AUTOMATED TOURISM KNOWLEDGE GRAPH AND INTENT GENERATION FROM AUDIO CONTENT EXTRACTED FROM VIDEOS, BY UTILIZING NLP

Senuri Sarindi Seneviratne

199486X

Degree of Master of Science

Department of Computational Mathematics

University of Moratuwa
Sri Lanka

July 2022

AUTOMATED TOURISM KNOWLEDGE GRAPH AND INTENT GENERATION FROM AUDIO CONTENT EXTRACTED FROM VIDEOS, BY UTILIZING NLP

Senuri Sarindi Seneviratne

199486X

Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science

Department of Computational Mathematics

University of Moratuwa
Sri Lanka

July 2022

Declaration, copyright statement and the statement of the supervisor.

I declare that this is my own work and this thesis does not incorporate without acknowledgement 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 University of Moratuwa the non-exclusive right to reproduce and distribute my thesis, 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).

Signature: Date:

The above candidate has carried out research for the Master's thesis under my supervision.

Signature of the supervisor: Date:

Abstract

Generating a knowledge graph for a chatbot is a time-consuming exercise which needs the help of an expert relevant to the field. This thesis presents our approach to synthesizes the creation of a knowledge graph and intents for a chatbot. Currently, the creation of a knowledge graph and intents for a chatbot is a tedious process and this process does not extract data from videos. Developing a chatbot also requires the support of experienced software engineers.

This platform allows a user to build a customized chatbot according to a specific requirement in any field, without the intervention of experts. It also allows for the seamless development of a comprehensive knowledge graph from the video content through a simple and less tedious approach. The platform uses Natural Language Processing (NLP) machine learning models such as Naive Bayes and Logistic Regression and grammar correction techniques to supplement the experience of the users.

The working process of this proposed system is Knowledge Extraction and generating the Knowledge Base. The user inserts keywords related to the chatbot's domain as the first step of the process. The system retrieves the search results from YouTube. Finally, NLP will be used to retrieve data contained in videos to create a preliminary knowledge graph and intents for a chatbot. A scheduler is then activated automatically from time to time to update the knowledge graph and intents. The knowledge graph and intents generated have been tested on a chatbot created using the Rasa framework, with the chatbot giving the correct answers when questioned by a user.

Acknowledgment

First and foremost, I would like to extend my sincere gratitude to Dr. Sagara Sumathipala for his invaluable support, guidance and motivation during the implementation of the project and preparation of the thesis. The constructive criticism given to me during the multiple rounds of supervision has helped me to improve and refine my thesis according to the guidelines shared by the University of Moratuwa.

Next, I would like to thank Thilanka Kumarasinghe from my workplace for helping me understand the importance and application of a chatbot for different industries, which motivated me to select this topic for my Master’s thesis.

Finally, I would like to extend my gratitude to my parents and the sister for all the morale support given to me and for all sacrifices they made on my behalf to ensure that I succeed in this project. The guidance and motivation they gave me has enabled me to fuel my inner drive to do the absolute best in completing this thesis.

Table of contents

List of figures

List of tables

List of abbreviation

Abbreviation	Description
FN	False Negative
FP	False Positive
FPR	False Positive Rate
NLP	Natural Language Processing
NLU	Natural Language Understanding
OIE	Open Information Extraction
RE	Relation Extraction
TN	True Negative
TP	True Positive
TPR	True Positive Rate
URL	Uniform Resource Locator

Chapter 1

1.0 INTRODUCTION

1.1 Prolegomena

A chatbot is an Artificial Intelligence software which is built to develop a conversation. This conversation is built with the user through the Natural Language Processing (NLP) technology. A chatbot builds an advanced communication or interaction between the human and the machine on media such as websites, mobile applications and messaging applications. A chatbot to build an advanced communication a good knowledge base will be a great benefit.

A knowledge base is just like the brain behind any application which consists of very complex unstructured and structured data. It is a repository or a collection of data and information which is used to build the communication of a chatbot.

However, Chatbots knowledge base and intent generation currently have a few challenges to their development. The creation of a knowledge base for a chatbot is a tedious process and this process currently does not extract data from video contents. Developing a chatbot also requires the support of experienced developers and comes at a high cost. The research problem is the identification of an accurate method for extracting knowledge from video contents.

This platform allows a user to build a customized chatbot knowledge base and intents according to a specific requirement in a particular field, without the involvement of experienced human resources in that field. Since the solution does not involve experienced human resources, it comes at a lower cost. It also allows for the seamless development of a comprehensive knowledge graph from audio content obtained from videos, through a simple and less tedious approach. Proposed solution is capable of generating a comprehensive knowledge graph from video content.

1.2 Aims and objectives

The goal of the research is to develop an efficient database which is used to create an efficient chatbot communication.

In order to reach this aim, we have defined the following objectives

- Critically review existing approaches for knowledge graph generation for chatbots.
- In depth study on the technologies used to generate chatbots.
- Design and develop algorithms to generate a Knowledge Base
- Evaluate the development approach simulation and compare and contrast with the existing systems.

1.3 Background and motivation

A chatbot can be defined as a program developed to simulate a smart communication on a text [1]. Chatbots use embedded knowledge to identify a user's query and give an appropriate response [1]. Chatbots are developed for specific domains such as education, business, and healthcare [45]. Current developments in NLP and advancement communication technologies and computing power have facilitated fast development and implementation of chatbots in multiple industries [45].

In order to emulate perfectly on human dialogs, chatbots needs to identify the inputs provided by a specific user and to formalize it in a proper way to provide an appropriate response. Chatbots are also known as the intelligent software that are being engaged as actioned agents by using natural languages that helps to develop natural conversations with people. Chatbots are the source of answers that provides answers for the questions raised by the user in smart communication using text method by using its embedded knowledge. Speech is also being applied in many terms with the support of artificial intelligence as speech is a strong mode of communication. Based on the research findings it highlighted the fact that there is a need to develop a database which has the ability to store all the data and to develop a web interface [1].

1.4 Problem definition

Generating a knowledge graph for a chatbot might be a time-consuming exercise. Most of the current chatbots are being developed with the knowledge of developers or experts of a specific domain. The research problem is “the lack of a simple methodology for automatic generation of the knowledge graph and intents for a chatbot, using knowledge extracted from audio content obtained from videos for a specific domain”. Most of the current chatbots are being developed with the knowledge of the developers or experts on the specific domain. They are expensive and needs programming skills.

1.5 Your solution

This research focuses on generating a comprehensive knowledge graph and chatbot intents from audio content obtained from videos, through a simple and less tedious approach.

1.6 Resource requirements

A computer with a 8GB RAM ,240GHz or higher processing power, windows 64 bit operating system and 1TB hard drive with a good internet connection. Which has Neo4j graph database

1.7 Summary

This chapter presented a critical review of literature on chatbots, problem and the solution. The research which has been defined as chatbots suffers due to lack of knowledge base. The available chatbots are expensive as they need the intervention of developers. They are also not reusable therefore the effort of development is a sunk cost. The next chapter provides a detailed discussion on the development and the issues of chatbots.

Chapter 2

2.0 CHATBOT – DEVELOPMENTS AND ISSUES

2.1 Introduction

Chapter 1 provided an overall picture of the research presented in this thesis. This chapter provides a critical review of literature in chatbots and knowledge bases. the literature review has been structured under several subheadings, namely early developments, latest developments and the challenges in chatbots and knowledge base development. This chapter also summarizes the challenges in chatbots and defines the Research Problem The research problem is “the lack of a simple methodology for automatic generation of the knowledge graph and intents for a chatbot, using knowledge extracted from audio content obtained from videos for a specific domain”.

2.2 Early developments in Chatbots and knowledge bases

In 1950 Alan Turing the British mathematician and logistician was the first to conceptualize the idea of a chatbot. He proposed a question “Can a machine think” which begins from the meaning of word “machine” and “think” [10][25][6]. As the solution for the question “Can a machine think” Turing test was used to determine whether a computer has the ability to think like a human being [47]. During this Turing Test it is the interaction between three parties, where one human work as a questionnaire the other human and a machine behaves as respondents. The questioner cross-examines the respondents according to a subject area for a specific time period and decide which respondent was a machine and which respondent was a human being [12][34][6].

Due to the advancement of Natural Language Processing chatbots have evolved. In 1966 world’s first chatbot ELIZA was introduced which simulates the operations of a psychiatrist [42][20]. It was an inspiration to other developments of chatbots but its communication ability was limited. ELIZA uses pattern matching and template-based

response selection [45][14]. ELIZA takes text as it inputs and it generates responses by reassembly rules associated with some selected decomposition rules.[20] Since ELIZA’s knowledge is limited, it targets on particular domain, which is a major drawback. It doesn’t have the ability to continue long conversations [10][14].
Chatbot PARRY was introduced in 1972 which has the ability to act as a patient with schizophrenia. PARRY has an advanced control structure and it has the same response pattern structure as ELIZA [36]. It is considered that PARRY is more advanced than ELIZA since it has a better personality and a controlling structure [14]. It is a chatbot which has low capabilities, low response speed with the inability to learn from conversations [14]. PARRY has the capability of language understanding [36]. It consists of effective variables such as fear, mistrust and anger [36].

