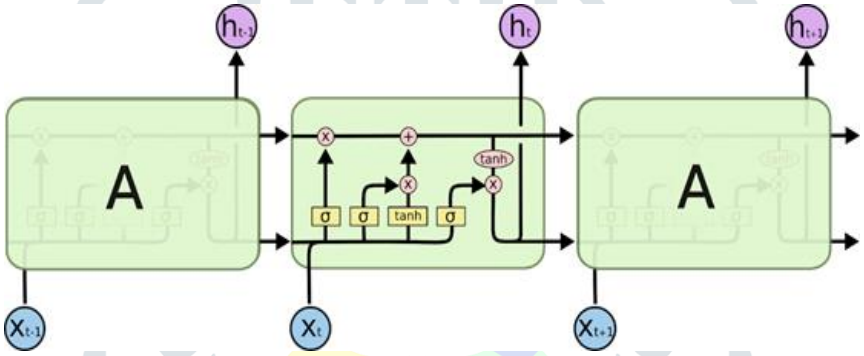

Figure 3: LSTM Architecture

In the above diagram, each line carries an entire vector, from the output of one node to the inputs of others. The pink circles represent pointwise operations, like vector addition, while the yellow boxes are learned neural network layers. Lines merging denote concatenation, while a line forking denotes its content being copied and the copies going to different locations [15].

The Google Speech Commands Dataset was created by the TensorFlow and AIY teams to showcase the speech recognition example using the TensorFlow API. The dataset has 65,000 clips of one-second-long duration. Each clip contains one of the 30 different words spoken by thousands of different subjects.

The clips were recorded in realistic environments with phones and laptops. {'unknown' : 0, 'silence' : 0, '_unknown_' : 0, '_silence_' : 0, '_background noise ' : 0, 'yes' : 2, 'no' : 3, 'up' : 4, 'down' : 5, 'left' : 6, 'right' : 7, 'on' : 8, 'off' : 9, 'stop' : 10, 'go' : 11, 'zero' : 12, one' : 13, 'two' : 14, 'three' : 15, 'four' : 16, 'five' : 17, 'six' : 18, 'seven' : 19, 'eight' : 20, 'nine' : 1, 'backward':21, 'bed':22, 'bird':23, 'cat':24, 'dog':25, 'follow':26, 'forward':27, 'happy':28, house':29, 'learn':30, 'marvin':31, 'sheila':32, 'tree':33, 'visual':34, 'wow':35}

In order to build our model, different layers from Keras were used. Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow.

The final model looks like

normalization2d_2 (Normalizatio	(None, 60, None, 1)	0	mel_stft[0][0]
permute_1 (Permute)	(None, None, 60, 1)	0	normalization2d_2[0][0]
conv2d_1 (Conv2D)	(None, None, 80, 10)	60	permute_1[0][0]
batch_normalization_1 (BatchNor	(None, None, 80, 10)	40	conv2d_1[0][0]
conv2d_2 (Conv2D)	(None, None, 80, 1)	51	batch_normalization_1[0][0]
batch_normalization_2 (BatchNor	(None, None, 80, 1)	4	conv2d_2[0][0]
squeeze_last_dim (Lambda)	(None, None, 80)	0	batch_normalization_2[0][0]
bidirectional_1 (Bidirectional)	(None, None, 128)	74752	squeeze_last_dim[0][0]
bidirectional_2 (Bidirectional)	(None, None, 128)	99328	bidirectional_1[0][0]
lambda_1 (Lambda)	(None, 128)	0	bidirectional_2[0][0]
dense_1 (Dense)	(None, 128)	16512	lambda_1[0][0]
dot_1 (Dot)	(None, None)	0	dense_1[0][0] bidirectional_2[0][0]
attSoftmax (Softmax)	(None, None)	0	dot_1[0][0]
dot_2 (Dot)	(None, 128)	0	attSoftmax[0][0] bidirectional_2[0][0]
dense_2 (Dense)	(None, 64)	8256	dot_2[0][0]
dense_3 (Dense)	(None, 32)	2080	dense_2[0][0]
output (Dense)	(None, 36)	1188	dense_3[0][0]

Total params: 1,293,935
Trainable params: 282,249

To implement this model, we divided the dataset of audio recordings into 8:2 ratio for train: testing datasets. The model was trained for 10 epochs and achieved an accuracy of 94%.

a) To get the best results, we trained our dataset on RCNN, SSD-mobilenet and faster RCNN. It was observed that faster RCNN (93.6%) had a much better accuracy than SSD (86%) and RCNN (91.2%). While RCNN and SSD are much faster than RCNN.

b) Using selective search, R-CNN extracts a set of regions from the given image [6], and then checks if any of these boxes contains an object. Unfortunately, the R-CNN model is slow due to these n-number of steps involved.

