Real-Time Speech-To-Text / Text-To-Speech Converter With Automatic Text Summarizer using Natural Language Generation And Abstract Meaning Representation

K. P. Vijayakumar, Hemant Singh, Animesh Mohanty

Abstract: Due to extensive needs for growth in various sectors, which include software, telecom, healthcare, defence, etc., there is a necessary increase in the number as well as the duration of meetings, conference calls, reconnaissance stakeouts, financial reviews. The obtained reports of these play a significant role in defining the plan of actions. The proposed model is to convert real-time speech to corresponding text and then to its respective summary using Natural Language Grammar (NLG) and Abstract Meaning Representation (AMR) graphs and then again turned back the obtained summary to speech. The proposed model intends to achieve the task using two major algorithms, 1) Deep Speech 2, 2) AMR graphs. The speech-recognition model recommended has a speedup of 4x if the algorithm runs on a Central Processing Unit (CPU), and the use of particular Graphics Processing Units (GPUs) for running deep learning algorithms can give a speedup of 21x. The performance of the summarizer used is close to the Lead-3-AMR-Baseline model, which is a solid baseline for the CNN/Dailymail dataset. The summarizer we use scores ROGUE score close to the Lead-3-AMR-Baseline model with an accuracy of 99.37%.

Keywords: Sequence to Sequence models, Neural Networks, Deep Speech 2, AMR parsing, Batch Normalization, SortaGrad, NLP, NLG, CTC.

I. INTRODUCTION

In this fast-paced world, propelled by the modern technological innovations, data to this century is what oil was to the previous one. Today, our world is parachuted by the gathering and dissemination of vast amounts of data. With such an enormous amount of data circulating in the vast digital space, there is a growing need to develop suitable machine learning algorithms for the intended task. In this era of deep learning, these algorithms automatically shorten longer texts and deliver accurate summaries that can fluently pass the intended messages.the paper is to introduce a real-time speech-to-text converter, the text will then be summarized, and the output will be given out as speech of the obtained summary. There have been many state-of-the-art machines for speech-recognition and text-summarization. But, the summarizer proposed will be able to process the speech as the input and should be able to output its respective summarization in an audio form. The focus of the proposed system is to replace the set of pipelined instructions by neural networks completely.The proposed model uses Deep Speech 2 for the Speech-to-text, which outperforms Deep Speech 1 in accuracy by reducing the error rate up to 43% for English [8]. Also, JAMR parser is used to parse the input to their respective summary. The parser is close to accuracy of Lead-3-AMR Baseline and comparison with other parsers is given in Section V. in Table 4.The remainder of the paper has been divided into following sections. Section II describes the related work and the literature survey. Further, Section III gives an idea about our proposed model and its architecture diagram. Section IV describes the implementation and all related formulas. In Section V results of our modules are presented in tabular form. Section VI talks about the conclusion and future work and improvements for the proposed model.

II. LITERATURE SURVEY

The author in [1] has done comparative analysis on the performance of three different algorithms. The author explains the different text summarization techniques. Extraction based summarization techniques are based on the mining of essential keywords from the given extract, which in turn are included in the summary. For comparison, three keyword extraction algorithms, namely TextRank, LexRank, Latent Semantic Analysis (LSA), were used. Three algorithms are explained and implemented in python language. ROUGE 1 is an evaluation metric used to evaluate the effectiveness of the extracted keywords. The results of the algorithms compared with the handwritten summaries and evaluation of the performance. In the end, the TextRank Algorithm gives a better result than the other two algorithms. And subsequently, in [2], the author has reviewed different techniques of Sentiment analysis and various methods of text summarization. These strategies are then utilized to decide the feelings and estimations in the content information, similar to surveys about motion pictures or items. In text-summarization, the author uses the NLP, and linguistic features of sentences are used for checking the importance of the words and sentences that can be included in the final summary.

