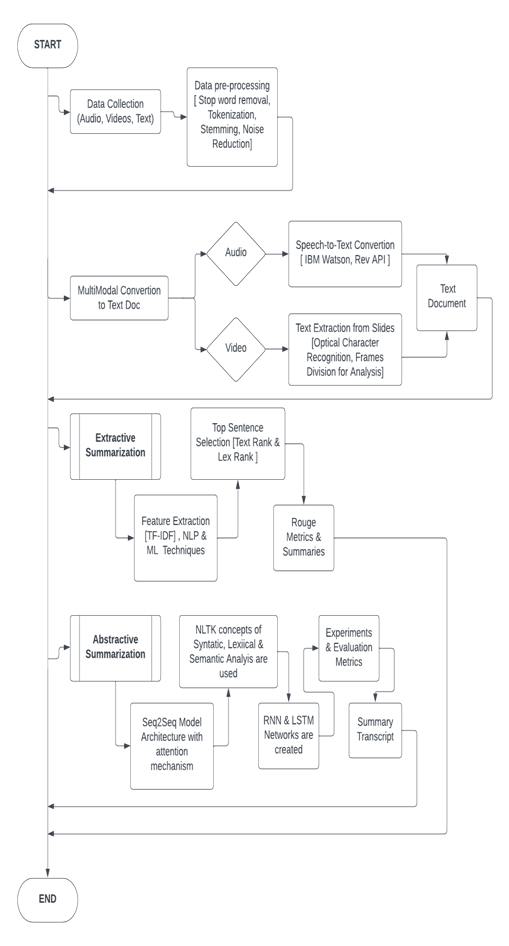
**TITLE**: Business Meeting Summary Generation Using NLP

**Team Number**: CSM07

**Team Members**: A. Dheeraj Reddy, A. Chinmaye, D. Saketh, Mohammed Irfan

**Github Link**: https://github.com/dheeraj2804/Business-Meeting-Summary-Generation

**Project Overflow:**

****

**Objectives**:-

* Develop tailored Summaries for better understanding and resourceful transcription   
   1. Extractive Summary [ General conversation meeting ]

2. Abstractive Summary [ More advanced jargon based meeting which includes techniqalities and other things ]

* Considering multi-media data inputs for more robustness.
* Develop a system that balances the accuracy and speed of generated summaries.

**Work Progress:-**

1. **Work Completed**

Completed Extractive Summarization Code using spacy environment and NLP techniques.

**Base Code:**

**import spacy**

**from spacy.lang.en.stop\_words import STOP\_WORDS**

**from string import punctuation**

**from heapq import nlargest**

**text = """In the fast-paced world of contemporary business, effective communication and information dissemination are crucial elements for success. Business meetings serve as vital platforms for collaboration, decision-making, and strategy formulation. However, the wealth of information exchanged during these meetings often poses a challenge for professionals to efficiently distill and retain key insights. Recognizing the growing significance of addressing this challenge, the field of business meeting summary generation has emerged as a focal point for research and development. Business meetings can last for a long time, and it can be challenging to capture the crucial details of any given discussion. It has been shown that traditional note-taking techniques are less beneficial. """**

**def summarizer(rawdocs):**

**stopwords = list(STOP\_WORDS)**

**# print(stopwords)**

**nlp = spacy.load('en\_core\_web\_sm')**

**doc = nlp(rawdocs)**

**#print(doc)**

**tokens = [token.text for token in doc]**

**#print(tokens)**

**word\_freq = {}**

**for word in doc:**

**if word.text.lower() not in stopwords and word.text.lower() not in punctuation:**

**if word.text not in word\_freq.keys():**

**word\_freq[word.text] = 1**

**else:**

**word\_freq[word.text] += 1**

**# print(word\_freq)**

**max\_freq = max(word\_freq.values())**

**# print(max\_freq)**

**for word in word\_freq.keys():**

**word\_freq[word] = word\_freq[word]/max\_freq**

**# print(word\_freq)**

**sent\_tokens = [sent for sent in doc.sents]**

**# print(sent\_tokens)**

**sent\_scores = {}**

**for sent in sent\_tokens:**

**for word in sent:**

**if word.text in word\_freq.keys():**

**if sent not in sent\_scores.keys():**

**sent\_scores[sent] = word\_freq[word.text]**

**else:**

**sent\_scores[sent] += word\_freq[word.text]**

**#print(sent\_scores)**

**select\_len = int(len(sent\_tokens) \* 0.3)**

**# print(select\_len)**

**summary = nlargest(select\_len, sent\_scores, key = sent\_scores.get)**

**# print(summary)**

**final\_summary = [word.text for word in summary]**

**summary = ' '.join(final\_summary)**

**#print(text)**

**print(summary)**

**#print("Length of original text ",len(text.split(' ')))**

**#print("Length of summary text" , len(summary.split(' ')))**

**return summary, doc, len(rawdocs.split(' ')), len(summary.split(' '))**

1. **Need to-be Completed**

* Need to complete the first phase of the project where the multimedia data like audio and video(Slideshow PPT’s) inputs need to be taken and converted to Text transcript for further processing in the next phases.  
  **Technologies using**: IBM Watson, Hugging face transformer based model like VOSK or Rev-AI audio to text and OCR.
* Need to complete the Abstractive Summarization part where neural network models need to be trained for summarizing the text context to produce unique content.

**Technologies using**: LSTM-RNN cells, Seq-to-Seq Models and Attention Mechanism.

* Finally test the project and develop a static page where a user uploads their meeting and can access the meeting summary.

\*\*\*\* Need assistance in integrating the front-end and back-end part and provide guidance in technical aspects of audio and video to text transcription as well as with the Abstractive summarization part. \*\*\*\*