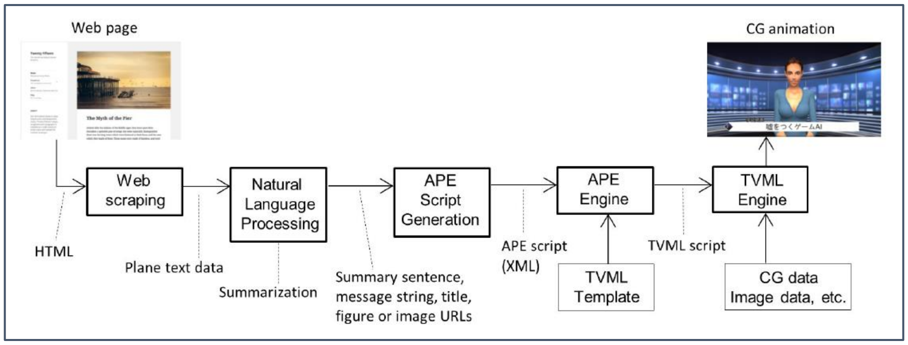
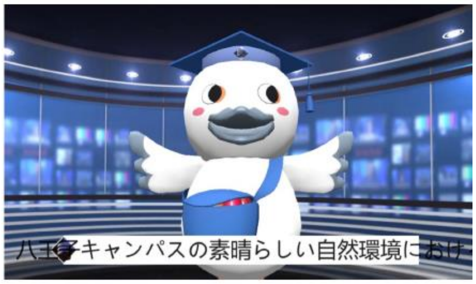
Research Paper- 1

<table>
<tr><td><b>Paper Serial</b></td><td colspan="3"></td></tr>
<tr><td><b>Paper Name</b></td><td colspan="3">Automatic Generation of News Content from Blog Posts</td></tr>
<tr><td><b>Conference/Journal Name</b></td><td colspan="3">International Journal of Asia Digital Art & Design</td></tr>
<tr><td rowspan="2"><b>Journal Category</b></td><td rowspan="2">W</td><td><b>Year of Publication</b></td><td>2021</td></tr>
<tr><td><b>Language</b></td><td>English</td></tr>
<tr><td><b>Objective of the Research</b></td><td colspan="3">This Paper describes the development of an application which makes use of TVML (TV programs Making Language) to create news scripts using the data scrapped from web and the different TVML techniques such as camera angle, artwork etc.(such techniques would be used to generate appropriate pictures suitable to the content of the news). In the end it applies a CG character to present the news.</td></tr>
<tr><td><b>Major Contribution/ Gaps Addressed</b></td><td colspan="3">This paper aimed to automate the system for news generation and presentation to address the journalism related issues such news biasness, fake news etc.</td></tr>
<tr><td><b>Approach/ Method/ Technique</b></td><td colspan="3">The system starts with scrapping data using a web scrapper. This is the plane text which is then passed from NLP Processing (to summarize it). That summarized sentences, title etc. are passed from an APE Script generator and then an APE engine to generate a TVML script. This TVML script is then passed through TVML Engine and is presented using a CG character.
 </td></tr>
<tr><td><b>Application Domain</b></td><td colspan="3">Open Domain</td></tr>
<tr><td><b>Data Set Details</b></td><td colspan="3"></td></tr>
<tr><td><b>Experimental Setup</b></td><td colspan="3">2 experiments were conducted to define the accurateness of the system. The first one aimed to obtain a message string and a message title. In this 53 articles were given to the system and the system was supposed to generate the content from that.
The second experiment aimed at automatically developing suitable artwork, sound effects etc. for the given news script.</td></tr>
<tr><td><b>Evaluation/Testing</b></td><td colspan="3">-</td></tr>
</table>

Technique	
Results	The result for first experiment showed that out of 22 generated strings (of sentences), 12 strings were correct i.e. it gave an accuracy of 55%. For second experiment the result did generate some CG characters and text strings to appear when the news channel starts for example an output screen is given down below: 
Limitations/ Assumptions	-
Future Directions/ Open Issues	Their future work includes integrating all of the elements of the news and generating images that are relatable to the news content and generating a more natural CG character as compared to just an avatar.