Also, in this paper, an overview has been done of past research by the author, business related to content rundown and Sentiment examination, with the goal that new research regions that can be investigated by thinking about the benefits and faults of the current flow systems and techniques.In [3], the author proposed a particular version of KNN. The likeness between feature vectors is computed by thinking about the comparability among characteristics as well as values. Text summarization is viewed as the task of classification by the author. The text is then partitioned into paragraphs or sentences to classify it into a 'summary or 'non-summary' by the classifier. The modified version of KNN leads to a more compact representation of data items and better performance.As discussed in [4], the author presents an exhaustive survey on abstraction based text summarization techniques. The paper presents a study on two broad abstractive summary approaches: Structured based abstractive summarization and Semantic-based abstractive summarization. The author presents the review of various researches on both approaches of abstractive summarization. The author also covered the different methodologies and challenges in abstractive summary.As stated in [5], a text representation using Abstract Meaning Representation (AMR) is proposed. Originally, AMR is intended to represent the concepts and their relationships of one sentence only. In this manner, the set of sentences that compose an entire text results in a set of disconnected AMR graphs. The proposal is an architecture and method to add new data to offer more semantic information at the sentence level but to link and merge such graphs too. The final goal is to achieve a unique, invariant and independent standard representation of entire documents. Such canonical representation will allow us to generate new variants of the analyzed text, such us summaries, simplifications, etc. to satisfy different user's needs.The inference derived from related works is that, in the field of text-summarization, there have been many methods to get the task of summarization done. Also, from very early times, the task of summarization is processed using hand-engineered pipelined systems. Different authors propose different pipelined systems to get this task done until the very concept of neural network was explored in this domain. In [1:4], the authors use various hand-engineered principles for text-summarization, but in [5] the author highlights the better accuracy of using deep neural networks for the task. This has been the basic motivation of using neural networks for different tasks in the proposed system.

III. PROPOSED WORK

The purpose of the system is to convert real-time speech-to-text and summarize the respective text first, and then output the summary in the form of speech. First of all, the input shall be taken by voice through a microphone array. Then the information is given for CTC Loss and Batch Normalization, and this is the preprocessing step. After this, the pre-processed data will be given to the voice-to-text algorithm, Deep Speech 2 iteratively until there is still a need of dimensionality reduction. After the voice is converted to text, AMR Graph parsing will be done to summarize the recognized text and the summary will be generated. To convert the summarized text to voice, Google text-to-speech engine is being used.The architecture diagram of the proposed model is shown in Fig.1. All the algorithms and parsing techniques are discussed in this section itself. The list of algorithms and modules are as follows:

A. Calculation of CTC Loss Function
B. Batch Normalization
C. Deep Speech 2
D. AMR Parsing
E. Google Text-to-Speech Engine.