c) Fast R-CNN passes the whole image to ConvNet which generates regions of interest. Rather than using three different models, it uses only one model which extracts features from the regions and allocates them into different classes, and then returns the bounding boxes.

d) However, Fast R-CNN is not fast enough when applied on a huge dataset as it applies selective search for detecting the regions.

e) Faster R-CNN solves the issue of selective search by replacing it with RPN (Region Proposal Network). The feature maps are obtained from the input image using Convolutional Network and then passed through RPN which in turn returns object proposals. Then these maps are classified and the bounding boxes are predicted.[5]

The system consists of three stages: training, testing and a recognition phase. In the training phase, 2016 images were passed through the faster RCNN model wherein the images were divided into 36 categories that is 26 English alphabets and 10 digits. For the testing phase, 504 images were passed through the same model. All these images were annotated manually for better precision. The model was trained for 20336 steps with a loss of 0.7 - 0.9 and on an Intel core i5 processor with 12 GB RAM and 64-bit operating system. Faster-RCNN is one of the most well-known object detection neural networks. At the conceptual level, Faster-RCNN is composed of 3 neural networks — Feature Network, Region Proposal Network (RPN), Detection Network.

The Feature Network is a well-known pre-trained image classification network such as VGG excluding a few bottom or top layers. The functionality of this network is to generate appropriate features from the images. The output of this network model maintains the structure of the original image. Furthermore, we compute convolution by sliding filters all along our input image and the output is a 2-D matrix called feature map. It consists of reducing the number of features in the feature map by omitting pixels with low values.

The RPN is usually a sophisticated network consisting of three convolutional layers. It consists of a common layer that feeds into two layers where one is for classification and the other for bounding box regression. The function of RPN is to generate bounding boxes called ROI (Region of Interests) which have a high probability of containing any object. The result of this network is a number of bounding boxes identified by the pixel coordinates of two diagonal corners, and a value which can be 1, 0, or -1 which in turn indicates whether an object is present in the bounding box or not and hence the box can be omitted.

The Detection Network takes input from both the Feature Network and RPN, and outputs a final class and bounding box. It is composed of 4 Fully Connected or Dense layers. It also consists of 2 stacked common layers shared by a classification layer and a bounding box regression layer. To help it detect only the insides of the bounding boxes, the features are shaped or cropped according to the bounding boxes. The RPN and Detection Network thoroughly need to be trained. As these are the layers where most of the heavy computation of Faster-RCNN lie.

Training the RPN consists of a number of bounding boxes to be generated with the help anchor boxes. An anchor is basically every pixel in an image. A group of squares of pixels is allocated to every anchor. Prior to the feature extraction some reshaping is applied on the original image. Consistently, anchors are positioned across both dimensions of the reshaped image. Then a number of bounding boxes of different shapes and sizes surround each anchor. The average number of bounding boxes surrounding per anchor is 9, which leads to a huge number of anchor boxes per image.

In the reduction phase, Non-Maximum Suppression (NMS) is used. It reduces the number of boxes by removing the ones which overlap with boxes which have a higher probability. Roughly around 2000 and 300 boxes are removed during the training and testing phase respectively. Further, in the training process these 2000 boxes are reduced to 256 boxes and then sent to the Detection Network while in the testing phase these boxes go straight to the Detection Network. To generate labels for RPN classification which determines whether the bounding box contains a foreground, background or unnecessary objects. Intersection over Union of all boxes against all the ground truth boxes are taken. With the help of IoUs the 256 ROIs are labelled as foreground or background or ignored. After deleting the boxes labelled ignored, the cross-entropy loss is calculated.

Then the bounding box regression is applied where the RPN tries to compress the boundary around the center of the anchor boxes. For this, targets need to be generated along with their respective losses to be computed for back propagation. From the center of the ground truth box to the anchor box, the target delta or the distance vector is measured and normalized to the size of the anchor. The size target is that of the log of the ratio of size of every dimension of the ground truth over the anchor box. The loss is calculated by using an expression called 'Smooth L1 Loss'. The losses are back propagated the usual way to train RPN. The RPN can be trained by itself or combined with the Detection Network.

The training of the Detection Network is somewhat similar to that of RPN. First, the ROI generated from the IoUs of all the 2000 boxes or so are calculated against each ground truth bounding. Then the ROIs are labeled as foreground or background depending on the corresponding threshold values. Then a constant number of ROIs are selected. If there are not enough ROIs to fill the fixed number, then some ROIs are randomly duplicated [6].