Research Paper-2

Paper Serial	R5 – S – NER		
Paper Name	FLAIR: An Easy-to-Use Framework for State of the Art NLP		
Conference/Journal Name	https://aclanthology.org/N19-4010.pdf		
Journal Category	NLP	**Year of Publication**	2019
		Language	English
Objective of the Research	Easy Framework for NLP		
Major Contribution/ Gaps Addressed	As compare to others The core idea of the framework is to present a simple, unified interface for conceptually very different types of word and document embeddings. This effectively hides all embedding-specific engineering complexity and allows researchers to “mix and match” various embeddings with little effort		

Approach/ Method/ Technique	A PyTorch NLP framework: Framework builds directly on PyTorch, making it easy to train your own models and experiment with new approaches using Flair embeddings and classes.
Application Domain	Open Domain
Data Set Details	FLAIR includes convenience methods for downloading standard NLP research datasets and reading them into data structures for the framework. It also includes model training and hyperparameter selection routines to facilitate typical training and testing workflows. In addition, FLAIR also ships with a growing list of pre-trained models allowing users to apply already trained models to their text
Experimental Setup	FLAIR only requires a current version of Python (at least version 3.6) to be available on a system or a virtual environment. Then, the simplest way to install the library is via pip, by issuing the command: pip install flair. This downloads the latest release of FLAIR and sets up all required libraries, such as PYTORCH. Alternatively, users can clone or fork the current master branch of FLAIR from the GitHub repository. This allows users to work on the latest version of the code and create pull requests. The GitHub page1 has extensive documentation on training and applying models and embedding.
Evaluation/Testing Technique	
Results	https://github.com/flairNLP/flair
Limitations/ Assumptions	Not all word and document embeddings currently supported by FLAIR, such as class CharacterEmbeddings.
Future Directions/ Open Issues	Can be used to create in more language.

Research Paper-3

Paper Serial			
Paper Name	End-to-end Conversion Speed Analysis of an FPT.AI-based Text-to-Speech Application.		
Conference/Journal Name	Chung2020.pdf		
Journal Category	W	**Year of Publication**	2020
		Language	English
Objective of the Research	The main objective of the research is to design an application that takes three inputs from user i.e. text, speech, and voice and convert the text into speech and returns a converted audio file as well. It's mainly focused on the relationship between the end-to-end conversion time and the length of the		

	input text.
Major Contribution/ Gaps Addressed	Creating a text-to-speech (TTS) application that is linked with FPT.AI server via it's API, to convert the text into seven different Vietnamese speeches. Also they have addressed that end-to-end conversion time is depends on the length of the input text i.e. if the length of text increases then the time taken by the TTS to perform task will be increases as well. MOS (main opinion source) is one of the parameter of TTS system that is used to measure the naturalness of the generated speech.
Approach/ Method/ Technique	To achieve their goals, they propose an approach that uses FPT.AI API to linked the connection between local host and remote FPT TTS server. In this approach TTS API gets four arguments from the user to generate http request before posting it to the server i.e. the POST data that holds the text that is converted into speech, the second input is the speed of voice and the voice category. Once it'll gets the input it'll generate the request to the server. For every request server will returned the response to host application, and provide a http link to downloads the converted audio file in *.mp3.
Application Domain	Restricted Domain.
Data Set Details	In the proposed article the application is initialized as a localhost so they use FPT.AI API to linked the connection between local host and remote FPT TTS server.
Experimental Setup	The getVoice's algorithm is modified as follows:
Evaluation/Testing Technique	After evaluation they have measured the performance of the system that, as far as the length of input text is directly proportional to the end-to-end conversion time to obtained the converted speech i.e. 500 character input text will take 9s to 10s to converted into speech as compare to 400 character input text.
Results	https://fpt.ai/tts
Limitations/ Assumptions	The Assumptions of the applications are: - The length of input text is based on 10, 100, 200, 300, 400, and 500 characters. - They have set the speed of voice to zero by default. - They have set the Thu Dung voice as the default voice of the application.
Future Directions/ Open Issues	They will improve their GUI in order to make it more attractive for users and will improve its voice quality.

