

**CHATBOT SYSTEM FOR COMPUTERS, ACCESSORIES
& REPAIR CENTER RECOMMENDATION**

Final Report
R.P.W.G. Rathnaweera

(IT20237554)

B.Sc. (Hons) Degree in Information Technology Specialized in Data
Science

Department of Information Technology

Sri Lanka Institute of Information Technology
Sri Lanka

September 2023

CHAT-BOT SYSTEM FOR COMPUTERS, ACCESSORIES & REPAIR CENTER RECOMMENDATION

R.P.W.G. Rathnaweera

(IT20237554)

The dissertation was submitted in partial fulfillment of the requirements for the
BSc (Hons) in Information Technology Specializing in Data Science

Department of Information Technology


Sri Lanka Institute of Information Technology
Sri Lanka

September 2023

DECLARATION

I declare that this is my own work, and this proposal does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to the Sri Lanka Institute of Information Technology, the nonexclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic, or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Name	Student ID	Signature
Rathnaweera R.P.W.G	IT20237554	

The above candidates are carrying out research for the undergraduate Dissertation under my supervision.

Signature of the Supervisor
(Dr. Lakmini Abeywardhana)

Date

.....

.....

ABSTRACT

In today's rapidly evolving digital landscape, the ubiquity of electronic devices has become an essential facet of modern life. To ensure that individuals acquire the most suitable devices tailored to their needs, people traditionally sought assistance or embarked on exhaustive device searches through various means. This necessitates the creation of a centralized solution to streamline the process of identifying optimal devices based on specific requirements. This thesis introduces an innovative approach: a chat-bot recommendation system for computers, accessories, and repair centers. This novel method leverages Large Language Models to comprehend user preferences and provide them with tailored recommendations. Given the dynamic nature of the digitized world, chat-bot systems are well-positioned to engage users effectively. Moreover, this study aims to offer comprehensive solutions to users by aggregating device and repair center reviews. This approach encompasses web scraping from select websites to collect pertinent data and incorporates image processing techniques for seamless identification of computer accessories. These features will be accessible via the proposed chat-bot system, enabling individuals with varying levels of electronic device knowledge to find solutions effortlessly.

Keywords: Chat-Bot, Image Processing, Recommendations, Automatic Speech Recognition

Table of Contents

DECLARATION	iii
ABSTRACT	iv
LIST OF FIGURES	vii
LIST OF TABLES	viii
LIST OF APPENDICES	ix
LIST OF ABBREVIATIONS	x
1. INTRODUCTION	1
1.1 Background & Literature survey	3
1.2 Research Gap	5
1.3 Research Problem	8
2. OBJECTIVES	10
2.1 Main Objectives	10
2.2 Specific Objectives	10
2.2.1 Identification of websites to be Scraped.	10
2.2.2 Custom Dataset Creation	11
2.2.3 Preprocessing of scraped data	11
2.2.3 Google Maps Location Scraper	11
2.2.4 Automated Review Summaries	12
2.2.5 Identification of language and domain on speech data.	12
2.2.7 Evaluation Matrices to evaluate the system.	13
3. METHODOLOGY	14
3.1 CTC Architecture Approach	14
3.2 Sequence to Sequence Approach.	16
3.2.1 Dataset utilized.	16
3.2.2 Data Preprocessing	16
3.2.3 Model Architecture	19
3.2.4 Challenges and solutions.	21
3.3 System Architecture	22
3.3.1 User Interaction	23

3.3.2 Chatbot Integration	23
3.3.3 Natural Language Processing (NLP)	23
3.3.4 Recommendation System	24
3.3.5 User Responses	24
3.3.6 Image Recognition	24
3.3.7 Database Utilization	24
3.3.8 YouTube Integration	25
3.4 Commercialization of the Product	25
3.4 Software solution	26
3.5 Requirement gathering and analysis	27
3.5.1 Requirements gathering	27
3.5.2 Functional Requirements	28
3.5.3 Non-Functional Requirements	29
3.5.4 Feasibility study (Planning)	29
3.6 Tools and Technologies	30
3.6.1 Tools	30
3.6.2 Technologies	31
3.7 Implementation	31
3.8 Deployment	32
4 WORK BREAKDOWN STRUCTURE AND TIMELINE	33
5 GANTT CHART	34
6 RESULTS AND DISCUSSIONS	35
6.1 Results	35
6.2 Discussions	37
6.3 Future Work	38
7 REFERENCES	39
8 APPENDICES	41

LIST OF FIGURES

Figure 1 - Usage of ASR.....	14
Figure 2 - Model Architecture Code	15
Figure 3 - Loading Dataset.....	16
Figure 4 - Resampling the loaded dataset code.....	17
Figure 5 - Chunking audio clips to 30 seconds Code.	18
Figure 6 - Define Data collector code.....	18
Figure 7- Sequence to Sequence Architecture	19
Figure 8 - Model Training Code	20
Figure 9 - System Overall Architecture	22
Figure 10 - Logo of proposed system	26
Figure 11 - Word Accuracy Equation.....	36

LIST OF TABLES

Table 1 - Existing system and feature comparison7
Table 2 - Accuracy Comparison37
Table 3 - Automatic Speech Recognition Model Results37
Table 4 - Approach Comparison.....38

LIST OF APPENDICES

Appendix A: Sample Questionnaire41

LIST OF ABBREVIATIONS

Abbreviation	Description
ML	Machine Learning
IT	Information Technology
NLP	Natural language Processing
CNN	Convolutional Neural Network
NER	Named-entity recognition
SRS	System Requirements Specifications
ASR	Automatic Speech Recognition
API	Application Programming Interface
GCP	Google Cloud Platform
WBS	Work Breakdown Structure
RNN	Recurrent Neural Network
DNN	Deep Neural Network
CTC	Connectionist temporal classification
WER	Word Error Rate
CER	Character Error Rate
Seq2Seq	Sequence to Sequence

1. INTRODUCTION

In today's modern digitalized era, the proliferation of technology and online platforms has revolutionized the way people meet their diverse needs through e-commerce websites and platforms. While this technological advancement has empowered users to effortlessly browse, compare, and purchase a wide array of products, a significant portion of users still grapple with the challenge of finding the most suitable products. This struggle can be categorized into distinct facets, the most prominent being the need to define their purpose and requirements clearly. Users must ascertain whether a product aligns with their intended purpose or if alternative options need consideration. Additionally, budget considerations can introduce further complexity, as users weigh the cost of one product against another, leading to uncertainty about whether to proceed with a purchase all these variables are intricately interrelated, with one factor often influencing the others.

Addressing this multifaceted challenge, chatbot systems emerge as a viable solution. Chatbot systems are software programs crafted to engage users in conversations that mimic human interactions. As technology evolves, chatbots have gained immense popularity and prowess. They excel in interacting with users, aiding, and offering product recommendations tailored to individual needs. By harnessing the capabilities of chatbot systems, users can mitigate their struggles when attempting to make informed purchase decisions.

Notably, recent advancements in technology, particularly in Automatic Speech Recognition (ASR), have further enhanced the capabilities of chatbot systems. ASR technology allows users to interact with the chatbot using voice queries, significantly elevating the user experience. This voice-driven approach increases user satisfaction by providing a more natural and convenient means of communication. Users can now

articulate their queries and preferences verbally, enabling the chatbot to comprehend and respond effectively.

The chatbot recommendation system, enriched with ASR functionality, acts as a knowledgeable guide, akin to a wise elder, helping users navigate the labyrinth of product choices. This system functions by gathering information about a user's purpose, budget constraints, and other pertinent factors provided by the user, whether in text or voice form. It then leverages this data to recommend the most suitable products and services, thereby simplifying the decision-making process.

In this research endeavor, our focus centers on the exploration of chatbot recommendation systems, particularly their efficacy when coupled with Automatic Speech Recognition (ASR) technology, in addressing the diverse needs of users. Our investigation is rooted in a comprehensive analysis of the factors influencing customer satisfaction within these systems, shedding light on their potential to enhance day-to-day activities through increased efficiency and effectiveness. Our ultimate objective revolves around elevating the user experience, fortifying user satisfaction, and making meaningful contributions to the domain encompassing computers, accessories, and repair centers.

1.1 Background & Literature survey

Automatic Speech Recognition (ASR) technology plays a pivotal role in our proposed research endeavor, and it is crucial to discuss its significance before delving into the broader context. ASR technology, designed to transcribe spoken language into written text, stands at the forefront of our research efforts, and we aim to develop it to cater specifically to the local Sri Lankan population, addressing of South Asian English accents. This localized ASR component holds immense promise in facilitating speech-to-text capabilities, making it a cornerstone of our research.

As undergraduate students with a penchant for technology, we identified everyday issues related to product purchases and computer accessory repair centers. This exploration led us to recognize a significant gap in the landscape of computer accessories and repair centers in Sri Lanka.