In 1983 the story telling chatbot Racter was developed by William Chamberlain and Thomas Etter. The book named “The Policeman’s beard a half constructed” was written by Racter and was published in 1983[12].

In 1998 Jabberwacky was developed by Rollo Carpenter and in 1197 it was made public [22]. It is the first chatbot which uses Artificial Intelligence [14]. It was written in CleverScript, it is a spreadsheet-based language used to develop chatbots [14]. Jabberwacky has drawbacks where it is unable to work with a large amount of users and unable to reply at high speed [14].

The term chatbot was first coined in 1991 with TINYMUD whose functionality was to chat. It was an artificial player [14].

Dr Sbatso was developed in 1992 by Creative Labs. It is a fully voice operated chatbot which is an AI speech synthesis program. It acts as a psychologist [43][18].

In 1995 ALICE (Artificial Linguistic Internet Computer Entity) was introduced which was the first online chatbot inspired by ELIZA [1][14][27]. It was created by Richard

Wallace [1]. ALICE didn’t have any actual perception of the conversation and based on pattern matching [14]. IT is an open-source natural language processing chatbot which was developed using Artificial Intelligence Markup Language [1][14]. According to the architecture of ALICE “language knowledge model and “chatbot engine” are two separated modules. Therefore, it has the ability to plug alternative language knowledge models [27]. ALICE depends on very large amounts of input patterns which are rule matching to output templates [5]. The major drawback in ALICE is there was no intelligent features and didn’t have the ability to express attitudes or emotions [14].

Albert One was developed in 1997 by Robby Garner. It mimics the behavior of a human and in 2001 it was listed in the Guinness book of world records. It is based on the FRED framework [48].

In the first decade of the 2000s, harvesting of knowledge automatically from text and web sources became a major area of research [16]. Construction of a knowledge base often harvests knowledge from textual sources through NLP methods [16]. NLP methods range from classical techniques such as word tagging and word sense lexicons to more recent methods on context-sensitive language models using neural embeddings [16]. Harvesting of knowledge from videos using NLP has still not been researched in detail.

A pipeline for creating a Knowledge Base of entities related to vehicle complaint understanding from short texts, specifically posts in public Questions and Answers (QA) forums [46].

Auto-growing knowledge graph using BERT - expands data sources through crawlers and extract new relational knowledge. Analyze the collected data from social media, news, and other sources to derive the connection with word [39].

Watson process valuable information often exists in forms of language that can be hard for computers to understand: PDFs, charts, tables, call logs, handwritten documents,

blog posts, news articles [3] and tweets. In Automatic corpus expansion it identifies the seed documents and from the web retrieves related documents [15][31]. Next it extracts self-contained text nuggets from those documents [15][31]. Score the nuggets based on whether they are informative with respect to the original seed document [15][31]. Most informative nuggets were merged into the expanded corpus [15][31].
Bot framework uses general knowledge, pdf, and data sources. Additionally, it can be extended by cognitive tasks speech, language, vision, knowledge and search [23][28][17].

All the information will be gathered through Kbot from multiple text sources and present the data in the form of a knowledge card [2].

JAiCOB - receives the question vector and the answer type from the question processor as an input. The question vector is, in essence, a list of keywords ordered by importance. The relevant document and information that tallies with the keywords will be retrieved by generating queries from Elasticsearch [11].

An Automatic Ontology Generation Framework with An Organizational Perspective - An unstructured text corpus is converted into domain consistent ontological form [37].
Automatic Ontology Generation Using Database Schema Information - A methodology which reads an OWL-DL-level ontology based on the schema information, which is the metadata of the relational database, and converts the individual of the ontology into a relational database [7].

2.3 Breakthrough in chatbots (Latest/Modern)

Smart personal voice assistants brought forward the development of Artificial intelligence chatbots [14] which has the ability to understand voice commands, talk using digital voices and different types of tasks were handled [14].

SmartChild was developed in 2001. which was invented by ActiveBuddy. It is also known as a thing which came before Samsung's VOICE and Apple's Siri [43][18].

In 2008 Ron C Lee developed a web-based chatbot. Tutor Mike was developed to help the students who are learning English [33]. In 2018 Tutor Mike won the second place of the Loebner Prize competition [22][32].

Siri was a personal assistant developed in 2010 by Apple. Communications and Inquiries were made by the user through voice commands. Using various internet services Siri makes recommendations and responds to user requests [14]. The main weakness of Siri is it requires an internet connection [14].

Watson was introduced in 2011 by IBM which has the capability to understand natural human language [1]. Watson is a question answering system which applies techniques like machine learning technologies, knowledge representation, information retrieval, automated reasoning and natural language processing [1].

Google Now was developed in 2012 of which the initial focus was to provide information to users by considering the preference, location and time of day [14]. The next generation of Google Now is Google Assistant which was developed in 2016 which uses in-depth Artificial Intelligence [14].

Microsoft's Xiaolce is a social chatbot which was launched in China in May 2014[35]. It works as a long-term companion to the person who uses it. It is Intelligent Quotient, Emotional Quotient and designed to have a personality. It has the ability to core chat which has the ability to maintain a lengthy conversation. For state tracking this system uses a dialogue manager [30][10]. This was developed in an empathetic computing framework. It has the ability to understand the user intents and respond accordingly in a dynamic manner. It's aim is to pass the time-sharing test which is a form of the Turing Test. Xiaolce consists of a paired database which extracts data from two data sources, one is conversational data from the internet (public forums, bulletin boards, forums and social networks) and the second form of the database which has been constructed from the human-machine conversation with Xiaolce [49].

In 2014 Amazon and Cortana were introduced [14]. Cortana recognizes voice and performs tasks like sending emails and texts, playing games and chitchats [14]. It is said that Cortana has the ability to run a program with malware which was a major drawback of it [14].

In 2021 pocket therapist was developed which is an intelligent chatbot which gives psychological support for mental health situations during COVID-19 pandemic. This is developed to improve psychic conditions of human. It was developed using Natural Language Processing and sequence-by-sequence architecture. It has the capability of expressing emotions and it is adoptive. The chatbot is based on deep learning technologies. At first it understands the conversation context then it does the motion recognition and gives the response according to the patient's situation. This uses web scraping to find the best answer for the person [40].

2.4 Challenges (future trends/issues) in chatbots

One of the main challenges [24] of the chatbot is to handle NLP issues by mastering their syntax. If the question " what's the weather?" was asked from a chatbot a reply will be received but for the question "Could you check the weather?" may not receive a proper answer [24]. Such types of programming issues fall in the NLP category and the companies such as Facebook, Google with Deep Text and Syntax Net are focusing on these areas [24].

Two aspects of designing and development of Chatbots are NLP and Machine Learning. Our computer systems should be able to learn the correct response which can be achieved with efficient programming with AI concepts [4].

In chatbots distinguishing between identifying problems and giving recommendations could help in developing trust and loyalty by the actors involved [11]. Before conversations begin, how they are to be structured should be explicitly stated - that is, if they are to explore a broad topic or to focus on a specific issue [11], if all the single contributions shared by the interlocutors are taken into account if specific utterances and declarations are given more weight [9].

Unfortunately, chatbots require experienced developers and highly professional programming skills [1]. A complicated development platform is used to develop every chatbot [1]. A chatbot may also lead to the incurring of additional costs such as installation or maintenance costs or both [42].

Having the right knowledge base is very important for the development of a chatbot [1]. A chatbot needs to have access to knowledge of the domain it operates in, to converse intelligently [45]. The aid of a bot developer is required to build the knowledge base [1].

Large numbers of knowledge sources from many domain areas such as formal ontologies, reference terminologies and knowledge bases have been published on the internet and in literature, making them highly accessible and attractive for reusing [13]. However, there are still only a minimum number of knowledge-based applications that have been built through reusing available knowledge sources [13]. Reusing existing knowledge mostly requires a large effort [13]. Developing the core ontology of a knowledge base manually by including information from all sources is too tedious [13]. While there are tools available for developing ontologies (such as Protege-2000) and tools capable of merging and combining knowledge sources, the reuse of knowledge sources is still a labor-intensive process [13].

For the creation of a knowledge base, a process called knowledge engineering is required where knowledge about the specific domain is obtained from human experts and other sources by knowledge engineers [44]. Knowledge from experts is collected through interviews and other forms of questioning methods [26]. Hence, an expert's presence is required throughout the knowledge base creation process.