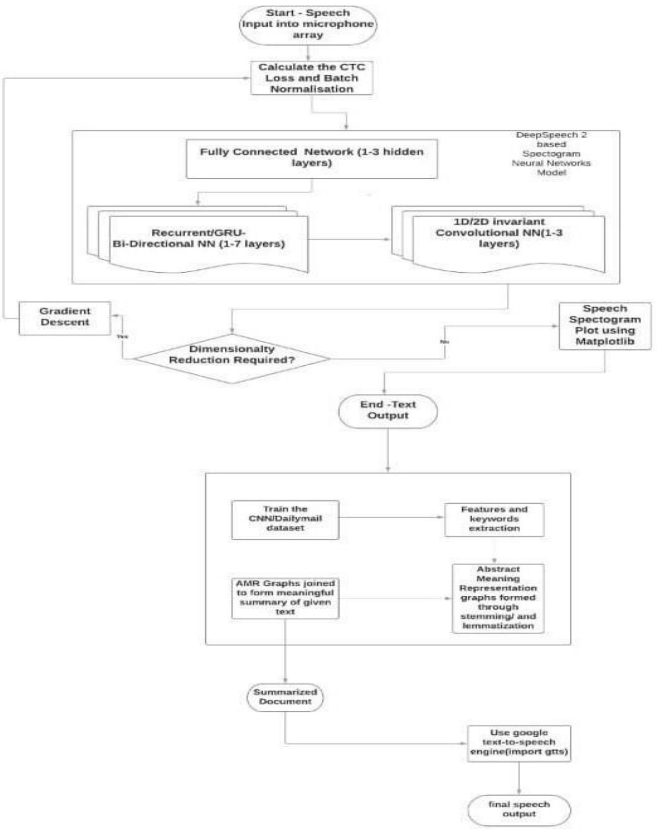

Fig 1: Architecture Diagram for the proposed model

A. Calculation Of CTC Loss :

To find the most likely word sequence, different path combinations are searched. We need to sum over all paths that generate the same word sequence. For example, to find the probability for the word "hello", we sum over all the corresponding paths like "heɛllɛlloo", "hhellɛɛlɛo", "ɛeɛllɛɛloo" etc. In Fig. 2, the CTC Loss tells the possibilities of interpreted words from the audio input.

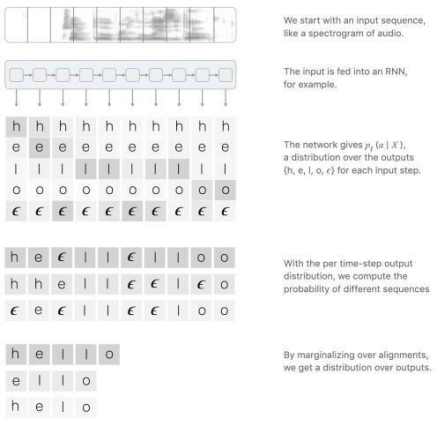

Fig 2: CTC Loss

In Fig. 3, the formula to calculate the CTC Loss Function is given.

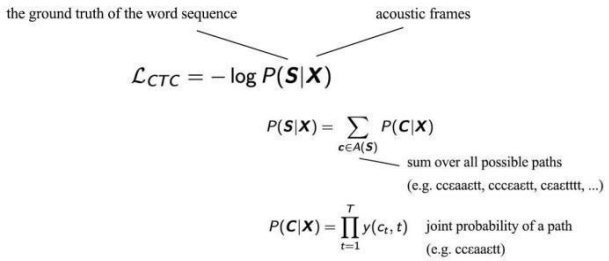

Fig 3: CTC Loss Function corresponding to the CTC Loss

B. Batch Normalization:

According to [12], Batch Normalization (BN) is a viable technique to quicken model preparing and improve the speculation execution of neural systems significantly.

According to [8], the author has explored different avenues regarding the Batch Normalization (BatchNorm) strategy to prepare these more profound neural networks quicker. BatchNorm can speedup the convergence of RNNs training. For the deep neural networks of RNNs on large data sets, the BatchNorm substantially improves final generalization error in addition to accelerating training.

C. Deep Speech 2:

It is a basic multi-layer model with a single repetitive layer that can't misuse a large number of long periods of marked discourse. In this way, to gain from vast datasets, the capacity of the model is expanded by depth.

The author in [8] has explored architectures with up to eleven layers comprising many bi-directional recurrent layers and convolutional layers R Alugubelli. (2016).et.al. These models have about multiple times the measure of calculation per information model as that of the models in Deep Speech 1, making quick improvement and computation.

Though many of the results make use of bi-directional recurrent layers, excellent models exist using only uni-directional recurrent layers. This feature makes such models significantly more regular to convey. Together, these characteristics allow the optimization of deep RNNs and improve performance by more than 40% in both English and Mandarin [8].

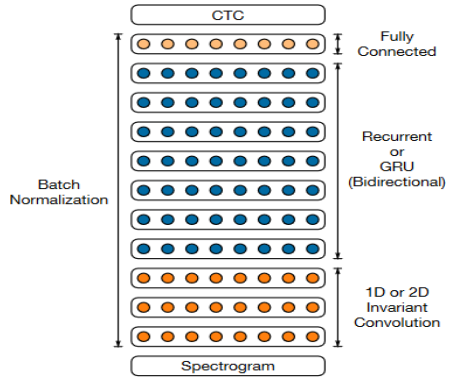

Fig 4: Architecture Diagram of the Deep Speech 2, trained on both English and Mandarin speech.