In the modern technological era, users increasingly rely on online and video reviews to inform their purchasing decisions. Platforms like Google, YouTube, and Reddit serve as valuable resources for gathering insights into the pros and cons of various products. However, our analysis revealed that local websites catering to computer retail shops [1] lacked comprehensive review-based recommendation systems, leaving users with limited options for making informed choices.

Moreover, these platforms failed to elicit specific user requirements, necessitating physical visits to computer accessory shops, which often resulted in communication gaps and potential misunderstandings.

Another pressing issue we observed was the abundance of computer accessories repair centers across Sri Lanka, coupled with the scarcity of positively rated ones [2]. Although local websites listed these repair centers, they offered scant positive reviews and lacked

proximity-based search features [3]. This discrepancy further underscored the gaps in the computer accessories domain.

The ubiquity of geo-location services and Google Maps in people's daily lives presented an opportunity to leverage these tools for location-based searches and reviews. This application had the potential to help users identify the best computer accessories repair centers, addressing a critical need.

Additionally, we identified a significant omission in the form of image recognition capabilities on Sri Lankan computer accessories retail websites [4]. The absence of a system allowing users to upload images for product recognition and recommendations presented a clear need for innovation.

In response to these real-world challenges, we conceived a comprehensive solution. Our envisioned system would employ a chatbot for user interactions, a recommendation system for product and repair center suggestions, and a video-to-speech system for transcribing speech in video reviews.

Within the domain of modern technology, chatbots have emerged as versatile systems capable of simulating human conversations through messaging [5]. They play an essential role in user interactions and requirement identification.

The recommendation system is another critical component, designed to provide tailored suggestions based on user requirements [6]. In our system, it would be instrumental in recommending products, accessories, and repair centers, drawing insights from user reviews. The accuracy of our overall system significantly hinges on the performance of this specific component.

Given our reliance on user reviews, we recognized the shortage of written user feedback on various platforms. To address this gap, we proposed the development of a text-to-speech system [7]. Automatic Speech Recognition system would convert spoken content into text, and compare this transcribed text with the recommendations database, enriching the information available to users.

Considering these considerations, our research team embarked on the development of a chatbot recommendation system for computer accessories and repair centers as our research focus. The localized ASR component, tailored to South Asian English accents, stands as a critical innovation, amplifying the significance of enabling speech-to-text capabilities for our users. This technology promises to substantially enhance user interactions and foster a more inclusive user experience, making it a cornerstone of our research efforts.

1.2 Research Gap

According to the findings of the literature reviews a few aspects that were unconcerned in past research initiatives have been highlighted. Table 1 clearly shows the comparison between each previously completed research project and the proposed solution.

The Research [8] In this paper, the authors address the challenge of improving Automatic Speech Recognition (ASR) accuracy using deep learning techniques. While Convolutional Neural Networks (CNNs) have shown promise in ASR, they often have a limited number of layers, which may not capture the full complexity of human speech signals. To address this limitation, the authors propose a novel architecture called RCNN-CTC, which combines deep and wide CNNs with residual connections and the Connectionist Temporal Classification (CTC) loss function.

RCNN-CTC is designed as an end-to-end system capable of simultaneously exploiting both temporal and spectral features of speech signals. Additionally, the authors introduce a CTC-based system combination approach, which differs from conventional frame-wise senone-based methods. The subsystems used in this combination are diverse and complementary to each other.

Experimental results demonstrate the effectiveness of their approach. The proposed RCNN-CTC system achieves the lowest word error rate (WER) on two datasets: WSJ and Tencent Chat, outperforming several widely used neural network systems in ASR.

The Research [9] Convolutional Neural Networks (CNNs) for speech recognition which tries to prove that CNN which is more excellent in performance in image processing can also be used for speech recognition tasks due to local capture patterns and ability to train large amount of data.

The Research [10] Deep Learning-based Semantic Personalized Recommendation System, deep learning techniques can be used to improve the performance of this system consists of two main components:

1. A feature extractor, which is responsible for extracting useful features from the input data,
2. A recommender, which uses the extracted features to generate personalized recommendations The feature extractor uses a Deep Learning model based on Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to extract semantic features from the input data, which includes textual, visual, and contextual information.

Application Reference	Usage of Automatic Speech Recognition for computer & computer accessories domain	Usage of Sequence-to-Sequence Approach for ASR
Research [8]	X	X
Research [9]	X	X
Research [10]	X	X
Proposed System	✓	✓

Table 1 - Existing system and feature comparison

The proposed system has been meticulously crafted with the Sri Lankan community in mind, aiming to address a significant gap in the current e-commerce landscape. When users visit e-commerce platforms offering a variety of laptops and devices, they typically encounter filtering and product sorting options, which, while useful, often fall short of fulfilling their specific requirements. Notably, these platforms lack a robust recommendation system. Moreover, speech-to-text functionality is conspicuously absent, necessitating users to possess IT-related knowledge or conduct exhaustive manual research across multiple review websites to gather insights about products.

To elevate user-friendliness, The Research incorporates cutting-edge Automatic Speech Recognition (ASR) technology, ushering in a transformative approach to product searches. Acknowledging the user preference for avoiding text-based input, ASR introduces a seamless and convenient alternative. With ASR, users can articulate their product preferences vocally, thereby enhancing the overall user experience significantly. But our innovation doesn't stop there. ASR serves a dual purpose within the system. Beyond simplifying product searches, it also plays a pivotal role in converting speech to

text. This feature allows us to transform YouTube video reviews into text, thereby enriching our recommendation engine with a wealth of user-generated content. By leveraging ASR in this manner, we can provide users with an even more comprehensive and informed set of product recommendations.

Furthermore, the developed system incorporates an image identification component, further enhancing the user experience. Users can now visually search for products, eliminating the need for extensive textual queries. The combination of ASR for voice-based queries and image identification for visual searches positions our system as a holistic solution tailored to meet the diverse needs of users, simplifying the product discovery process, and offering well-rounded recommendations.

1.3 Research Problem

This research delves into real-world user practices, where individuals commonly engage in extensive product research, consideration of reviews before making purchasing decisions. In today's digital landscape, users frequently turn to popular search engines like Google or Bing to formulate queries regarding specific products. Subsequently, they navigate through various online platforms, such as websites, video streaming services, and community forums, in an endeavor to meticulously analyze articles, comments, and discussions associated with the product under consideration.

However, this traditional approach presents substantial challenges. It places a significant time and effort burden on users, necessitating them to sift through copious amounts of information. Additionally, the conclusions drawn from this manual research process may not necessarily align with their unique requirements. This research problem revolves around the development of a chatbot system designed to prioritize user queries and deliver personalized recommendations pertaining to computers, accessories, and repair centers. The primary objective is to streamline the decision-making process and efficiently cater to users' desires and needs.

At the core of this system's design lies its capacity to interpret and comprehend user queries. Through an interactive phase, users articulate their fundamental requirements, enabling the system to generate increasingly personalized outputs. These personalized recommendations consider various criteria, including price, durability, and more, culminating in tailored suggestions that align with individual needs. The system functions as a hybrid, seamlessly connecting recommendations concerning computers, accessories, and repair centers through a chatbot interface. This integration enhances user interaction and elevates the overall user-friendliness of the system.

A central question driving this research problem is: How can comprehensive product and repair center reviews be extracted across all forms of media? This inquiry encompasses diverse media types, encompassing both text and video reviews. Notably, the research primarily concentrates on the extraction of video reviews associated with specific products. These reviews contain spoken content, which the research intends to retrieve and subsequently compare with their written counterparts. The resultant information is systematically stored in a database, laying the foundation for subsequent analysis and the generation of product recommendations.

Within the context of this research problem, the crucial component of Automatic Speech Recognition (ASR) assumes a pivotal role. ASR technology empowers the system to

convert speech from video reviews into textual format, enriching the database with valuable content derived from spoken feedback. This innovative application of ASR significantly contributes to the research's overarching objective – offering users a seamless, comprehensive, and user-friendly approach to product and repair center recommendations across various media types.

2. OBJECTIVES

2.1 Main Objectives

At the heart of this research endeavor lies the principal objective: the development of an innovative Automatic Speech Recognition (ASR) system. This ASR system is offering a dynamic and user-centric approach. The integration of ASR into the chatbot's query input field, featuring an intuitive microphone icon, will empower users to communicate with the chatbot using their natural speech. This progressive shift from conventional text-based interactions to voice-enabled communication seeks to enhance user engagement, accessibility, and overall satisfaction.

2.2 Specific Objectives

These mentioned specific objectives must be reached to achieve the overall objective.