In the first decade of the 2000s, harvesting of knowledge automatically from text and web sources became a major area of research [3]. Construction of a knowledge base often harvests knowledge from textual sources through NLP methods [16]. NLP methods range from classical techniques such as word tagging and word sense lexicons to more recent methods on context-sensitive language models using neural embeddings [16]. Harvesting of knowledge from videos using NLP has still not been researched in detail.

Table 2 1 Summary of the benefits and issues of the previous work

Research	**Contribution**	**Limitations**
[2]	A knowledge graph-based chatbot system over linked data, optimized for community interaction	Limited text-based data sources with privacy preservation and use of limited knowledge bases
[11]	A modular architecture chatbot that enables adaptation of students' real pedagogic needs to the design of the architecture, while being flexible in maintaining a conversation	Limited information in the Knowledge Base and corresponding Dialogflow intents.
[14]	The study organizes critical information that is a necessary background for further research activity in the field of chatbots	Views of different authors have been considered without systematically evaluating them and only English bibliography have been considered, ignoring works in other languages
[17]	Cloud search service with built-in AI capabilities that enrich all types of information to help identify and explore relevant content at scale	The number of knowledge bases allowed and number of deep links that can be crawled for extraction of question answer pairs from a URL is limited
[23]	A comprehensive framework for building enterprise-grade conversational AI experiences	Supports only a few programming languages and few operating systems

2.5 Problem definition

Generating a knowledge base for a chatbot might be time-consuming. The research problem is the identification of an accurate method for harvesting knowledge from different media (video, text) for creation of the knowledge base. Most of the current chatbots are being developed with the knowledge of the developers or experts on the specific domain. It is expensive and requires programming skills.

2.9 Summary

This chapter presented a critical review of literature of chatbots and identified achievements and research challenges in the field. It has been revealed that chatbots suffers due to lack of knowledge base and the once that are already available are expensive, needs application development resources and they are not reusable. Therefore, in order to address above concerns the knowledgebase development should be automated for chatbots. In the next chapter a detailed discussion is provided on the technology adopted on implementing chatbots.

Chapter 3

3.0 TECHNOLOGIES USED FOR CHATBOTS

3.1 Introduction

In chapter 2 a comprehensive critical review of literature on the use of chatbot generation was presented. The research problem was defined as the identification of an accurate method for harvesting knowledge from different media (video, text) for creation of the knowledge base. In order to solve this problem, we have recognized the need for research into auto generated knowledge bases for chatbots. This chapter describes theory behind chatbots and KB generation and their further developments

3.2 Technologies of Chatbots

A chatbot can be classified into different categories: knowledge domain, service provided, input processing response generation methods, goals and human aid.
Knowledge domain-based classification considers the amount of knowledge that can be accessed by a chatbot or the amount of data it can train. Further, knowledge domain can be categorized into two sections namely open domain and closed domain. Open domain chatbots discuss a general domain and provide the response accordingly. However, Close domain chatbots focus on a specific area or domain where it might fail to respond to other domain queries.

Service provider is chatbot's sentimental proximity to the user. Service provided can be classified into interpersonal, intrapersonal and inter-agent chatbots. Interpersonal chatbot will lie in the communication domain. It provides services such as restaurant bookings. Those aren't companions of the users. It has the ability to have a personality, can be friendly and it might remember the information of the user. It is not obligent and it exists within the user's personal domain. Intrapersonal chatbots occur within the user's personal domain. Inter-agent chatbots are continuously encountered.
Goal based classification are the primary goal chatbots. Categorized into informative, Conversational/Chatbased and task based. Informative chatbots provide information

to the user that are taken from a fixed source or stored beforehand. Conversational/Chatbased mimic the behavior of a human being and communicate like another human to the user. Task-based chatbots are responsible for performing a specific task.

Input, processing and response generation methods consist of three models namely Rule-based model, retrieval-based model and generative model. Rule based model responses are chosen based on a set of predefined rules, which is based on lexical form recognition of the input text trying to create new text answers. Knowledge is humanly hand-coded and conversational patterns are used to organize and present it. Retrieval based model uses APIs to query and analyze resources. Generative model answers the user by considering the current and previous messages. It is more human-like and it uses deep learning techniques and machine learning algorithms.

At least one element in the chatbot will be utilized from human computation in human aided chatbots

3.3 Technologies of Knowledge base development

A technology applied to store all the completed structured and the unstructured information that are being used in a computer system is a knowledge base technology. The initial term of the first knowledge base system is the connection with the expert systems. Different machines, techniques and measures in knowledge-based systems are being applied in implementing heuristic human reasoning in order to capture different types of traditional algorithmic solutions for the problems. Common-KADS and Protege are the engineering tools and methods that were created to streamline in developing the KBSs in 90s where large quantities of developed and jects were implemented [8].

KBS is a program used to query or extend the knowledge base. However, a knowledge base is expressing all the collected knowledge using a language representation. Furthermore, a KBS is a computer system that is being programmed with artificial intelligence in initiating problem-solving methods for humans while referencing a database of knowledge relevant to a specific subject. In other words, based on Artificial intelligence viewpoint system based artificial technologies and methods are KBSs. The three types of KBSs are neural networks, case-based reasoning systems and expert systems. Further KBSs can be considered as the software and the hardware system that supports to complete a specific task with a specific form of knowledge [8].

However, when comparing the knowledge-based system with the Knowledge management system has a huge scope. Where knowledge management includes process, technologies, knowledge, people and organizational culture. At present the KBSs are being developed by using the knowledge of the engineering techniques. However, though the knowledge base system is supportive for future directions it is also a drawback with few issues; where it has a poor understanding of the structure and also the knowledge acquisition is expensive [8].

A chatbot can be defined as a program developed to simulate a smart communication on a text [1]. Chatbots use embedded knowledge to identify a user's query and give an appropriate response [1]. Chatbots are developed for specific domains such as education, business, and healthcare [45]. Current developments in NLP and increasing communication technologies and computing power have facilitated fast development and implementation of chatbots in multiple industries [45].

3.4 Technologies of Knowledge graph development

A knowledge graph has three categories: context-rich knowledge graphs, external sensing knowledge graph, Natural Language Processing knowledge graph [5].

Context-rich knowledge graphs gives us the understanding that a simple keyword search or single word importance identification is not a good idea for large scale heterogeneous knowledge corpus. A context-rich knowledge has the ability to graph the files and internal documents with knowledge along with metadata tags.

The second category is external-sensing knowledge graphs that have the ability to map data to internal entities of interest. This piles the external data sources for mapping.

The final category is the Natural Language Processing Knowledge Graph. These work with the human language nuances and the complexity. This has the ability to understand some features like misspelled sentences or words, technical terms, acronyms and more. This has the ability to map the meaning and generate knowledge graphs.

3.5 Technologies of Knowledge extraction

The process is known as RE where we use it to extract information. Types of RE are pattern matching and RE. It matches patterns and extract information. Supervised RE where this uses supervised learning algorithm it contains a set of entities and relations to train, with many labeled as data. RE using Semi supervised and it also uses Bootstrapping with a smaller number of labeled data. The final one is the RE from Unsupervised it does the RE when there are no data with labels [21].

3.6 Chatbot Anatomy

Domain-specific dialogue acts are called intents [19]. Intent identifying has been most prominently used by call center bots, which ask the user “how can I help you?” and subsequently user intent identification to redirect the user to one of N predefined redirection options [19]. Many of the same machine learning algorithms used for DA classification are used for intent identification [19]. Utterances in chatbots are considered as anything which the user says or inputs. Entities are used to modify the intents. Broadcasting is the method of sending messages to the users protectively. Chatbots have channels which are the medium of chatbot conversation. Conversational UIs are the User Interfaces which are used to interact with the user. The term pilot is the stage of development where a small group of people will be testing the chatbot for bugs and improvement. The proof of concept is the development stage where as long as the input is artificially constrained it will function properly. Chatbot responses are the thing that the chatbot says to the user’s input.

3.7 Summary

This chapter presented a critical review of literature in chatbot technologies and knowledge base technologies and identified achievements in the field and research challenges. In the next chapter a detailed discussion is conducted on the approach.

Chapter 4

4.0 APPROACH

4.1 Introduction

In chapter 3 the discussion was on the technologies that can be used to solve the research problem of the identification of an accurate method for extraction of knowledge from video content for creation of the knowledge base and chatbot intents. A solution has been proposed for Automated Knowledge base and intent generation to address the above problem. The new approach is named as Smart Knowledge Bot, generating a knowledge base using KB technologies. The new approach Smart Knowledge Bot is inspired by numerous evidences of the application of KB technologies for similar projects. Rest of the chapter has been structured with the subheading hypothesis, input/output, process, users and features.

4.2 Hypothesis

The hypothesis of this solution is to generate a knowledge base and chatbot intents based on NLP and machine learning techniques by extracting knowledge from video contents.

4.3 Input

User inputs for the proposed system is keywords related to the area where the user wants to generate the Knowledge Base and chatbot intents.