D. AMR Graphs:

AMR was formally introduced by Banarescu et al. (2013); the aim is to propose work on the statistical and the graphical Natural Language Generation. AMR represents meaning using graphs; the aim is to achieve an abstractive summary rather than extractive summaries. AMR graphs are directed labeled graphs. Fig 4 shows the AMR graph of the sentence, "I looked carefully all around me" generated by JAMR first-3 parser [13].

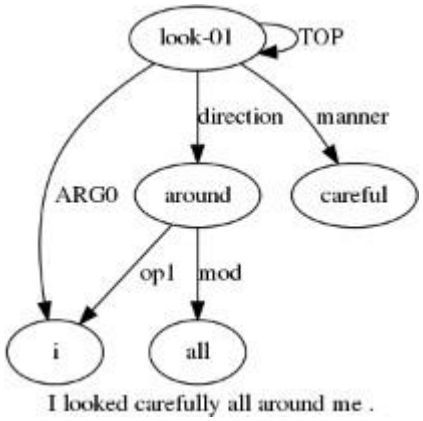

Fig 4: The Abstract Meaning Representation graph of the sentence: "I looked carefully all around me."

E. Google Text-To-Speech Engine :

The Google text-to-speech engine will convert the summarized text to speech. Being able to be used widely with a continuous internet connection, the Google text-to-speech engine is best suited for the task.

F. Abbreviations And Acronyms :

Table 1: Abbreviations used throughout the paper

NLG	Natural Language Generation
AMR	Abstract Meaning Representation
LSA	Latent Semantic Analysis
SVM	Support Vector Machine
NLP	Natural Language processing
KNN	K-Nearest Neighbor
CTC	Connectionist Temporal Classification
BatchNorm/BN	Batch Normalization
RNN	Recurrent Neural Network
WER	Word Error Rate
CNN	Cable News Network

IV. IMPLEMENTATION

The proposed system is implemented by applying the following modules:-

A. CTC Loss Function:

ASR is primarily composed of two significant steps [9]:-

1. Mapping:-
   In the mapping phase, the acoustic information of an audio frame is to be mapped to the triphone state, and this is the alignment process.
   The alignment shall map acoustic information to a phone. Then the phone sequence is searched for the optimal word sequence.

2. Searching:-

This alignment-free, one-to-one mapping maps an audio frame to a relatively high-level component that can be searched now. As a result, now this deep neural network works with the character sequence to make the further task easier and more interpretable.In Fig. 5, the overall flow for the CTC is given below.

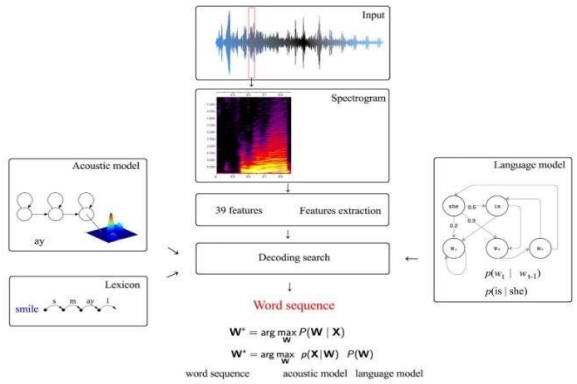

Fig 5: Various stages of the CTC to output word sequences from the audio input.

B. Batch Normalization:

Batch Normalization is done to speed up the training phase significantly. The Batch Normalization reduces the amount of the hidden unit values that shift around during the training (covariance shift).

The purpose of the Batch Normalization is to normalize the output of the previous activation layer. This is done by subtracting the batch mean and dividing by the batch standard deviation. Which, as a result, improves the training speed almost by ten times.

A recurrent layer is implemented as:

(1)

, where the activations of layer l at a time step t, is the previous layer and , is the current layer.

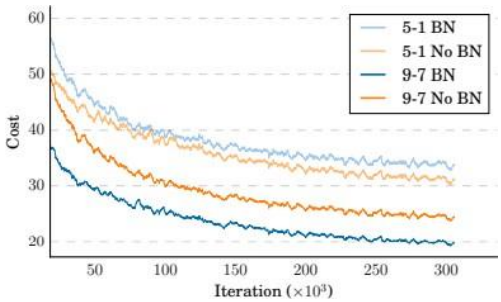

Fig 6: According to [8], this is the training curve with and without BatchNorm(BN).