2.2.1 Identification of websites to be Scraped.

- For the research study without using a prebuilt dataset regarding devices and service centers we are developing the dataset with the help of Web Scraping

technique. Scraping the needed data and information from the text and video reviews which was already been published in various sources like online forums, online video streaming platforms, google reviews, retails websites and from product manufacturing sites, first thing is to clearly identify those websites, Since we are focusing on Sri Lankan going to scrape data from local websites, and not going to use any API to access the global websites like Amazon etc.

2.2.2 Custom Dataset Creation

- Complementing the central objective is the imperative task of constructing a bespoke dataset, meticulously curated from local computer and laptop device websites. This dataset will be used on chatbot's product recommendation. It encompasses a wealth of data, including device names, price points, specifications, product links, availability status, and user-generated reviews were pertinent.

2.2.3 Preprocessing of scraped data

- Once we have scraped the data, we need to clean and process it to ensure that it is accurate and consistent. This could involve removing duplicates, correcting errors, and formatting the data.

2.2.3 Google Maps Location Scraper

- development of a sophisticated Google Maps location scraper, a critical component enabling the chatbot to provide users with recommendations for repair centers. This scraper, through meticulous data retrieval, assembles a comprehensive database of location details. This includes center names, precise addresses, user-generated reviews, review counts, contact telephone numbers, service categories, and relevant web links. By procuring and organizing these invaluable details, the chatbot can seamlessly direct users to top-rated repair

centers located within a 10-kilometer radius of their specified location, significantly streamlining their decision-making process.

2.2.4 Automated Review Summaries

- automation of the review summarization process. As users interact with the system and articulate their requirements, the recommendation system will promptly offer a recommended list of devices that satisfy the user needs. From that list users have the freedom to request in-depth information about specific devices. In response, the system retrieves summary of a review sourced from YouTube. This summary, accompanied by the URL of the corresponding video, helps the users with rich insights into device specifics. This augmentation of the user experience enhances their ability to make informed decisions and reinforces the chatbot's role as a valuable informational resource. [12]

2.2.5 Identification of language and domain on speech data

- models may incorporate domain-specific vocabulary and language models to improve the accuracy of the speech-to-text system. For instance, a speech-to-text system designed for computer accessories recommendation may use computer details related vocabularies and dictionaries.

2.2.6 Collection of Speech data for model training

- Since we develop a speech to text system using a model, we must train that model to get more accurate results so that must collect diverse speech data where to build and robust speech recognition system. The “Common Accent” dataset will be used to train the model.

2.2.7 Evaluation Matrices to evaluate the system.

- To evaluate the performance of the system we will be using an evaluation matrices commonly used metrics to evaluate the performance of speech-to-text systems are Word Error Rate (WER) [13], Character Error Rate (CER), Word Accuracy
 - Word Error Rate (WER) measures the percentage of incorrectly recognized words in the transcribed text. (Lower WER rate higher accuracy)
 - Character Error Rate (CER) measures the percentage of incorrectly recognized characters in the transcribed text. (Lower CER rate higher accuracy)
 - Word accuracy can be gained using the WER (Higher Word Accuracy meant to be a good text word prediction)

Collectively, these research objectives converge to underscore the central mission of this study: to enhance user experiences through the pioneering integration of ASR technology, the strategic acquisition of data via web scraping, and the automated synthesis of review summaries. In focusing on these objectives, this research seeks to contribute to the dynamic landscape of user-friendly technological solutions while embracing the unique context of Sri Lankan users.

3. METHODOLOGY

For this research endeavor, two distinct methodologies were meticulously explored, each offering a unique perspective on the development of an Automatic Speech Recognition (ASR) system.

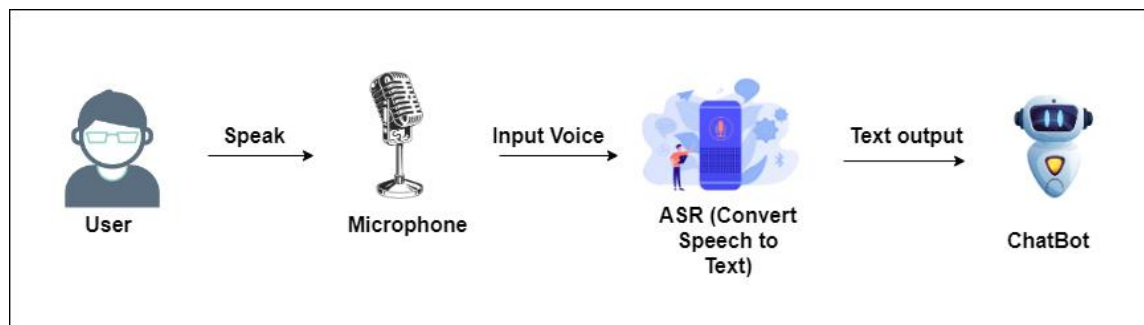


Figure 1 - Usage of ASR

3.1 CTC Architecture Approach

The initial approach delved into the utilization of Connectionist Temporal Classification (CTC) architectures [14], a prevalent choice for ASR model development up to the year 2022. In this framework, the foundation was laid upon the renowned Deep Speech 2 model, which had notably employed the LJ Speech dataset.

The LJ Speech dataset is a public domain resource comprising 13,100 succinct audio clips. These clips feature a solitary speaker reciting passages extracted from seven non-fiction books. Accompanying each audio clip is a transcription. The dataset's audio snippets exhibit varying durations, ranging from one to ten seconds, with a cumulative length of approximately 24 hours. The texts themselves, originating from works published between 1884 and 1964, are situated in the public domain. The audio recordings, captured during the years 2016-17, have been generously provided by the LibriVox project and reside in the public domain.

The prevailing approach in ASR model development, prior to year 2022, periodically based around the adoption of CTC architectures. Many research papers during this period leaned on models characterized by connectionist temporal classification, such as Wav2Vec2 and Hubert. These models primarily employ an encoder-only design,

followed by a linear classification head using the CTC framework. However, a noteworthy limitation of this approach is its propensity for spelling errors in the predicted output. These spelling inaccuracies were encountered in the initial model iteration, prompting a reconsideration of the methodology.

CTC training typically relies on entirely unlabeled data and adopts an encoder-only design, with a linear classification CTC head atop it. This architecture encompasses a fusion of both RNN (Recurrent Neural Network) and CNN (Convolutional Neural Network) components, resulting in a hybrid system. Nevertheless, it necessitates substantial computational resources for training and evaluation. An attempt to enhance model accuracy by running approximately 20 epochs on the LJ Speech dataset yielded suboptimal results.

```

1 def build_model(input_dim, output_dim, rnn_layers=5, rnn_units=128):
2
3     # Model's input
4     input_spectrogram = layers.Input((None, input_dim), name="input")
5     # Expand the dimension to use 2D CNN.
6     x = layers.Reshape((-1, input_dim, 1), name="expand_dim")(input_spectrogram)
7     # Convolution layer 1
8     x = layers.Conv2D(
9         filters=32,
10        kernel_size=[11, 41],
11        strides=[2, 2],
12        padding="same",
13        use_bias=False,
14        name="conv_1",
15    )(x)
16    x = layers.BatchNormalization(name="conv_1_bn")(x)
17    x = layers.ReLU(name="conv_1_relu")(x)
18    # Convolution layer 2
19    x = layers.Conv2D(
20        filters=32,
21        kernel_size=[11, 21],
22        strides=[1, 2],
23        padding="same",
24        use_bias=False,
25        name="conv_2",
26    )(x)
27    x = layers.BatchNormalization(name="conv_2_bn")(x)
28    x = layers.ReLU(name="conv_2_relu")(x)
29    # Reshape the resulted volume to feed the RNNs layers
30    x = layers.Reshape((-1, x.shape[-2] * x.shape[-1]))(x)
31    # RNN layers
32    for i in range(1, rnn_layers + 1):
33        recurrent = layers.GRU(
34            units=rnn_units,
35            activation="tanh",
36            recurrent_activation="sigmoid",
37            use_bias=True,
38            return_sequences=True,
39            reset_after=True,
40            name=f"gru_{i}",
41        )
42        x = layers.Bidirectional(
43            recurrent, name=f"bidirectional_{i}", merge_mode="concat"
44        )(x)
45        if i < rnn_layers:
46            x = layers.Dropout(rate=0.5)(x)
47    # Dense layer
48    x = layers.Dense(units=rnn_units * 2, name="dense_1")(x)
49    x = layers.ReLU(name="dense_1_relu")(x)
50    x = layers.Dropout(rate=0.5)(x)
51    # Classification layer
52    output = layers.Dense(units=output_dim + 1, activation="softmax")(x)
53    # Model
54    model = keras.Model(input_spectrogram, output, name="DeepSpeech_2")
55    # Optimizer
56    opt = keras.optimizers.Adam(learning_rate=1e-4)
57    # Compile the model and return
58    model.compile(optimizer=opt, loss=CTCLoss)
59    return model
60
61
62 # Get the model
63 model = build_model(
64     input_dim=fft_length // 2 + 1,
65     output_dim=char_to_num.vocabulary_size(),
66     rnn_units=512,
67 )

```

Figure 2 - Model Architecture Code

Consequently, this prompted a transition to the second approach, seeking a more effective avenue for ASR system development.