4.4 Output

A Knowledge Base and chatbot intents generated for the specific domain or multiple domains along with a chat communication.

4.5 Process

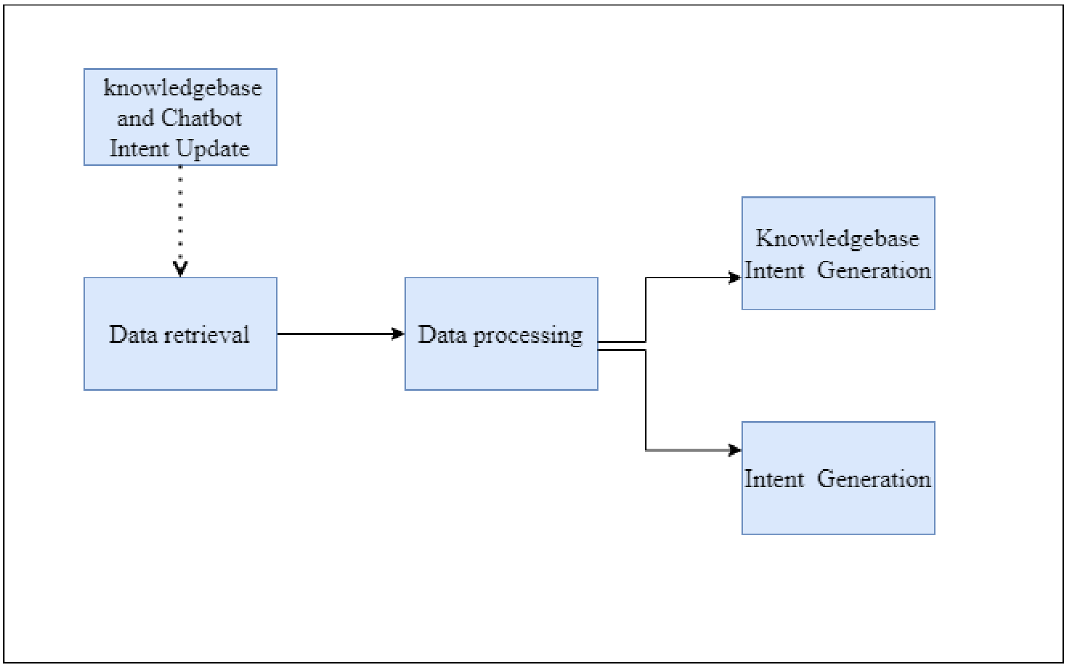


Figure 4 1 High level architecture

As shown in the above Figure 4.1 the proposed method has four main components named Data retrieval module, Data processing module, Knowledge graph generation module, Intent generation module and Knowledge graph and chatbot intent update module. These four main components have sub tasks which are discussed in this document.

In the Data retrieval step, we are retrieving knowledge related to user keywords from video contents. In the data retrieval step the system will use the keywords provided by the user to retrieve information by searching on the internet through YouTube search technology supported by Python [29]. Text extraction and tagging will be done using text mining Spacy and NLP techniques [35].

At the same time Audio clips will be extracted from YouTube videos using Python. Phrases and words will be detected from the audio clips and converted to text format through an NLP Python library. The converted text will be checked for English language and filter out only the English text.

In the data processing step the retrieved dataset is given as the input and the data that has been retrieved and processed through NLP and spacy is sent through logistic regression functionality and stored the correct data in a Neo4j graph database. Neo4j has the ability to tag each node with an appropriate label, which can be used to filter out the necessary information needed for the chatbot [41]. Also, a list of chatbot intents will be generated during this stage of data processing. The data stored in Neo4j will be shown to the user as a preliminary knowledge structure through a user-friendly interface developed through Angular.

In the knowledge graph generation step output from the data processing step will be taken as the input. The system will identify the subject, predicate and object to generate the final knowledge base.

In the intent generation step the system will identify the specific intents which will be easy for a chatbot system to train it and will generate examples under the intent and create a file which has the same structure of a Rasa chatbot framework nlu file which has the intents.

In the knowledge update module, it will take the chatbot user reply on the answers provided and update the knowledge graph. Then the scheduler will run the Data retrieval module, Data processing module, Knowledgebase generation module and Intent generation module for all the keywords which were given to the user previously from a list of keywords.
When the customer communicates with the chatbot, it will query the knowledge base for related intents.

4.6 Users

This research will have many uses in different industries. The people who need to generate a dynamic knowledge base and chatbot intents using the video contents available in YouTube will be the users of this system.

4.7 Feature

Following are the features of the generated knowledge base and the chatbot intents using video contents.

- Facilitate the user to develop a knowledge graph and chatbot intents within a limited time span
- Facilitate the user to create a chatbot in a simpler manner without the need of a software engineer or with minimal programming skills
- This system helps the user to create a knowledge graph and chatbot intents for a chatbot at a lower cost.
- Facilitates to generate a dynamic knowledge graph and chatbot intents using scheduler approach.

4.8 Summary

This chapter presented the approach to chatbot and it provides an understanding on the input, output, process, users and features. In the next chapter a detailed discussion is provided on the design of the proposed system.

Chapter 5

5.0 DESIGN

5.1 Introduction

Chapter 4 presented the approach, Knowledge base and intent creation including video content extraction for a chatbot approach. This chapter presents the top-level architecture of knowledge base generation, eleven modules which comes under five major modules namely Data retrieval module, Data processing module, Knowledgebase generation module, Intent generation module and Knowledgebase and chatbot intent update module.

The Data retrieval module includes four sub modules, namely Data Retrieval from YouTube, Video date filter, Video downloader and Text converter, Data processing module which has three sub modules namely Data extractor, Feature weight assigner and Logistic model correct triplet extractor. Knowledge graph generation module has two sub modules namely Identifying basic nodes and edges and creating a knowledge graph. Intent generation module has two sub components namely user intent and example generator and Rasa nlu file updater. Final component is Knowledge graph and the chatbot intent update module has one sub module namely Knowledge update scheduler. The top-level architecture of design of knowledgebase generator is shown in Figure 5.1

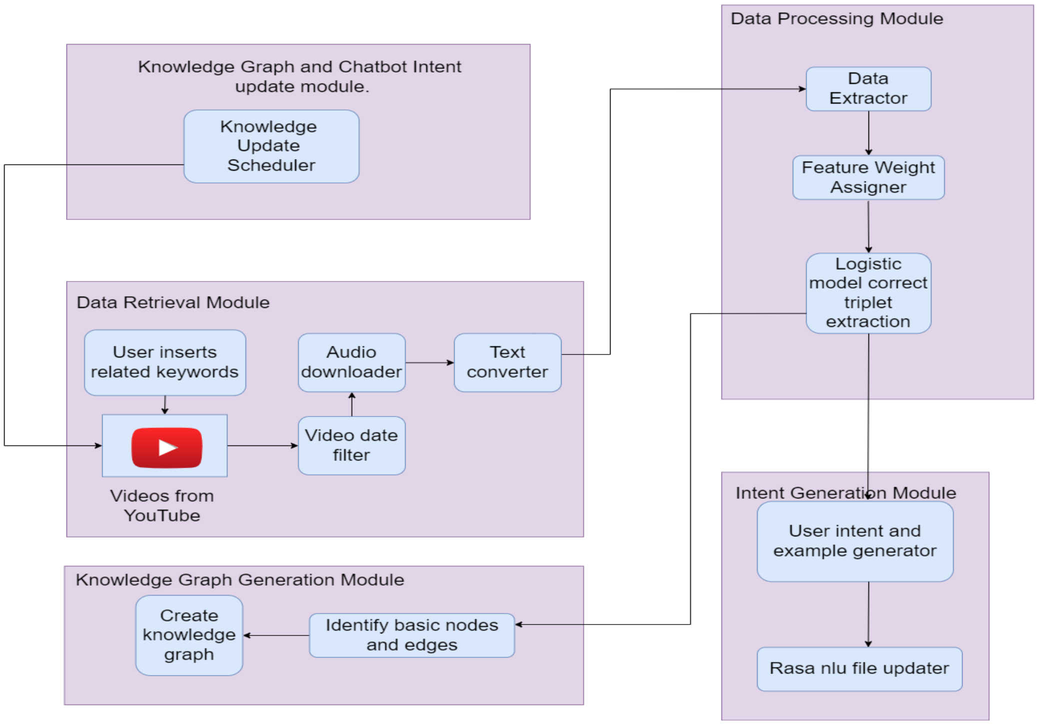


Figure 5 1Top level architecture of knowledge base generator

5.2 Data retrieval module

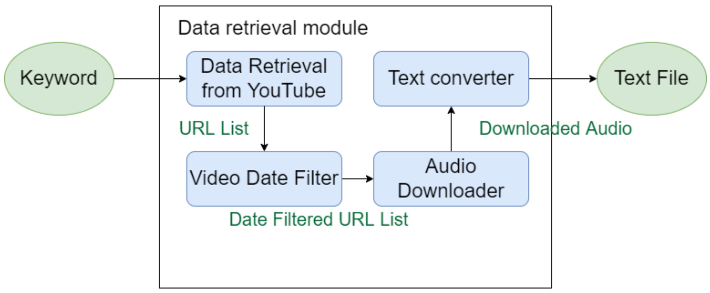


Figure 5 2 Data retrieval module architecture

The data retrieval module processes the input data to retrieve the output and it is sent to the data processing module. When the user inputs the keyword, it uses the keyword of the user as the input to search through the video content and retrieves it. As the output of the module, downloaded text files which match the criteria of the user will be inserted to the data processing module.