C. Deep Speech 2:

The job of this neural network is to perform end-to-end speech recognition and convert it into text. The idea behind Deep speech 2 [8] was to replace decades worth hand-engineered knowledge, which is applied to the current state-of-the-art automatic speech recognition pipelines, with deep learning approaches.

The input to the neural network is a sequence of log spectrograms of audio clips that are power normalized, calculated on 20ms windows. The outputs are the alphabets of each language, at each time-step *t*, the RNN will make a prediction, i.e., p(, where *{ a, b, c, ... , z, space, apostrophe, blank}*

D. AMR Parsing:

For the purpose of summarization, we use AMR graphs with the dataset of CNN/Dailymail. The reason to choose this dataset is that the summary of the given text is given in 3-4 lines. The summarization through AMR graphs can be divided into three sequential steps. These steps are:-

Step 1: Sentences to their respective AMR Graphs

Step 2.1: Selecting important Sentences in the text

Step 2.2: Extracting the Summary Graph

Step 3: Summary Generation

E. Google Text-to-Speech Engine:

The final output should be the summarized document in audio format (.mp3 or .wav). The Google Text-to-Speech Engine offers a reliable interface. The summarized report can be directly fed to the text-to-speech engine or can be kept at the endpoint of an API to output the audio of the summarized report.The advantage of using the Google Text-to-Speech Engine is that it is available in major languages, so the proposed model can be portable to different languages (according to the input language).

V. RESULTS

For the Speech recognition system, the performance metric used is WER. The WER is Word Error Rate, which is a performance metric used to measure the performance of speech-recognition system. As for the accuracy of the summarizer, the metric used to evaluate its performance is ROGUE scores in comparison with the Lead-3-AMR-Baseline.In the given Table 2, the WER is compared on the development set with varying depths of RNN to show the application of the Batch Normalization on hidden units.

Table 2: Comparison of WER on the development set as the depth of RNN is increased, and the WER of BatchNorm is compared with that of the baseline [8].

Architecture	Baseline	BatchNorm
5-layer, 1 RNN	13.55	14.40
5-layer, 3 RNN	11.61	10.56
7-layer, 5 RNN	10.77	9.78
9-layer, 7 RNN	10.83	9.52
9-layer, 7 RNN no SortaGrad	11.96	9.78

In [8], the human-level accuracy is obtained by asking Amazon Mechanical Turk workers to hand-transcribe the test set. The results are given in Table 3.

Table 3: According to [8], WER for the speech system is compared with that of crowd-sourced human-level performance.

	Test set	Ours	Human
Read	WSJ eval'92	3.10	5.03
	WSJ eval'93	4.42	8.08
	LibriSpeech test-clean	5.15	5.83
	LibriSpeech test-other	12.73	12.69
Accented	VoxForge American-Canadian	7.94	4.85
	VoxForge Commonwealth	14.85	8.15
	VoxForge European	18.44	12.76
	VoxForge Indian	22.89	22.15
Noisy	CHiME eval real	21.59	11.84
	CHiME eval sim	42.55	31.33

Table 4 shows the ROGUE scores on CNN/Dailymail corpus. The table has two columns. The first column contains baseline's ROGUE scores with different configurations and in the second column the ROGUE scores of the used summarizer is shown.

Table 4: ROGUE scores of the CNN/Dailymail corpus, of the baseline and the proposed system.

ROGUE Score	Method	
	Lead-3-AMR (Baseline)	First-3
1 Recall	40.4	38.1
1 Precision	27.8	28.8
1	31.7	31.6
2	5.8	5.7
L	16.8	16.9

The speedup achieved by the speech recognition is 4x-21x, depending on the processing unit used, and the accuracy of the summarizer is 99.37 %.

VI. CONCLUSION AND FUTURE WORK

The best English model for Deep Speech 2 is model with two layers of 2D, and then by three-unidirectional recurrent [8]. With the use of highly-optimized kernel from Nervana Systems and NVIDIA that are specially tuned up for deep learning can give a speedup of 4x-21x.