3.2 Sequence to Sequence Approach.

In the pursuit of refining the Automatic Speech Recognition (ASR) system, a second approach was meticulously explored, focusing on the Sequence to Sequence (Seq2Seq) methodology [15].

3.2.1 Dataset utilized.

The primary objective of this approach was to tailor the ASR system to recognize the nuances of the Sri Lankan accent. To achieve this, a dataset known as "Common Accent" was employed, boasting a substantial 100,000 training samples and 450 test samples. This dataset was meticulously curated to encompass voices featuring South Asian English accents, precisely aligning with the intended purpose.

Load Dataset

```
1 from datasets import load_dataset, DatasetDict
2
3 common_accent = DatasetDict()
4
5 common_accent["train"] = load_dataset("DTU54DL/common-accent", split="train")
6 common_accent["test"] = load_dataset("DTU54DL/common-accent", split="test")
7
8 print(common_accent)
```



Downloaded readme: 100% 3.42k/3.42k [00:00<00:00, 140kB/s]

Downloaded data files: 100% 2/2 [00:48<00:00, 20.61s/it]

Downloaded data: 100% 418M/418M [00:46<00:00, 11.2MB/s]

Downloaded data: 100% 19.3M/19.3M [00:02<00:00, 7.05MB/s]

Extracted data files: 100% 2/2 [00:00<00:00, 70.25it/s]

Generating train split: 100% 10000/10000 [00:02<00:00, 4073.67 examples/s]

Generating test split: 100% 451/451 [00:00<00:00, 3290.65 examples/s]

```
DatasetDict({
  train: Dataset({
    features: ['audio', 'sentence', 'accent'],
    num_rows: 10000
  })
  test: Dataset({
    features: ['audio', 'sentence', 'accent'],
    num_rows: 451
  })
})
```

Figure 3 - Loading Dataset

3.2.2 Data Preprocessing

Several crucial steps were undertaken to prepare the data effectively. The audio samples within the "Common Accent" dataset adhered to a common sampling rate of 16,000Hz. To optimize resource utilization and mitigate computational costs, the decision was

made to down sample the audio to 8,000Hz, a rate still well above the human speech frequency range.

Resampling

```
[ ] 1 from datasets import Audio
    2
    3 sampling_rate = processor.feature_extractor.sampling_rate
    4 common_accent = common_accent.cast_column("audio", Audio(sampling_rate=sampling_rate))
```

function to prepare our data ready for the model

Use the feature extractor to compute the log-mel spectrogram input features from our 1-dimensional audio array

Encode the transcriptions to label ids through the use of the tokenizer.

```
1 def prepare_dataset(example):
2     # load and resample audio data from 48 to 16kHz
3     audio = example["audio"]
4
5     example = processor({
6         audio=audio["array"],
7         sampling_rate=audio["sampling_rate"],
8         text=example["sentence"],
9     })
10
11     # compute input length of audio sample in seconds
12     example["input_length"] = len(audio["array"]) / audio["sampling_rate"]
13
14     return example
```

Apply the data preparation function to all of our training examples using huggingface Datasets' .map() method

```
[ ] 1 new_dataset = new_dataset.map(
    2     prepare_dataset, remove_columns=new_dataset.column_names["train"], num_proc=1
    3 )
```

Map: 100% ██████████ 7000/7000 [13:17<00:00, 2.01s/ examples]

Map: 100% ██████████ 3000/3000 [05:24<00:00, 3.85s/ examples]

Figure 4 - Resampling the loaded dataset code.

To extract meaningful acoustic features for subsequent analysis, spectrograms of the audio files were generated. Specifically, Mel-spectrograms were used, as they are particularly adept at capturing essential characteristics of the audio signal, aligning with human auditory perception.

To facilitate model training, audio chunks of 30 seconds duration were selected, and any datasets exceeding this length were truncated, while those falling short were padded with zeros at the end to meet the requisite duration.

Filter dataset audios into max length of 30 seconds

```
[ ] 1 max_input_length = 30.0
    2
    3 def is_audio_in_length_range(length):
    4     return length < max_input_length

[ ] 1 new_dataset["train"] = new_dataset["train"].filter(
    2     is_audio_in_length_range,
    3     input_columns=["input_length"],
    4 )

Filter: 100% ██████████ 7000/7000 [00:00<00:00, 7348.05 examples/s]

[ ] 1 new_dataset["train"]

Dataset({
  features: ['input_features', 'labels', 'input_length'],
  num_rows: 7000
})
```

Figure 5 - Chunking audio clips to 30 seconds Code.

Hugging Face's Whisper feature extractor class used transforming the raw audio data into input features tailored for the ASR model. Instances with durations less than 30 seconds were appropriately padded, while those exceeding the threshold were truncated.

Define a Data Collator

Takes out processed data and prepares pytorch tensors ready for the model

```
1 import torch
2
3 from dataclasses import dataclass
4 from typing import Any, Dict, List, Union
5
6
7 @dataclass
8 class DataCollatorSpeechSeq2SeqWithPadding:
9     processor: Any
10
11     def __call__(
12         self, features: List[Dict[str, Union[List[int], torch.Tensor]]]
13     ) -> Dict[str, torch.Tensor]:
14         # split inputs and labels since they have to be of different lengths and need different padding methods
15         # first treat the audio inputs by simply returning torch tensors
16         input_features = [
17             {"input_features": feature["input_features"][0]} for feature in features
18         ]
19         batch = self.processor.feature_extractor.pad(input_features, return_tensors="pt")
20
21         # get the tokenized label sequences
22         label_features = [{"input_ids": feature["labels"]} for feature in features]
23         # pad the labels to max length
24         labels_batch = self.processor.tokenizer.pad(label_features, return_tensors="pt")
25
26         # replace padding with -100 to ignore loss correctly
27         labels = labels_batch["input_ids"].masked_fill(
28             labels_batch.attention_mask.ne(1), -100
29         )
30
31         # if bos token is appended in previous tokenization step,
32         # cut bos token here as it's append later anyways
33         if (labels[:, 0] == self.processor.tokenizer.bos_token_id).all().cpu().item():
34             labels = labels[:, 1:]
35
36         batch["labels"] = labels
37
38         return batch
```

Figure 6 - Define Data collector code.

3.2.3 Model Architecture

The ASR model harnessed the power of the Sequence to Sequence (Seq2Seq) framework, incorporating both encoder and decoder components interconnected through a cross-attention mechanism. The encoder's primary function was to compute hidden-state representations of the audio inputs, effectively capturing their salient features. Meanwhile, the decoder assumed the role of a language model, with the capacity to transcribe the audio into textual representations.