5.2.1 Data Retrieval from YouTube

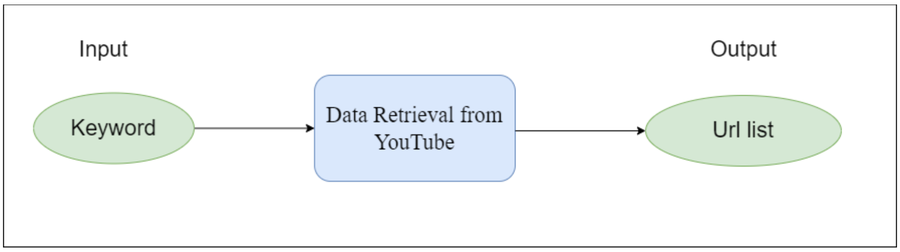


Figure 5 3 Architecture of data retrieval from YouTube

This module will use the keyword entered by the user as the input and search for a list of URLs which matches the user keyword and obtains the list of URLs as the output of this module. Next this URL list will be entered to the video data filter module as its input.

5.2.2 Video date filter

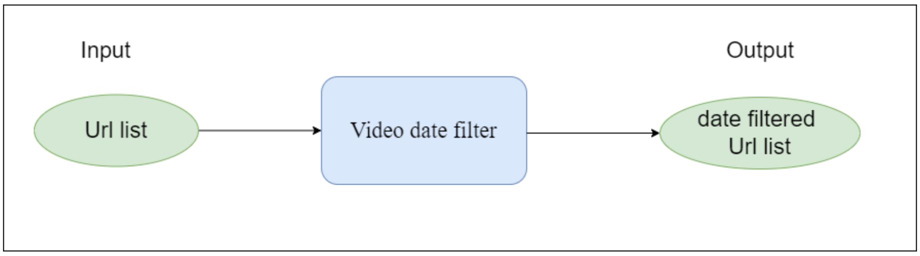


Figure 5 4 Architecture of video date filter

This module takes the list of URLs as the input which is the output of the data retrieval from the YouTube module. This filter out the list of URLs which the information from the videos content isn't used previously for the data extraction and knowledge graph generation. The filtering process is done from the video creation date of the YouTube. This URL list is the output of this module which will be the input of the audio downloader module.

5.2.3 Audio downloader

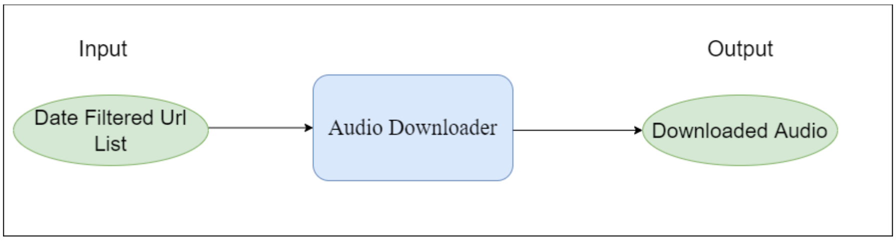


Figure 5 5 Architecture of audio downloader

This module takes the list of date filtered URLs as the input which was the output of the video date filter module. This downloads the list of URLs as audio and audios are the output of this module which will be the input of the text converter module.

5.2.4 Text converter

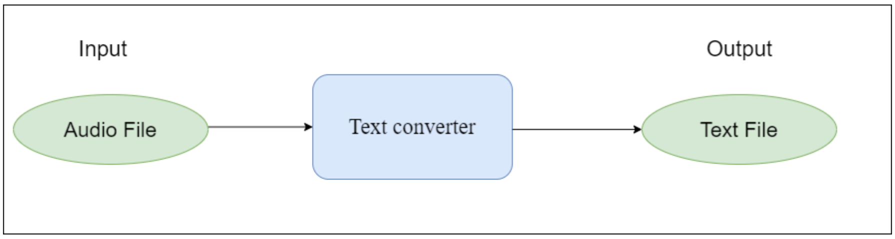


Figure 5 6 Architecture of text converter

This module takes the audio file as the input which was the output of the audio downloader module. It processes the audio files and gives the output as the text file. The input for the data processing module will be a text file.

5.3 Data processing module

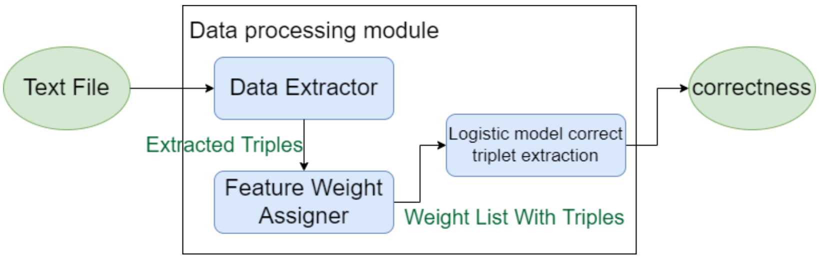


Figure 5 7 Data processing module architecture

The data processing module processes the input which is the text file, extract the data, assign the weights and finally train it by sending it to a logistic regression model. Correct extraction triplets are the output of this module which will be the input of the knowledge graph generation module and intent generation module.

5.3.1 Data extractor

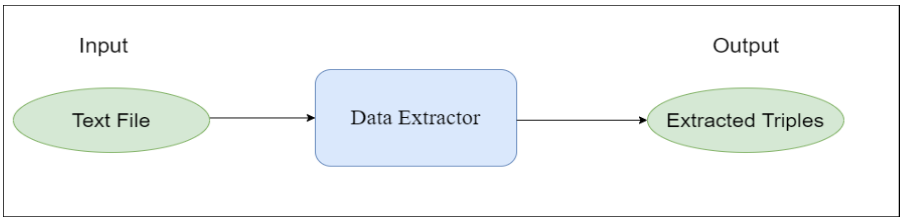


Figure 5 8 Architecture of data extractor

This data extractor module takes the text file as the input and does the data extraction process by triple extraction using NLP and spacy. Extracted triples will be the output of this data extractor module and this output will be the inputs for the feature weight assigner module.

5.3.2 Feature weight assigner

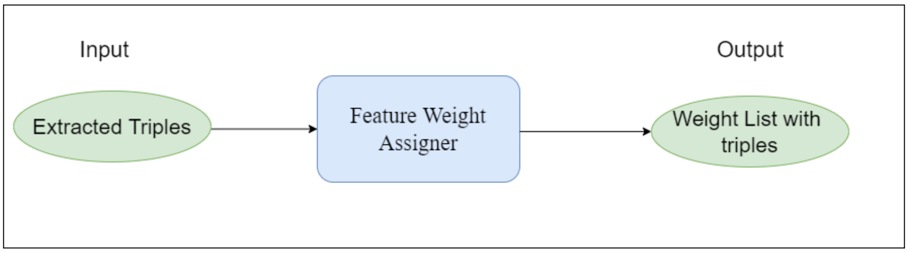


Figure 5 9 Architecture of feature weight assigner

The feature weight assigner module takes the extracted triples as the input which was the output of the data extractor module and process it using spacy and nlp and a list of weights assigned for the features of the triples will be the output of this module. The weight list with the triples will be the input for the logistic model correct triple extraction module.

5.3.3 Logistic model correct triplet extraction

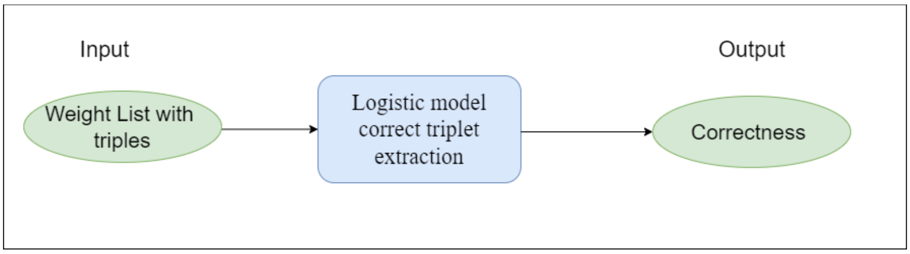


Figure 5 10 Architecture of logistic model correct triplet extraction

In this module weight list with triples will be the input which was the output of the feature weight assigner. This will train the weights by sending it to a logistic regression model and will give the correctness of the triple. This correct triple will be the inputs of the knowledge graph generation model.