In Table 4, the results on the CNN/Dailymail corpus, ROGUE scores are presented using the first-3 model. The results achieved are competing with the Lead-3-AMR baseline with an accuracy of 99.37%.

There still are a lot of scopes to improve upon:-

- First of all, the dataset used for summarization is CNN/Dailymail, which comprises of news articles. However, there is a need to summarize the spoken text, and there are no such suitable datasets available, which can significantly improve the output.
- There is still a need to have better quality AMR parsers and generators. Parsers and generators with better accuracy can improve the model's efficiency significantly.
- To train different models, there was extensive use of GPUs. So High-Performance Computation elements like GPUs should increase our model's accuracy.

REFERENCES

1. U. Hahn and I. Mani, "The challenges of Automatic Summarization," IEEE Volume:33 Isuue:11
2. S.Ramana, M.V.Ramana Murthy, & N.Bhaskar. (2017). Ensuring data integrity in cloud storage using ECC technique, International Journal of Advanced Research in Science and Engineering, BVC NS CS 2017, 06(01), 170–174. ISSN Number: 2319-8346
3. PeeyushMathur,NikhilNishchal, "CloudComputing:Newchallengeto the entire computer industry", 2010 1st International Conference on Parallel, Distributed and Grid Computing (PDGC -2010).
4. R Alugubelli. (2016). Exploratory Study of Artificial Intelligence in Healthcare. International Journal of Innovations in Engineering Research and Technology, 3(1), 1–10.
5. Roopha Shree Kollolu Srinivasa (2018), CLASSIFICATIONS OF WIRELESS NETWORKING AND RADIO , Wutan Huatan Jisuan Jishu, Volume XIV, Issue XI, November/2018, Page No: 29-32.
6. I. Ahmad and K. Pothuganti, "Smart Field Monitoring using ToxTrac: A Cyber-PhysicalSystem Approach in Agriculture," 2020 International Conference on Smart Electronics and Communication (ICOSEC), 2020, pp. 723-727.
7. sridevi Balne, Anupriya Elumalai, Machine learning and deep learning algorithms used to diagnosis of Alzheimer's: Review, Materials Today: Proceedings, 2021, https://doi.org/10.1016/j.matpr.2021.05.499.
8. MALYADRI KORIPI(2021), "5G VISION AND 5G STANDARDIZATION" Parishodh JournalVolume X, Issue III, March/2021 Page No: 62-66
9. MALYADRI KORIPI (2021), "A REVIEW ON SECURE COMMUNICATIONS AND WIRELESS PERSONALAREA NETWORKS(WPAN)" Wutan Huatan Jisuan Jishu, Volume XVII, Issue VII, July/2021, page. 168-174
10. Roopha Shree Kollolu Srinivasa (2020), A REVIEW ON WIDE VARIETY AND HETEROGENEITY OF IOT PLATFORMS, The International journal of analytical and experimental modal analysis, Volume XII, Issue I, January/2020 , Page No:3753-3760.
11. N.Bhaskar, S.Ramana, & M.V.Ramana Murthy. (2017). Security Tool for Mining Sensor Networks . International Journal of Advanced Research in Science and Engineering, BVC NS CS 2017, 06(01), 16–19. ISSN Number: 2319-8346.
12. MALYADRI KORIPI(2021), "A REVIEW ON ARCHITECTURES AND NEEDS IN ADVANCED WIRELESSCOMMUNICATION TECHNOLOGIES" A Journal Of Composition Theory, Volume XIII, Issue XII, DECEMBER 2020, Page No: 208-214.
13. Roopha Shree Kollolu Srinivasa (2020), INFRASTRUCTURAL CONSTRAINTS OF CLOUD COMPUTING, International Journal of Management, Technology and Engineering, Volume X, Issue XII, DECEMBER/2020, Page No: 255-260.

14. S.A. Babar and Pallavi D. Patil, “Improving Performance of Text Summarization,” Elsevier, Procedia Computer Science Volume 46, 2015.