- When the system started it gets the start time after that it obtains the data, voice and speed from the user.
- After that it'll prepare payload header and then it'll set timeout = 0

- And then send a TTS request, and checks that does the response have rate limit exceeded or not? If yes, then it processed to context link and get the end time. And if no, then it'll again check that does the response succeed?

Research Paper-4

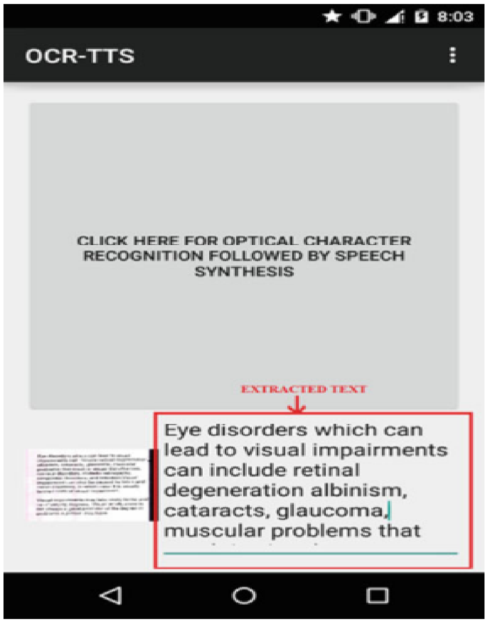
Paper Serial	2347-2693		
Paper Name	Natural Language Processing		
Conference/Journal Name	JCSE International Journal of Computer Sciences and Engineering		
Journal Category	NLP	**Year of Publication**	2018
		Language	English
Objective of the Research	Differentiate between various algorithms and future scope of research in NLP		
Major Contribution/ Gaps Addressed	It is a review paper which is providing basic knowledge about all algorithms used in NLP		
Approach/ Method/ Technique	**Algorithm used for efficiency of processing the language in text:** • Long short-term memory • Sequence 2 Sequence model • Named Entity Recognition model • User preference graph model • Word Embedding model • Feature based sentence extraction using fuzzy inference rules. • Template based algorithm using automatic text summarization **Algorithm for speech form efficiency**: • Word Recognition • Acoustic Modeling • Connectionist temporal classification • Phase based machine translation • Neural machine translation • Google neural machine translation		
Application Domain	Open Domain		
Data Set Details	Algorithm: Connectionist Temporal Classification (CTC): Turning Sounds into Bits: This is called sampling. By Nyquist Theorem if we sample at least twice as fast as the highest frequency we can recover the original signal back. For speech recognition a sampling rate of 16,000 samples per second is optimal. After this there would be array of numbers with each number representing the sound wave's amplitude at 1/16000 a second intervals. We could directly give the sampled data to neural network but finding the pattern in such a large dump of data would be difficult and require lot of computations. Increasing the time complexity of the algorithm. So Pre		