5.4 Knowledge graph generation module

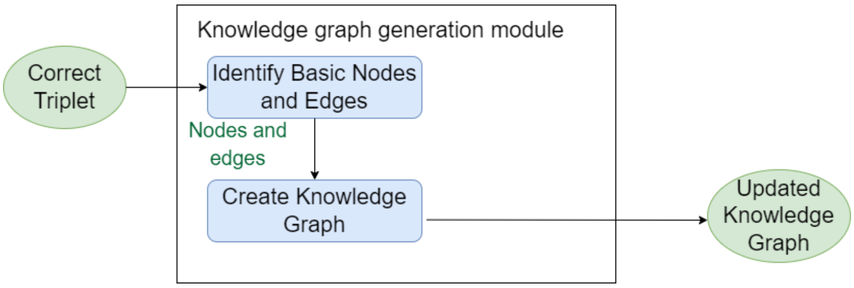


Figure 5 11 Knowledge graph generation module architecture

The knowledge graph generation model takes the output of the data processing model as the input which is the correct triples. Using these triples, it identifies the things which are needed to create the knowledge graph and proceed with processing and generate the knowledge graph.

5.4.1 Identify basic nodes and edges

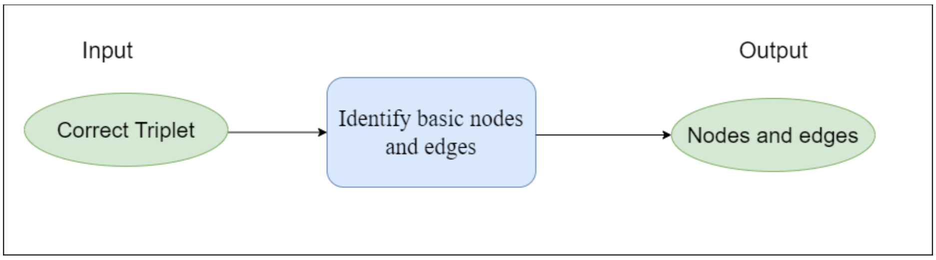


Figure 5 12 Architecture of identify basic nodes and edges

The identified basic nodes and edges module take the correct triple as the input which was the output of the data processing module. This identifies the nodes and edges for a knowledge graph from the input and the identified nodes and edges will be the output of this module and it will be the input of the created knowledge graph module.

5.4.2 Create knowledge graph

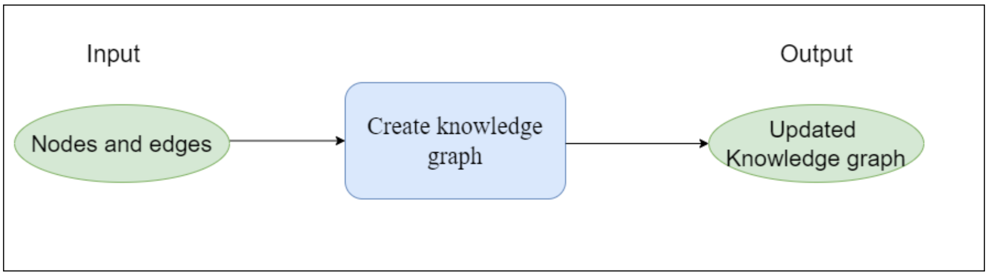


Figure 5 13 Architecture of create knowledge graph

In the create knowledge graph module the identified nodes and edges will be the input which was the output of the identify basic nodes and edges module and it processes the nodes and edges and saves it in the knowledge graph. Updated knowledge graph with the correct triples will be the output of this module.

5.5 Intent generation module

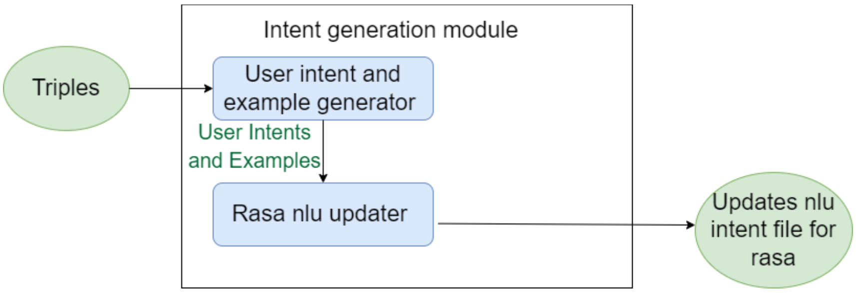


Figure 5 14 Intent generation module architecture

The intent generator module will take the correct triple as the input which was the output of the data processing model. It will process the input and generate the intents and intent examples for the chatbot. The output of this module is the updated nlu (intent) file.

5.5.1 User intent and example generator

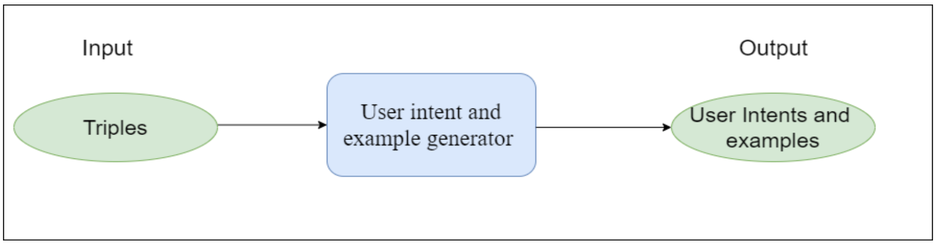


Figure 5 15 Architecture of user intent and example generator

Use intents and example generator module takes the output of the data processing module as the input which was the correct triples. It processes those triples in the user intent and example generator module and it creates intents and intent examples which are related to that intent. The intent and intent examples will be the output of this module and it will be taken as the input of the rasa rule updater module.

5.3.2 Rasa nlu updater

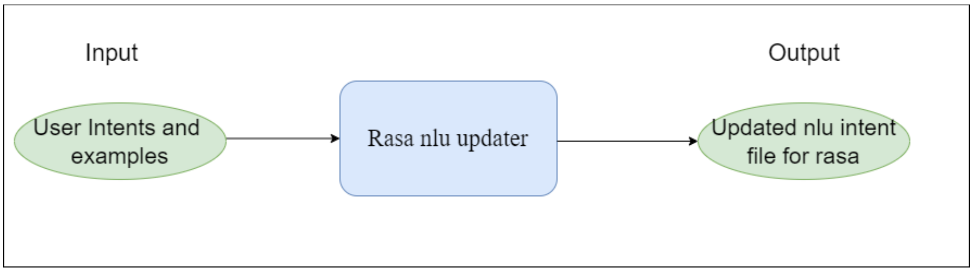


Figure 5 16 Architecture of Rasa nlu updater

In the rasa nlu updater module it takes the user intents and examples as the user input and processes it and it updates the rasa nlu file based on the semantic similarity of the intents. The output of this module will be the updated rasa nlu file.

5.6 Knowledge graph and chatbot intent update module

In the knowledge graph and chatbot intent update module it will take the keyword list(the list of keywords which the user previously used for the knowledge graph and ontology creation).This processes these keywords and gets the output as updated knowledge graphs and intents.

5.6.1 Knowledge update scheduler

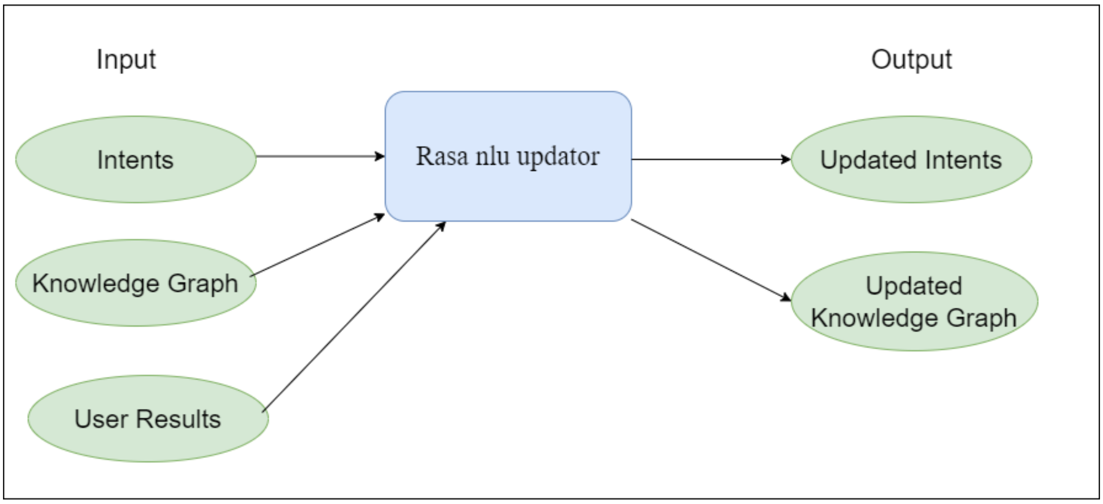


Figure 5 17 Architecture of knowledge update scheduler

The knowledge update scheduler takes the knowledge graph, rasa nlu intents the list of keywords and the Chatbot user results are the inputs of this module which was the output of all the previous modules (Data retrieval module, Data processing module, Knowledge graph generation module and Intent generation module) and the chatbot chat. It processes and runs all the other above-mentioned modules once a week for an updated knowledge graph and chatbot rasa nlu intents and also it will classify whether the extracted knowledge from previous extractions satisfy the users using a Naïve Bayes classifier and update the knowledge graph and intents. The output of this module will be the updated knowledge graph and updated rasa chatbot intents.

