PROJECT PLAN DOCUMENT

Team number	***46***
Project Title	***Summarization of Intelligent Caller Response (ICR) System Responses***
Document	Project Plan
Creation date	***2023/02/17***
Created By	***Adyansh Kakran, Rohan Girish, Roja Sahoo, Sarthak Chittawar***
Client	***Arjun Rajasekar, RCTS***

Brief problem statement

The intelligent caller response (ICR) is a multi-organization effort, to create a voice based smart response system capable of answering questions in Indian vernacular languages. The existing system architecture consists of an automated speech recognition (ASR) module, machine translation (ML) module, a semantic document search engine, and a text-to-speech (TTS) module. The larger project is currently under the data collection and model adaptation phase. The current effort is to develop a module to summarize the responses from the document search engine, to produce more "human friendly" responses. This will be limited to the use of existing summarization engines and exploring input content requirements to output quality. The major development is on building the API wrappers to the chosen ML engines and a simple UI for the purposes of demonstration.

Team Members

Adyansh: Handling the website interface for communicating with the models.

Rohan: Finding and testing abstractive summarization models.

Roja: Finding and testing extractive summarization models.

Sarthak: Compiling test cases and expected output for the summarization models.

These are just broad ideas of what each team member oversees. Every team member works on every aspect of our project.

Team Communication

1. Weekly Meetings every Friday at 10am
2. Discord Server with TA and WhatsApp group without.
3. Mailing List with both client and TA.

Development Environment

1. MERN Stack for website.

2. GPT3, BERTSUM, HuggingFace, NLTK, other Summarization models.
3. Jupyter Notebooks – to make it easier to visualize the data.
4. Scikit, Pandas, other Python libraries.
5. FastAPI – for API calls

Milestone Schedule

Milestone	Due Date	Release	Deliverable?
Take some common sample data and use different text summarizers to summarize the data. Observe the differences in each of these engine outputs and spot their strengths and weaknesses.	Last week of Feb	NA	No
Identify the best summarizer and then make a basic UI for input/output of the text summarizer. The summarizer must be connected at the backend. (Only basic summarizer, detai debugging not required)	1st/2nd week of March	R1	No
From the data variation observed, choose a particular suitable domain for the text summarizer and make amendments to the existing application	In next 2-3 weeks	R2	Yes