	Processing is done
Experimental Setup	Feature based sentence extraction using fuzzy inference rules: The stated algorithm is based on evaluating a sentence in the input data on basis of some rules which categorize the statement and assign those values as low, medium and high. The algorithm consists of 4 stages which process the data and gives the final output as the processed summary. These stages are: first is Preprocessing then Feature extraction followed by Fuzzy logic scoring And Sentence selection and assembly Template based algorithm for automatic text summarization: As we saw that in feature-based algorithm the sentences are evaluated on basis of some basic criteria or basic features and the sentences as arranged in the input data are added as same in the output summary while in case of template-based extraction algorithm some modification is being done after extracting the text to arrange the information in a proper and grammatical way to arrange in the summary and to make it more look like a human work. The template-based algorithm for automatic text summarization is implemented in two phases A. Text pre-processing B. And information extraction
Evaluation/Testing Technique	Algorithm: Feature based sentence extraction using fuzzy inference rules This algorithm used certain features to determine the importance of the sentence on which it is included into the summary. These features are 1. Title feature 2. Term weight 3. Sentence length 4. Sentence position 5. Thematic word 6. And fuzzy logic On the basis of these features the sentence is evaluated by the algorithm and the output is generated.
Results	Different algorithms are being used to improve NLP Different AI now use NLP algorithms to recognize and process the voice command given by user.
Limitations/ Assumptions	**Limitations of algorithms:** -The disadvantage of PBMT is that it is difficult to build and maintain. If there is a need to add a new language than bilingual corpora of that language should be present. -The limitations of NMT are ambiguous words into German. It also issues with forming verb continuous tenses. Dominant problem for NMT are prepositions. -Limitation of Acoustic Modeling: These models take lots and lots of training data to create and for many users they work just fine. However, for many others they do not. This is because the data used to generate the model contains samples from tens of thousands of different speakers, so they are generic. Making specific models for individuals is not economical, neither is making models for accents with small populations. Acoustic models are a

	limitation of the technology. -Limitation of CTC algorithm: One disadvantage of this algorithm is that if the input audio file is of HULLO then the algorithm would not be able to recognize it correctly since the database of written text does not contain a greater number of HULLO. So, the algorithm would malfunction when the reader says words which aren't present in the database of written text.
Future Directions/ Open Issues	Easily modification can be made for further enhancements in the algorithms

Research Paper-5

Paper Serial			
Paper Name	Portable Text to Speech Converter for the Visually Impaired		
Conference/Journal Name	Soft computing system volume 397		
Journal Category	W	**Year of Publication**	2016
		Language	English
Objective of the Research	The main objective of this research is to design an application which helps the partially vision loss people by scanning the image from any book or document and then convert the extracted text into speech form.		
Major Contribution/ Gaps Addressed	Creating an application which helps the people who are partially vision loss. The people who have partial vision loss needs the third person to read for them, therefore, the need to put light on this matter they implement an application that scans the page containing text and convert the text into speech.		
Approach/ Method/ Technique	To achieve their goals, they apply a technique i.e. firstly the scanner scanned the page containing text, graphs or images, etc. then it sends it to an android application which is connected through Bluetooth module. After sending it to android application the Tesseract optical character recognition (OCR) library will extract the plain text from the scanned image and once the text is extracted then the library text to speech(TTS) is used to convert the text into speech.		
Application Domain	Android mobile application.		
Data Set Details	In the proposed article the application transfer scanned file to android phone over Bluetooth.		
Experimental Setup	The Experimental setup works as firstly the scanner scan the document, then transfer scanned file to android phone over Bluetooth them its check whether the file was transfer successfully or not? If no, then it will retransfer it or else it will open the most recently received file in the application. After it will extract the text from the opened file in the application and convert extracted text into speech.		
Evaluation/Testing Technique	--------		

Results	 OCR-TTS CLICK HERE FOR OPTICAL CHARACTER RECOGNITION FOLLOWED BY SPEECH SYNTHESIS OCR . 4 System model of portable text to speech converter **Fig. 5** Screenshot of the android application  8:03 OCR-TTS CLICK HERE FOR OPTICAL CHARACTER RECOGNITION FOLLOWED BY SPEECH SYNTHESIS EXTRACTED TEXT Eye disorders which can lead to visual impairments can include retinal degeneration albinism, cataracts, glaucoma, muscular problems that
Limitations/ Assumptions	The Assumptions of the applications are: - The scanner should be placed in horizontal to the scanned document. - The application is only for the android mobile phones and not for the apple iPhone OS. - Android phone must be within the range of Bluetooth of the scanner.
Future Directions/ Open Issues	------