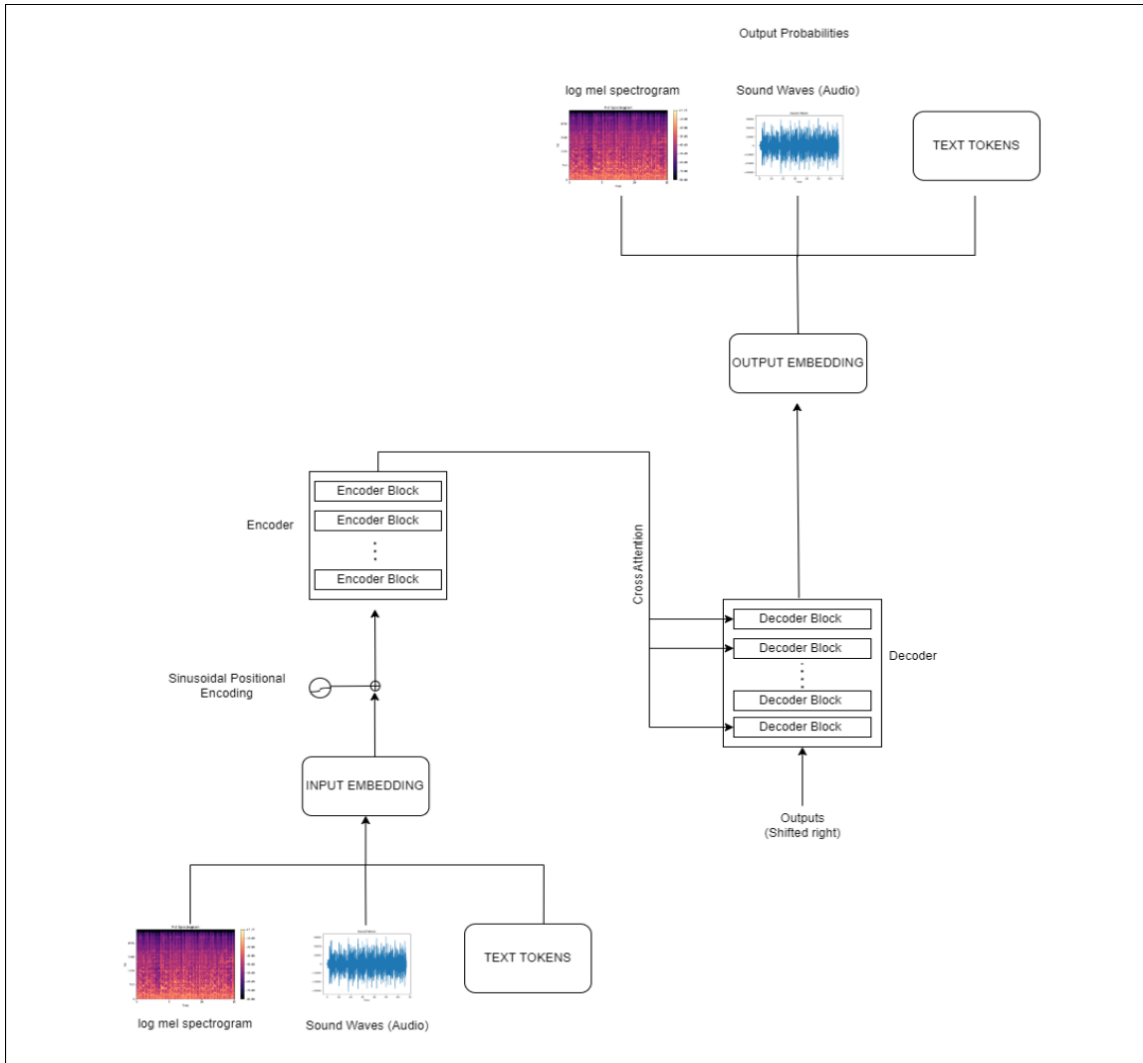


Figure 7- Sequence to Sequence Architecture

Define the Training Configuration

```
[ ] 1 from transformers import Seq2SeqTrainingArguments
    2
    3 training_args = Seq2SeqTrainingArguments(
    4     output_dir="./Wishwa98/ASRForCommonVoice",
    5     per_device_train_batch_size=16,
    6     gradient_accumulation_steps=1,
    7     learning_rate=1e-5,
    8     lr_scheduler_type="constant_with_warmup",
    9     warmup_steps=50,
   10     max_steps=2500,
   11     gradient_checkpointing=True,
   12     fp16=True,
   13     fp16_full_eval=True,
   14     evaluation_strategy="steps",
   15     per_device_eval_batch_size=16,
   16     predict_with_generate=True,
   17     generation_max_length=225,
   18     save_steps=500,
   19     eval_steps=500,
   20     logging_steps=25,
   21     report_to=["tensorboard"],
   22     load_best_model_at_end=True,
   23     metric_for_best_model="wer",
   24     greater_is_better=False,
   25     push_to_hub=True,
   26 )
```

Forward the training arguments to the hugging face Trainer

```
[ ] 1 from transformers import Seq2SeqTrainer
    2
    3 trainer = Seq2SeqTrainer(
    4     args=training_args,
    5     model=model,
    6     train_dataset=new_dataset["train"],
    7     eval_dataset=new_dataset["test"],
    8     data_collator=data_collator,
    9     compute_metrics=compute_metrics,
   10     tokenizer=processor,
   11 )
```

Figure 8 - Model Training Code

3.2.4 Challenges and solutions.

It's worth noting that Seq2Seq models inherently entail a slower decoding process, as decoding steps occur sequentially. Additionally, they exhibit a higher appetite for training data to achieve convergence. To address these challenges, the model underwent a fine-tuning process, leveraging a dataset meticulously curated for perfection.

In particular, the Whisper model served as the foundation for this ASR system, with room for further fine-tuning to accommodate translation tasks in addition to transcription.

In essence, this approach leveraged the power of Seq2Seq models and advanced acoustic feature extraction techniques to refine the ASR system's capacity to recognize and transcribe the distinct nuances of the Sri Lankan accent, striving for optimal accuracy and performance.

In addition to the core methodologies mentioned above, this research harnesses the power of web scraping to further enrich the ASR system's capabilities.

3.3 System Architecture

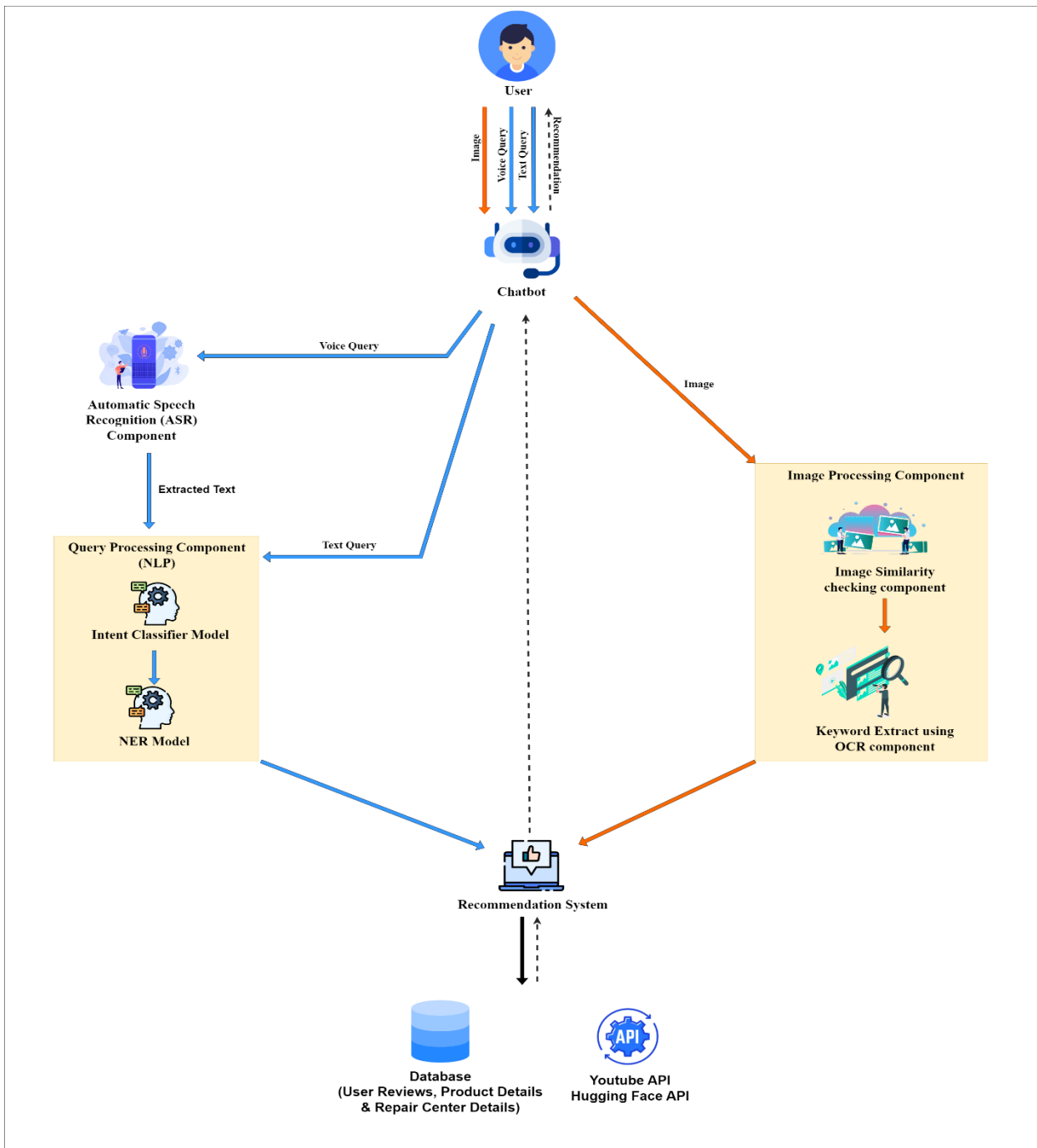


Figure 9 - System Overall Architecture

The high-level diagram Figure 9 of the proposed system provides an overview of its intricate workings, illustrating the seamless interaction between the user and the various components.

3.3.1 User Interaction

At the heart of the system lies the user, seeking information about specific products or service centers. The user's query, representing their fundamental requirement, is the catalyst for this journey. Users have the flexibility to input their queries or messages through the chatbot interface in three distinct formats: text, voice, or image, denoted by blue-colored and orange-colored arrows.