5.7 Summary

This chapter presented an overall design of the chatbot knowledge graph generation and intent generation. The next chapter provides a detailed discussion on the implementation of this system.

Chapter 6

6.0 IMPLEMENTATION

6.1 Introduction

Chapter 5 presented the Design, the overall design and the input and output of all the components and sub components. It provided a brief description of the module architecture. This chapter presents the implementation of knowledge graph generation and chatbot intent generation. It has eleven modules which come under five major modules namely Data retrieval module, Data processing module, Knowledgebase generation module, Intent generation module and Knowledgebase and chatbot intent update module.

6.2 Data retrieval module

6.2.1 Data Retrieval from YouTube

In this module it takes the user keyword and generates anode for the specific keyword in the knowledge graph and then writes the keyword-to-keyword text file for later reference when the scheduler is updating the knowledge graph and chatbot intents. It does a YouTube search using the urllib.request library and obtains the list of URLs

6.2.2 Video date filter

This retrieves the last updated creation date from the graph if it is available and keeps track of a maximum date and if there are no last updated days it will take all the URLs. If the last updated creation date is available, it will retrieve the creation date of the YouTube videos using the YouTube data API. Then it takes the URLs which are after the creation date of the last updated creation date and filters out the list of video URLs which are not taken for the process of information extraction.

6.2.3 Audio downloader

This module gets the list of URLs and using the youtube_dil library it downloads the mp3 files of the YouTube videos. Then this audio mp3 file will be mapped to a .wav file using pydub library and using the same library the .wav audio file was split into audio chunks.

6.2.4 Text converter

Those audio chunks were converted to text data using the python speech recognition module. These texts were sent through a language detector to detect whether the extracted text is from the English language. The language checked text was written in a text file for the process of data extraction.

6.3 Data processing module

6.3.1Data extractor

For the purpose of data extraction, we use OIE which comes under Relation extraction via unsupervised learning, the Reverb System. A saved text file is retrieved line by line and used for the information extraction process where the three triplets are found: a subject, predicate and object.

First the predicate is extracted from the given sentence. For the extraction of the predicate, we insert the line and analyze it using a spacy pattern. Spacy is an advanced Natural Language Processing Library which is used in python. It does the part of speech tagging and entity chunking of its own.

```
V | VP |VW*P

V =  ADV*|AUX*|VERB|PART*|ADV*
W = NOUN|ADJ|ADV|DET|ADP
P =  ADP|ADJ
```

Figure 6 1 Syntactic constraint pattern

Using the mentioned pattern in Figure 6.1 a predicate pattern was extracted which matches the syntactic constraints. This takes the longest sequence of the sentence which starts with a verb. After the extraction of the predicate, we use this predicate to extract the subject and related object for that subject.

For the extraction of the subject, we use the noun phrase extractor of spacy and find the noun phrases to the left of the extracted predicate in the sentence.

For the object extraction the same scenario is used just like the subject extraction with a small change. The extracted predicate is sent and finds a noun phrase which is in the right to the predicate in the sentence. Spacy is used for the noun phrase extraction.

6.3.2 Feature weight assigner

In the feature extractor the extracted triples are taken and weights are assigned according to the features. The weights are assigned according to nineteen features which match the characteristics of the extracted data (the subject, predicate and object).

Table 6 1 List of weights

Feature Number	Weight
Feature 1	1.16
Feature 2	0.50
Feature 3	0.49
Feature 4	0.46
Feature 5	0.43
Feature 6	0.43
Feature 7	0.42
Feature 8	0.39
Feature 9	0.25

Feature 10	0.23
Feature 11	0.21
Feature 12	0.16
Feature 13	0.01
Feature 14	-0.30
Feature 15	-0.43
Feature 16	-0.61
Feature 17	-0.65
Feature 18	-0.81
Feature 19	-0.93

The Table 6.1 displays the list of features and the related weights for those features. If the extracted sentence matches any feature the related weight will be added or else a weight of zero will be assigned.

The feature 1 will check whether the original sentence, the sentence which was given to the system from the text file will match to the concatenation of the triple subject, predicate and object. If it matches then a weight of 1.16 will be added to the weight list and if it doesn't match a weight of zero will be added to the weight list.

The feature 2 will get the extracted sentence the combination of subject predicate and object and will check whether the last preposition of the combination of the triple is the word 'for'. If the last preposition matches that word, then a weight of 0.50 will be added to the weight list if the last preposition doesn't match then a value of zero is added to the weight list.

The feature 3 is similar to feature 2 with a small difference, earlier it checked whether the final proposition of the triples matches the word ‘for’ and in feature 3 it will check whether the final preposition matches the word ‘on’ and if it matches that word it will add a weight of 0.49 to the weight list and if it doesn’t match a value of zero will be added.

Feature 4 is also somewhat similar to feature 2 and feature 3 where it checks whether the last preposition of the extracted text which is the combination of subject, predicate and object is the word ‘of’, if it matches that word a weight of 0.46 was added to the weight list or else a weight of zero is added.

In feature 5 it checks whether the length of the extracted sentence matches ten or it is less than ten, if so a weight of 0.43 will be added to the weight list and if it doesn’t satisfy the given condition will add a weight of zero to the weight list.

Feature 6 checks whether the original text, the text before data extraction contains a WH keyword (What, When, Where, Why and Whom) before the predicate value. If contains a WH keyword a weight of 0.43 will be assigned to that extraction and added to the weight list. If it doesn’t contain a value of zero will be added.

The feature 7 checks whether the predicate matches the VW*P pattern. if it matches the condition a weight of 0.42 will be assigned as the weight and will be added to the weight list or else a value of zero will be added to the list.

The feature 8 checks whether the last proposition of combination of subject, predicate and object matches the word ‘to’ if it matches the word a weight of 0.39 will be assigned as the weight and will be added to the weight list which will be used to get the prediction of logistic regression model or else a value of zero will be added to the list.

Feature 9 checks whether the last preposition of the extracted text matches the word “in” and if it matches that word then a weight of 0.25 will be assigned to that extraction and it is added to the weight list and if it doesn't satisfy the condition, it will add a weight of zero to the list of weights.

In feature 10 It checks whether the total word length of the extracted text is greater than ten and total length is less than twenty then a weight for that extraction is given as 0.23 which will be added to the weight list if the total length doesn't lie on these two boundaries, then a weight of zero was given.

Feature 11 checks whether the original sentence before data extraction starts with the extracted subject from the data extraction process. If it starts with the subject then a weight value of 0.21 will be added to the weight list and if it doesn't start with the subject then a value of zero will be added to the weight list.

The feature 12 checks whether the extracted object is a proper noun. If the extraction is a proper noun a weight of 0.16 will be assigned to the weight list or else a weight of zero will be added to the list.

In feature 13 it checks whether the extracted subject is a proper noun and if it is a proper noun a weight of 0.01 is added to the list or else the weight will be added as a zero to the list if it doesn't satisfy the condition.

Feature 14 checks whether a noun phrase is available to the left of the subject of the original sentence before extraction. If a noun phrase is available then a weight of negative 0.30 is added to the weight list or else a weight of zero is added to the weight list.

The feature 15 in this system will check whether the extracted subject predicate and object altogether has a word length greater than 20. If the word length is greater than 20 a negative 0.43 will be assigned to the weight list and if the sentence doesn't match these criteria a zero will be added to the weights.

The feature 16 will check whether the predicate matches pattern V a weight of -0.61 will be assigned or else a weight of zero will be added to the weight list.

The feature 17 will check whether the original sentence contains a preposition before the extracted subject and if a preposition contains then a weight of -0.65 will be assigned to the data extraction or else a weight of zero will be added to the weight list.

The feature 18 which is the last feature of this system will check whether there is a noun phrase to the left of the extracted object in the original sentence. If it contains a noun phrase then a value of negative 0.81 will be added to the weight list or else a value of zero will be added to the weight list since it doesn't match the condition.
If any of the extracted components (subject, predicate or object) or component is null then a weight of zero to all the features will be assigned.