Research Paper-6

Paper Serial	2248-9622
Paper Name	Comparing Speech Recognition Systems (Microsoft API, Google API And CMU Sphinx)
Conference/Journal Name	www.ijera.com

Journal Category		**Year of Publication**	2017
		Language	English
Objective of the Research	Comparing Different Speech Recognition Systems		
Major Contribution/ Gaps Addressed	Providing research regarding how Google API is Superior than other recognition systems		

Approach/ Method/ Technique	CMU SPHINX version-4 three main components in the Sphinx-4 structure, which includes the Frontend, the Decoder and the Linguist -Frontend implementations support MFCC, PLP, and LPC feature extraction. -the Linguist implementations support a variety of language models, including CFGs, FSTs, and N-Grams; and the Decoder supports a variety of Search Manager implementations. Microsoft has focused on increasing emphasis on speech recognition systems and improved the Speech API (SAPI) by using a context dependent deep neural network hidden Markov model (CD-DNN-HMM) THE GOOGLE API Google has improved its speech recognition by using a new technology in many applications with the Google App such as Goog411, Voice Search on mobile, Voice Actions, Voice Input (spoken input to keypad), Android Developer APIs, Voice Search on desktop, YouTube transcription and Translate, Navigate, TTS.
Application Domain	Open Domain
Data Set Details	TESTING DATA The audio files were selected from various sources to evaluate the Microsoft API, Google API, and Sphinx-4. According to CMUSphin, Sphinx-4's decoder supports only one of the two specific audio formats (16000 Hz / 8000 Hz). WAV files not supported by all three so they design tool which recognition all audio files in the same format (16000 Hz / 8000 Hz). The TIMIT corpus of read speech is designed to provide speech data for acousticphonetic studies and for the development and evaluation of automatic speech recognition systems. “The TIMIT corpus includes timealigned orthographic, phonetic and word transcriptions as well as a 16-bit, 16kHz speech waveform file for each utterance. Corpus design was a joint effort among the Massachusetts Institute of Technology (MIT), SRI International (SRI) and Texas Instruments, Inc. (TI)”
	audio files from ITU (International Telecommunication Union) which is the United Nations Specialized Agency in the field of telecommunications.

Experimental Setup	In this paper they have developed a tool that is used to test these models in Microsoft API, Google API, and Sphinx4. Also, they calculated the WER by using this tool to recognize a list of sentences, which we collected in the form of audio files and text translation. In this paper, they follow these steps to design the tool and test Microsoft API, Google API, and Sphinx-4. This system has been designed by using the Java language, which is the same language that has been used in Sphinx-4, as well as the C# that was used to test the Microsoft API and Google API. Libraries such as Text to Speech API, Graph API and Math API for different tasks. Moreover, this tool was connected with the classes of Sphinx4, Microsoft API and Google API to work together to recognize the audio files. Then we compared the recognition results with the original recording texts. calculated the word rate (WER) and accuracy according to these equations. $\text{WER} = (I + D + S) / N$ $\text{WER} = (0 + 0 + 1) / 9 = 0.11$ where I words were inserted, D words were deleted, and S words were substituted. The original text (Reference): the small boy PUT the worm on the hook The recognition text (Hypothesis): the small boy THAT the worm on the hook $\text{Accuracy} = (N - D - S) / N$ $\text{WA} = (9 + 0 + 1) / 9 = 0.88$
Evaluation/Testing Technique	Evaluation is based on the three API through creating a tool and running them through same audio file. After testing its been notice Google's acoustic model is best than others
Results	The Sphinx-4 (37% WER), Google Speech API (9% WER) and Microsoft Speech API (18% WER). Where S sentences, N words, I words were inserted, D words were deleted, and S words were substituted. CW correct words, EW error words Google is Superior
Limitations/ Assumptions	-THE GOOGLE API Google achieved an 8 percent error rate in 2015 that is reduction of more than 23 percent from year 2013
Future Directions/ Open Issues	-Its open for any other API or improvement in the existing APIs.