3.3.2 Chatbot Integration

The chatbot serves as the central hub, adeptly capturing and processing the user's input. If the user opts for a textual query, the chatbot seamlessly records it. Alternatively, if an image is uploaded, the image processing component takes charge. For voice queries, the user's spoken words are converted into text via the Automatic Speech Recognition (ASR) component and for the text queries it will direct to the Query processing component. In the end, regardless of the input format, all queries are unified as text.

3.3.3 Natural Language Processing (NLP)

With the gathered text-based input, the chatbot employs sophisticated Natural Language Processing (NLP) algorithms mainly intent classifier model and Name entity recognition is used to decipher the user's intent. This entails breaking down the received text into meaningful components, identifying critical keywords and phrases that encapsulate the user's request.

3.3.4 Recommendation System

Once the user's intent is discerned, the chatbot proceeds to match it with a predefined list of responses and actions. To cater to the user's needs, the chatbot communicates with the recommendation system, which is equipped with a repertoire of algorithms, including content-based filtering and collaborative filtering. These algorithms are instrumental in generating personalized recommendations tailored to the user's specific requirements.

3.3.5 User Responses

The culmination of this intricate process results in responses generated by the recommendation system. These responses are then relayed back to the user through the chatbot interface, as indicated by dotted arrows. The user receives these responses as answers to their queries or messages, ensuring a seamless and informative interaction.

3.3.6 Image Recognition

In addition to product recommendations, the system boasts image recognition capabilities. When a user uploads an image of a computer accessory to the chatbot interface, represented by an orange-colored arrow, the chatbot springs into action. Leveraging Image similarity checking component where it check the similarity scores between the inputted image and the existing image dataset and that will further directed to the Optical character recognition finally it identifies the accessory within the image.

3.3.7 Database Utilization

The system leverages a meticulously crafted database, meticulously constructed through web scraping efforts. This database serves as a valuable resource in responding to user queries with relevant information. When a user submits an image, the chatbot conducts a database search, returning a list of results matching the identified accessory.

3.3.8 YouTube Integration

To augment the recommendation process, the system taps into YouTube APIs to access user queries and retrieve video-related reviews. These reviews are summarized, and their respective YouTube links are provided to users for further exploration.

In essence, this comprehensive system seamlessly integrates user input, NLP, recommendation algorithms, image recognition, database utilization, and YouTube integration to provide users with a rich and informative experience, catering to their specific needs and inquiries.

3.4 Commercialization of the Product

Commercializing a Chat-Bot system for laptops, accessories, and service center recommendations can be a successful business venture if handled effectively. To do so there are certain factors to be considered.

The most initial and crucial factor is identifying the target market and potential buyers of the developed product. In that case, we propose this system towards computer selling companies. In such ways, they will be able to increase their sales & customer base as well.

In comparison to the competition seen in the available market, there are none too rare instances where we see Chat-Bot systems. Hence, commercializing such a product would be very useful. In addition, we don't see such system in the Sri Lanka market as well. So, implementing such a system would raise customer satisfaction and raise develop the competition in the Sri Lankan market as well.

As proposed, shown below is the product logo. We are targeting Sri Lankan e-commerce platforms, computer retail shops & repair centers to promote this product.

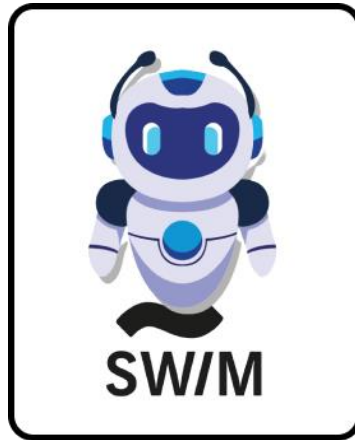


Figure 10 - Logo of proposed system

3.4 Software solution

The Software Development Life Cycle is a methodical approach to software development that ensures accuracy and consistency. When a need changes, developers cannot go back to the earlier step in the conventional process. As a result, the developers must adhere to every instruction precisely. But, if the agile methodology is used, developers will adapt to the change in requirements. As we employ the Agile strategy, the team members should adapt to the changes. Scrum is the agile framework that has the most sway in contrast to the others. Moreover, Scrum is a simple project management tool that can handle challenging problems. The Scrum approach, in general, breaks down the drawn-out waterfall process delivery into shorter cycles, enabling product teams and the end user to periodically evaluate functioning software and ensure that it fits their business needs. This ensures that the final product also meets the demands of the consumer. When needs change, SCRUM may be examined and modified. Therefore, the hypothesis that was tested in both the literature review and the actual survey is what served as the foundation for the authors' suggested remedy. Adjustments will be made on an ongoing basis in response to client demands and commercialization. [16]



Figure 11 - Agile methodology Structure

3.5 Requirement gathering and analysis.

The procedure of obtaining requirements will be the initial stage. As this phase is all about discovery, it simply entails understanding and identifying the technical requirements for the business project before moving on with a clear strategy. The information acquired at this stage will serve as the foundation for the System Requirements Specifications (SRS) document, therefore it is crucial.

3.5.1 Requirements gathering

The research aims to concentrate on and gain a better understanding of public opinion regarding the project, whether there are positive and negative marks on attempting to implement a chatbot recommendation system for computer accessories and repair centers. The initial business problem was identified by posing a questionnaire. There was discussion of the general flaws in the current manual reviews recommendation systems and consideration methods for computer accessories. Consumers of E-commerce websites and those with a technical bent around Sri Lanka were given the.

Google Form was used to administer the survey. The response rate of the public to the execution of each component was also included in the questionnaire. Further study involved interviews with several corporate users who discussed the shortcomings of the present systems. The input from the business users has significantly advanced the component's specified goals and project milestones. Please refer to APPENDIX A, APPENDIX B for the questionnaire.

3.5.2 Functional Requirements

1. Transcribing the Inputted Voice (Audio) to Text

The system must have the ability to recognize the inputted voice using the automatic speech recognition model and would be able to convert the inputted audio into transcribed text.

2. Filtering the accurate text

The system must have the ability to recognize and extract the most accurate text through a speech to text recognition from user generated video reviews.

3. Storing the extracted text

The system must have the ability to store the transcribe text into the database for future analysis where those extracted text can be used to analyze the vocal tones and patterns to determine the sentiment or emotions towards that device (sentimental analysis) and finally for the product recommendations.

3.5.3 Non-Functional Requirements

- Reliability

The functioning of the system should be highly stable and constant, and it shouldn't falter while offering recommendations to consumers. The final advice or the recommendation should be reliable.

- User-friendliness

The system will provide an easy environment for users to use the existing features like entering a query and having the recommendation results.

Since we are using Image processing with the chat bot the user interaction might get easier where it will directly impact the user friendliness.

- Accuracy

The system should have a high level of accuracy of predicting and recommending the products to the users since the system has promised the users to recommend the best product for their requirement. Systems accuracy can be increased due to the Seq2Seq approach used for the speech to text model development.

3.5.4 Feasibility study (Planning)

- **Economic feasibility**

The benefit and project development cost are covered in the economic feasibility study. When a good economic feasibility plan is in place, the procedure will be successful. Consequently, the suggested system should be effective and less costly.

- **Scheduled feasibility.**

In feasibility study, Since the project's intent will be defeated if the timeline is not followed, scheduled feasibility assesses the timeline (period) for the planned, proposed project. As a result, the tasks in the suggested solution ought to be finished in around the same amount of time.

- **Technical feasibility**

Technical feasibility is the assessment of the expertise, resources, and skills needed to create the proposed web application, as well as the knowledge of the system architecture and the communication abilities needed to comprehend the demands of the stakeholders to complete the suggested project solution.

3.6 Tools and Technologies

3.6.1 Tools

IDE: VS Code - Visual Studio Code provides an interactive workspace to develop deep learning models and contains crucial extensions like Visual Studio Code Tools for AI to quickly deal with machine learning-powered model.

Google Colab - The free cloud-based Jupiter notebook environment Google Colab enables the training of deep learning and machine learning models on CPUs, GPUs, and TPUs. The accessibility of free GPUs and TPUs is the main benefit of using Google Colab. Training models, especially deep learning ones, take several hours on a CPU, GPUs and TPUs on the local machines, on the other hand, can train these models in a matter of minutes or seconds.

GitLab - GitLab is a web-based version control system that allows developers to collaborate on code and track changes over time while several people work on the same project at once. It may be challenging to make sure that everyone is using the most recent version of the code and to keep track of developer changes. This is why tools for version control, like GitLab, are useful. To facilitate team collaboration and task

management, GitLab also offers capabilities like code reviews, issue tracking, and project management tools.

TensorFlow - A tool and technology for creating and implementing machine learning models is called TensorFlow. TensorFlow is a Google-developed open-source toolkit that gives programmers the ability to create and train a variety of machine learning models, including neural networks, deep learning models, and other statistical models.