The ROI width and heights are scaled to the feature size. As a batch, the set of cropped features for each image are passed through the Detection Network. The score and the bounding box for each class is the final output given by the last dense layers. With the help of IoU threshold values (foreground above 0.5, and background between 0.5 and 0.1) of all the ROIs, the labels are generated for Detection Network classification. The loss calculation is similar to that of the RPN network. Sparse cross-entropy is used for classification and Smooth L1 Loss is used for bounding boxes. The difference with RPN loss is that there are more classes (say 20 including background) to consider instead of just 2 (foreground and background). We divided the dataset of hand signs into 8:2 for train: testing datasets. There are in all 36 classes for classification which consists of a-z and 0-9. Each class has 70 images with different rotations. All these images were annotated manually for better precision.

V. CONCLUSION

People are achieving remarkable results with RNNs. Essentially all of these are achieved using LSTMs. They really work a lot better for most tasks! LSTMs were a big step in what we can accomplish with RNNs. In Sign language research works have focused mainly on the recognition of static signs of ISL from images or video sequences. Using various concepts of image processing and fundamental properties of image we tried to develop this system. Every God creature has an importance in the society, remembering this fact, let us try to include hearing impaired people in our day to day life and live together. As an extension to existing work, we are going try different variants of RNN and CNN object detection algorithms.

REFERENCES

[1] Suma Swamy and K.V Ramakrishnan, "Evolution of Speech Recognition – A Brief History of Technology Development", Elixir International Journal, 2013.

[2] Prachi Khilari and Prof. Bhope V. P, "A review on Speech to Text Conversion Methods" International Journal of Advanced Research in Computer Engineering & Technology, Volume 4 Issue 7, July 2015

[3] Alex Graves,Santiago Fernandez, Faustino Gomez and Jurgen Schmidhuber, "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks", International Conference on Machine Learning, 2006.

[4] Jie Huang, Wengang Zhou, Houqiang Li, Weiping Li, "Sign Language Recognition using 3D convolutional neural networks", IEEE International Conference on Multimedia and Expo, 2015

[5] Ross Girshick Microsoft Research, "Fast R-CNN", International Conference on Computer Vision, 2015.

[6] J.R.R. Uijlings, K.E.A. van de Sandel, T. Gevers , and A.W.M. Smeulders, "Selective Search for Object Recognition", International Journal of Computer Vision, 2012.

[7] Mahesh Kumar N B Assistant Professor (Senior Grade), "Conversion of Sign Language into Text", International Journal of Applied Engineering Research, Vol 13, 2018

[8] Ayushi Trivedi,Navya Pant, Pinal Shah,Simran Sonik and Supriya Agrawal, "Speech to text and text to speech recognition systems-A Review", IOSR Journal of Computer Engineering (IOSR-JCE),Volume 20, Issue 2, Ver. I.

[9] Prerana Das, Kakali Acharji, Pranab Das and Vijay Prasad, "Voice Recognition System: Speech-to-text", Journal of Applied and Fundamental Sciences.

[10] Bart Decadt, Jacques Duchateau, Walter Daelemans and Patrick Wambacq, “Phoneme-to-grapheme conversion for out-of-vocabulary words in large vocabulary speech recognition”, IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU '01.

[11] B. Raghavendhar Reddy and E. Mahender, “Speech to Text Conversion using Android Platform”, International Journal of Engineering Research and Applications (IJERA), Vol. 3, Issue 1, January -February 2013

[12] Dipali Bansal; Neelam Turk; Sunanda Mendiratta, “Automatic speech recognition by cuckoo search optimization based artificial neural network classifier”, International Conference on Soft Computing Techniques and Implementations, 2015

[13] Yongsen Ma, Gang Zhou, Shuangquan Wang,Hongyang Zhao,Woosub Jung,"SignFi: Sign Language Recognition Using WiFi",Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies,March 2018, Article No.: 23 https://doi.org/10.1145/3191755

[14] Yurika Permanasari, Erwin H. Harahap and Erwin Prayoga Ali, "Speech recognition using Dynamic Time Warping (DTW)", Journal of Physics: Conference Series, Volume 1366, 2nd International Conference on Applied & Industrial Mathematics and Statistics 23–25 July 2019, Kuantan, Pahang, Malaysia

[15] A lex Graves, Santiago Fernandez, Faustino Gomez and Jurgen Schmidhuber, “Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks”, 2006 International Conference on Machine Learning.

[16] Vijayaditya Peddinti, Daniel Povey, Sanjeev Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts", Center for Language and Speech Processing & Human Language Technology Center of Excellence, Johns Hopkins University, Baltimore, USA.

[17] Amit Das, Jinyu Li, Member, Guoli Ye, Rui Zhao, Member, Yifan Gong,"Advancing Acoustic-to-Word CTC Model with Attention and Mixed-Units" IEEE/ACM Transactions on Audio, Speech, and Language Processing, September 2019