3.6.2 Technologies

Python - Python is easy to learn and offers rapid model development speed. Python takes less code than some other programming languages, therefore we will be able to develop prototypes and test our ideas more quickly and simply with Python. Python offers several excellent libraries for handling audio. One of the most well-known and full of features is Librosa. Several speech recognition engines are supported by Speech Recognition, including Wit.ai, CMU Sphinx, and Google Speech Recognition.

Systems for automatic speech recognition (ASR) - These algorithms are capable of transcribing speech in real time. To translate voice into text

3.7 Implementation

Web application development

The finished solution includes a web application to real-time every output detail. Authors of the program should be well-versed in Visual Studio code and web application development.

Front End Development

For front-end web development React is used. Preact is a lightweight alternative to React, and it is well-suited for front-end development. Preact has a small footprint where

it is a lightweight library, weighing in at only 3kb minified and gzipped. Because of Preact's highly optimized virtual DOM implementation, user interfaces are quick and responsive. Preact's minimal footprint helps it run more quickly since there is less code to download and parse. Since React and Preact share a similar API, getting started with Preact shouldn't be a problem. This implies that when working with Preact, we may make use of our current expertise and abilities.

Back End Development

Python's Flask web framework is nimble and adaptable, enabling programmers to create web applications rapidly. Flask is a popular option for creating web applications of all sizes because of its simplicity, versatility, and usability. Python flask can be effectively used to develop the backend more easily since it's easy to start where getting started with flask is straightforward and intuitive because of its simple and intuitive API. The boilerplate code and settings needed for Flask are minimal. Without being constrained by superfluous functionality, this enables us as developers to create unique apps that cater to our requirements. Flask's modular architecture makes it simple to add or remove features as necessary. As a result, developing and maintaining complicated applications becomes simple.

Database handling

The voice input from the video reviewers and the text outputs from the written reviews will be stored in MongoDB, a NoSQL database that is free, open-source, highly scalable, and document-oriented. To manage the database, it will be necessary to be familiar with adding, updating, removing, securing, and filtering data with Mongoose. The ideal option to store these kinds of unstructured data is to utilize MongoDB because the system will generate millions of data for the server.

3.8 Deployment

Cloud Platform – Google Cloud Platform (GCP)

Since this is a chatbot system and more user interactions can occur so thinking about the scalability is more important, depending on demand, GCP enables us to scale your application resources up or down as necessary. This enables your application to manage traffic peaks and always provide a positive user experience. For the bulk of its services,

GCP offers an infrastructure with a 99.95% SLA for uptime. This ensures that our users can access our application at all times. Since we are considering the cost-effectiveness pay as you go pricing models would be important to us to deploy this web application.

4 WORK BREAKDOWN STRUCTURE AND TIMELINE

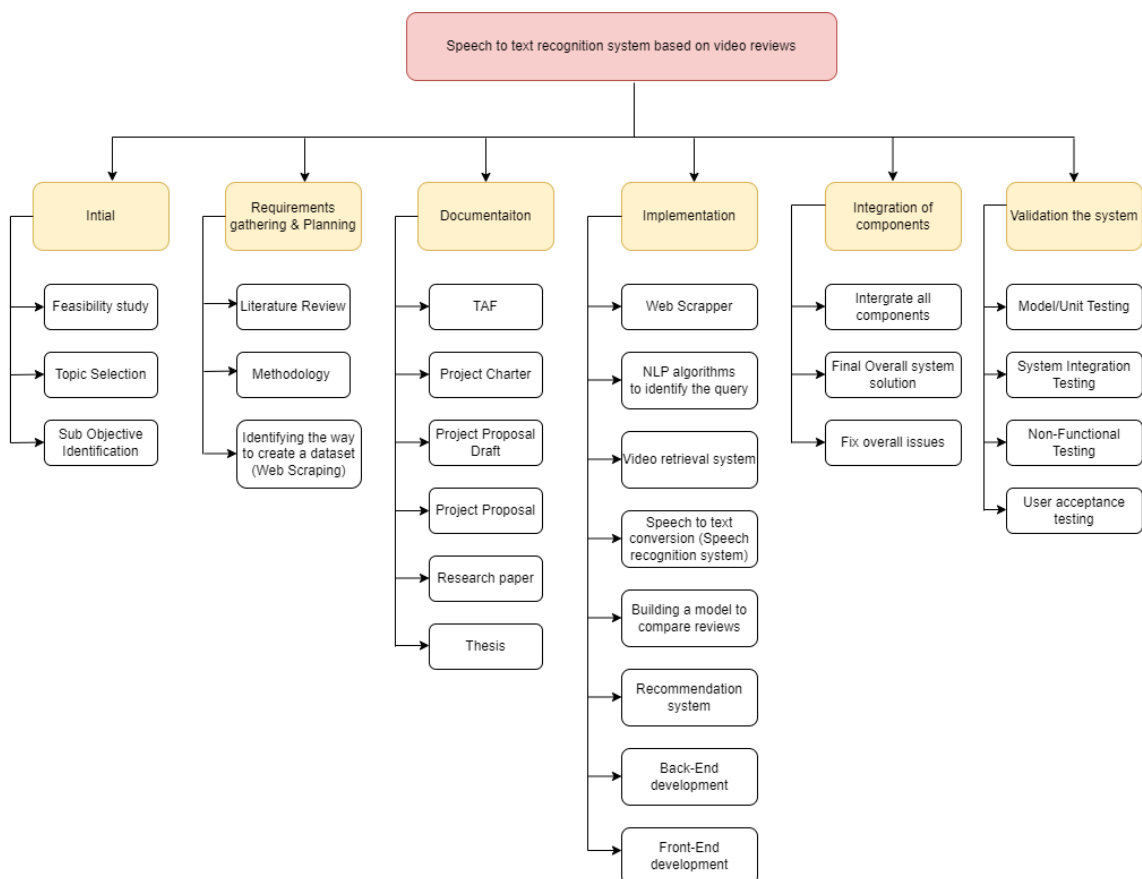


Figure 12 - Work Breakdown Structure

The project management tool of choice is a WBS (Work Breakdown Structure), which adopts a step-by-step methodology to finish the project and approaches the sub-goals and primary objectives with several moving parts. This tool aids in project breakdown

into manageable components and provides a clear view of the project's scope and deliverables.

5 GANTT CHART

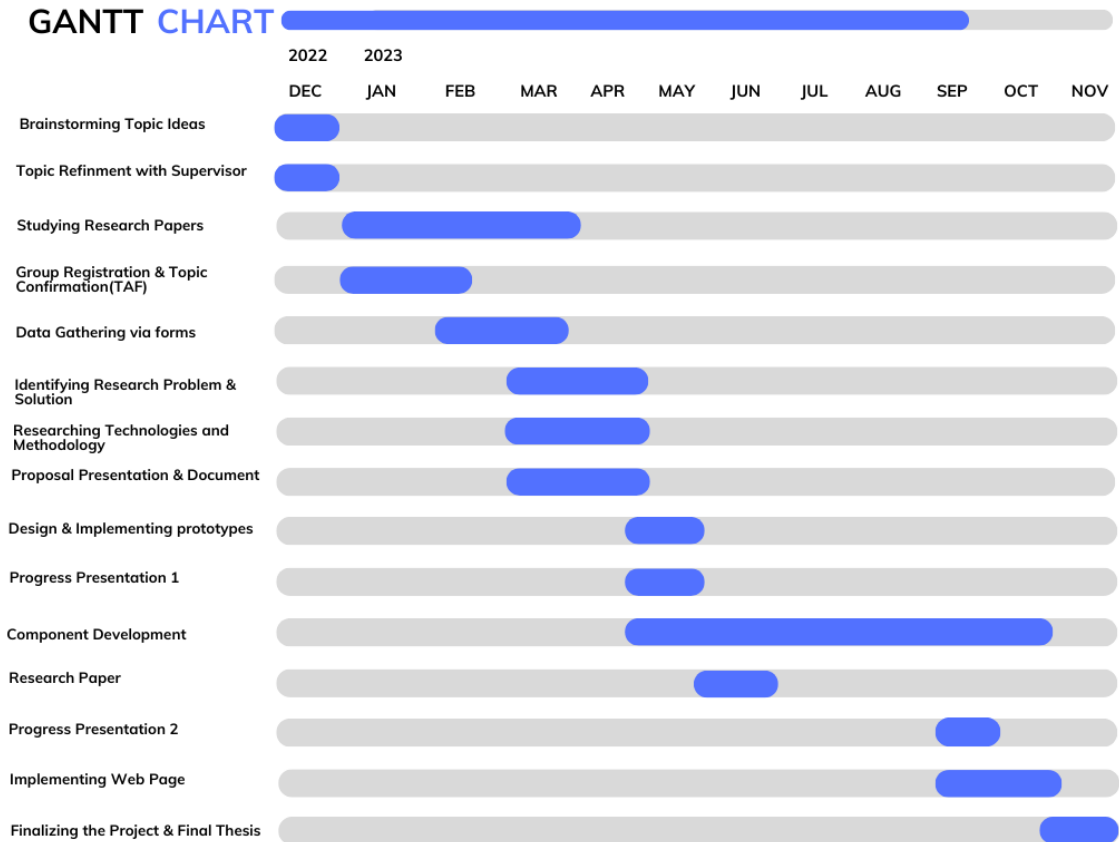


Figure 13 - Gantt Chart

6 RESULTS AND DISCUSSIONS

The ASR (Automatic Speech Recognition) component is a major element within the broader framework of this research. This section delves into the outcomes and insights gleaned from the deployment of the ASR system. Through meticulous evaluation and analysis, we uncover the impact of ASR on user interactions, user satisfaction, and the overall efficacy of the chatbot system. Moreover, we explore the ASR's ability to seamlessly bridge the gap between users and technology by enabling natural speech-based interactions. The subsequent discussion unravels the nuances, challenges, and potential enhancements that have emerged in the wake of ASR integration, shedding light on the transformative potential of this innovative technology.

6.1 Results

In our evaluation process the Word Error Rate (WER) used as the primary metric for assessing performance. WER offers a comprehensive analysis by categorizing errors into distinct types: Substitution, Insertion, and Deletion. It's important to note that our evaluation focused on the word level, where these errors were meticulously annotated on a per-word basis. The categories of errors in our assessment were represented as follows:

S: Substitution

D: Deletion

I: Insertion

N: Total number of words

$$WER = \frac{S + I + D}{N}$$

Figure 14 - Equation to Calculate the Word Error Rate

Word Error Rate is an evaluation matrix based on error counting mainly an error matrix without using an error matrix to evaluate our model that can use accuracy matrix named Word Accuracy by transforming the error matrix This transformation enables a straightforward assessment of our model's accuracy It provides valuable insights into the model's performance and its capacity to generate correct results. This shift in perspective offers a more user-friendly means to grasp the model's overall accuracy and effectiveness.

$$W_{Acc} = 1 - WER$$

Figure 11 - Word Accuracy Equation

As previously mentioned, two distinct approaches were employed, each yielding its own set of results:

1) CTC Approach:

When utilizing the CTC approach, had encountered the challenge of demanding computational resources. The attempt to train the model over approximately 20 epochs to enhance accuracy on the LJ speech dataset did not yield the desired results. The Word Error Rate (WER) stood at approximately 26%, which falls short of the desired performance. Notably, a noticeable gap emerged between the target text and the predicted text, contributing to the elevated WER.

2) Seq2Seq Approach:

In contrast, the Seq2Seq approach delivered more promising outcomes. The final model achieved a WER of 13.06, showcasing improved accuracy and performance in comparison to the CTC approach.

CTC – Model based on Deep Speech2	Seq2Seq Whisper finetuned model
<p>WER = 0.26 (26%)</p> <p>$W_{Acc} = 1 - 0.26 = 0.74$ (74 %)</p>	<p>WER = 0.13 (13%)</p> <p>$W_{Acc} = 1 - 0.13 = 0.87$ (87 %)</p>

Table 2 - Accuracy Comparison

Epoch	Step	Training Loss	Validation Loss	WER Ortho	WER
6	2500	0.006	0.423	17.92%	13.06%

Table 3 - Automatic Speech Recognition Model Results

6.2 Discussions

As the discussion unfolds, it becomes evident that developing ASR models with the Seq2Seq architecture represents the superior approach when compared to the CTC method. The Seq2Seq architecture's efficiency in resource utilization, reduced spelling errors, higher word accuracy, and robustness handling capabilities make it the clear choice for our research and integration into the chatbot system. This strategic decision aligns seamlessly with the overarching goal of enhancing user interactions, accessibility, and overall satisfaction within the chatbot system, solidifying the Seq2Seq architecture as the optimal path forward in ASR model development.

CTC Approach	Seq2Seq Approach
High usage of computational power	Comparatively low usage of computational power
High rate of spelling errors	Low rate of Spelling errors
Only use Encoder.	Use both decoder and encoder (Can enhance the ASR for multiple languages)
Word Accuracy is Low	Word Accuracy is High
Difficult to handle High robustness, Noise inputs.	High robustness, Noise inputs can be handled.

Table 4 - Approach Comparison

6.3 Future Work

In future endeavors, a significant focus will be directed towards enhancing the performance of our ASR model. Specifically, the aim is to construct a bespoke dataset encompassing Sri Lankan English-speaking accents tailored to the computer domain. This dataset will be meticulously designed and compiled to encapsulate the nuances and characteristics unique to the Sri Lankan English accent.

The primary objective of this future work is to fine-tune our ASR model using the newly created Sri Lankan English-speaking accent dataset. By doing so, expecting to achieve a substantially lower Word Error Rate (WER) compared to the results obtained from the previously employed dataset. This step represents a pivotal advancement in this research, as it enables the ASR system to be finely attuned to the specific linguistic attributes and idiosyncrasies of the Sri Lankan English accent within the computer domain. It not only paves the way for more accurate and effective voice recognition but also aligns the system more closely with the needs and preferences of the local user base. This initiative underlines the commitment to continually refining and optimizing

the ASR model to better serve the Sri Lankan community and, by extension, users with similar accents in the broader South Asian context.

7 REFERENCES

- [1] Nanotek, "Nanotek," [Online]. Available: <https://www.nanotek.lk/category/laptops>. [Accessed March 2023].
- [2] "lankayp," [Online]. Available: https://www.lankayp.com/category/Computer_repair/city:Colombo. [Accessed March 2023].
- [3] ceylonpages, "ceylonpages.lk," [Online]. Available: <https://ceylonpages.lk/category/computer-repair/>. [Accessed March 2023].
- [4] P. Baheti, "Image Recognition: Definition, Algorithms & Uses," V7 Labs, 2023. [Online]. Available: <https://www.v7labs.com/blog/image-recognition-guide>. [Accessed March 2023].
- [5] B. A. Shawar and E. Atwell, "Chatbots: Are they Really Useful?," *Journal for Language Technology and Computational Linguistics*, vol. 22, no. 5, pp. 29-49, 2007.
- [6] . P. K. Singh, P. K. D. Pramanik, A. Dey and P. Choudhury, "Recommender systems: An overview, research trends, and future directions," *International Journal of Business and Systems Research*, vol. 15, no. 6, pp. 14-52, 2021.
- [7] P. Vijay, "Voice recognition system: speech-to-text.," *Journal of Applied and Fundamental Sciences*, vol. 2, no. 7, p. 191, 2015.
- [8] Y. Wang, X. Deng, S. Pu and Z. Huang, "Residual Convolutional CTC Networks for Automatic Speech Recognition," *Research Gate*, vol. 1, no. 1, p. 10, 2017.
- [9] Abdel-Hamid and Ossama, "Convolutional neural networks for speech recognition," *EEE/ACM Transactions on audio, speech, and language processing*, vol. 22, no. 9, pp. 1533-1545, 2014.
- [10] S. Sharma, V. Rana and V. KumaR, "Deep learning based semantic personalized recommendation system," *International Journal of Information Management Data Insights*, vol. 1, no. 10, p. 100028, 2021.
- [11] R. Rosa, Gisele Maria Schwartz, W.V. Ruggiero and Demostenes Zegarra Rodriguez, "A Knowledge-Based Recommendation System That Includes Sentiment Analysis and Deep Learning," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 8, pp. 2124-2135, 2018.
- [12] Youtube, "Youtube," Youtube, 12 06 2023. [Online]. Available:

- <https://developers.google.com/youtube/v3/guides/implementation>. [Accessed 01 09 2023].
- [13] B. Worthy, "Medium," Medium, 02 December 2019. [Online]. Available: <https://medium.com/@bethworthy/what-is-word-error-rate-measuring-the-wer-of-machine-generated-transcripts-and-its-limitations-1457be914f3b>. [Accessed 01 08 2023].
- [14] S. Watanabe, T. Hori, S. Kim, J. R. Hershey and T. Hayashi, "Hybrid CTC/Attention Architecture for End-to-End Speech Recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240-1253, 2017.
- [15] A. Radford, J. W. Kim, T. Xu , G. Brockman, C. McLeavey and I. Sutskever, "Robust Speech Recognition via Large-Scale Weak Supervision," *PMLR*, vol. 202, no. 1, 2023.
- [16] . A. Srivastava, S. Bhardwa and S. Saraswat, "SCRUM model for agile methodology," *2017 International Conference on Computing, Communication and Automation (ICCCA)*, vol. 1, no. 11, pp. 864-869, 2017.

8 APPENDICES

Appendix A: Sample Questionnaire

<https://forms.gle/FNLZidzXDE5MZKbX8>