The feature 19 checks whether a coord, conjunction to the left of predicate in original sentence. If the condition meets the criteria a weight of negative 0.93 will be added to the list of weights or else a weight of zero will be added.

Before moving into the logistic regression model, we scraped 1,000 web sentences and extracted the given data using the extraction rules and created a list of extractions and then a weight for each feature was assigned using the feature weight assigner. All the information were written to an excel file and the excel file consisted with columns for the original sentence, extracted test (combination of the extracted triples) subject, predicate, object, weights for each feature in a separate column and the correctness. The correctness was labeled after the data extraction process after running all these processes by the system and create the dataset. The extractions were hand labeled as correct or incorrect using 1 and 0. So The structure of the dataset for the training of the logistic regression model is presented in the below Figure 6.2.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	Original Sentenc	Extraction	Subject	Predicate	Object	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11	f12	f13	f14	f15	f16	Correctness	
2	Sahara from Aral	Arabic sahara de	Arabic sahara de	largest desert in		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Figure 6 2 Dataset Structure

6.2.3 Logistic model correct triplet extraction

In this module the knowledge extracted will be classified as correct or incorrect using a logistic regression. An input weight list will be given as mentioned in the below Figure 6.3.

```
trainData([0,0,0,0,0,0.43,0,0,0,0,0,0,0,0.21,0,0,-0.3,0,0])
```

Figure 6 3 Weight list

The weights for the features are sent to the logistic regression model to find the correctness of a given data.

```
col_names = ['f1', 'f2', 'f3', 'f4', 'f5', 'f6', 'f7', 'f8', 'f9' ,'f10', 'f11', 'f12', 'f13' ,'f14', 'f15',
             'f16' ,'f17', 'f18', 'f19','label']  # load dataset

# load dataset
pima = pd.read_csv("E:/University of Moratuwa/2nd year/research/final project/NEW/smart-bot/DataSet/aa.csv",
                   header=None, names=col_names)

print(pima.head())
```

Figure 6 4 Loading dataset

Figure 6.4 shows how the features and the label which is used to mark whether the extracted data is correct or incorrect are loaded to a data frame for the logistic model to train.

```
#split dataset in features and target variable
feature_cols = ['f1', 'f2', 'f3', 'f4', 'f5', 'f6', 'f7', 'f8', 'f9', 'f10', 'f11', 'f12', 'f13', 'f14',
                'f15', 'f16' ,'f17', 'f18', 'f19']
X = pima[feature_cols] # Features
y = pima.label # Target variable

X = X.apply(pd.to_numeric, errors='coerce')
y = y.apply(pd.to_numeric, errors='coerce')

X.fillna(0, inplace=True)
y.fillna(0, inplace=True)

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.25,random_state=0)
```

Figure 6 5 Splitting data

Figure 6.5 displays how the features are loaded to X as features and the target variable the correct or incorrect value of the extracted text was assigned to y variable as numeric values. It will split to X_train, X_test, y_train and y_test with a test size of 0.25 and a random state of zero.

```
logreg = LogisticRegression()
```

Figure 6 6 Defining model

The figure 6.6 shows how the logistic regression model is instantiated with the available parameters.

```
y_pred=logreg.predict(X_test)
```

Figure 6 7 Training text data

As mentioned in the Figure 6.7 the correctness or incorrectness of the test data set will be predicted passing it to the logistic regression predict function.

```
correctness = logreg.predict([weightDataList])[0]
```

Figure 6 8 Predict using a weight list

Figure 6.8 displays how the implementation was done to predict the correctness of the weight list provided from extraction and weight assigning.

6.4 Knowledge graph generation module

6.4.1 Identify basic nodes and edges

Subject, predicate and object will be the extractions. The Subject and Predicate will be identified as nodes and predicate as edge. First, we check whether the subject exists as a node Figures 6.9, 6.10, and 6.11 will illustrate the code segment for that.

```
value = connector.returnValContain(subjectNode)
```

Figure 6 9 Subject contain checker caller

```
def returnValContain(self,subject):
    with self.driver.session() as session:
        return session.write_transaction(self.returnValueContain,subject)
```

Figure 6 10 Subject contain checker session creator

```
@staticmethod
def returnValueContain(tx,subject):
    return tx.run("MATCH (n :Ontology)-[r:contains]->(s:"+subject+")"
                  "RETURN 1").single()
```

Figure 6 11 Subject contain checker query

Thereafter it will check whether the subject value is contained under the keyword node. The Figure 6.12, 6.13 and 6.14 illustrate the mentioned process.

```
value2 = connector.returnNestValContain(subjectNode,keywordNode)
```

Figure 6 12 Subject and keyword contain checker caller

```
def returnNestValContain(self,subject,keyword):
    with self.driver.session() as session:
        return session.write_transaction(self.returnNestedValueContain,subject,keyword)
```

Figure 6 13 Subject and keyword contain checker creator

```
@staticmethod
def returnNestedValueContain(tx,subject,keyword):
    return tx.run("MATCH (n :Ontology)-[r:contains]->(s:"+keyword+")-[t:has]->(u:"+subject+")"
                  "RETURN 1").single()
```

Figure 6 14 Subject and keyword contain checker query

6.4.2 Create knowledge graph

```
if(value2 != None or value !=None):
    connector.matchNodeCreator(subjectNode,predicateNode,objectNode)
```

Figure 6 15 Check value none and node caller

The above given code checks whether one of the two conditions which were checked previously is not none. If one of the conditions is not none it will create a node for the object and it will create the connection for the subject in the knowledge graph. It creates the object node and generates the mapping between the keyword and object using the predicate. The figure 6.16 and figure 6.17 illustrate the implementation.

```
def matchNodeCreator(self,subject,predicate,object):
    with self.driver.session() as session:
        return session.write_transaction(self.createMatchedNode,subject,predicate,object)
```

Figure 6 16 Check value none and node creator

```
@staticmethod
def createMatchedNode(tx,subject,predicate,object):
    tx.run("CREATE (n:"+object+")")
    tx.run("MATCH (c:"+subject+"), (d:" + object + ")"
           "CREATE (c)-[r:"+predicate+"]->(d)")
```

Figure 6 17 Check value none and node query

The Figures 6.18, 6.19 and 6.20 will illustrate how the nodes are created when the extracted subject consists of the keyword searched.

```
elif(keyword in subject):
    connector.dataNodeSubAvailableCreator(predicateNode,objectNode,keywordNode)
    connector.close()
```

Figure 6 18 Keyword in subject node creator caller

```
def dataNodeSubAvailableCreator(self,predicate,object,keyword):
    with self.driver.session() as session:
        session.write_transaction(self.createDataNodeKeywordExisist,predicate,object,keyword)
```

Figure 6 19 Keyword in subject node creator

```
@staticmethod
def createDataNodeKeywordExisist(tx,predicate,object,keyword):
    tx.run("CREATE (n:"+object+")")
    tx.run("MATCH (c:"+keyword+"), (d:" + object + ")"
           "CREATE (c)-[r:"+predicate+"]->(d)")
```

Figure 6 20 Keyword in subject node creator query

If none of the conditions are satisfied it will create nodes subject and object and make connections between keyword and subject and also with subject and object using the predicate. Figures 6.21, 6.22 and 6.23 illustrate the explained functionalities.

```
else:
    connector.dataNodeCreator(subjectNode,predicateNode,objectNode,keywordNode)
    connector.close()
```

Figure 6 21 All nodes creator caller

```
def dataNodeCreator(self,subject,predicate,object,keyword):
    with self.driver.session() as session:
        session.write_transaction(self.createDataNodes,subject,predicate,object,keyword)
```

Figure 6 22 All nodes creator

```
@staticmethod
def createDataNodes(tx,subject,predicate,object,keyword):
    tx.run("CREATE (n:"+subject+")")
    tx.run("CREATE (n:"+object+")")
    tx.run("MATCH (a:"+keyword+"), (b:" + subject + ")"
           "CREATE (a)-[r:has]->(b)" )
    tx.run("MATCH (c:"+subject+"), (d:" + object + ")"
           "CREATE (c)-[r:"+predicate+"]->(d)")
```

Figure 6 23 All nodes creator query

6.5 Intent generation module

6.5.1 User intent and example generator

```
questionList =["who is","what is","when is","where is" ,"wht is","how is"]
exampleList = []
for qkey in questionList:
    sentence = gc.grammerCorrector(qkey +" " + predicate + " in "+ subject)
    exampleList.append(sentence)
```

Figure 6 24 Wh word adder

As given in the above Figure 6.24 it will take the predicate and the subject as the input and will have a predefined wh list and it will each keyword in the list with the predicate with the subject and generate the intent. send it through a grammar checker (GingerIt) and check the grammar and create the intent.

6.5.2 Rasa nlu updater

```
availability = sc.similarityChecker(keyword,subject,predicate)
if availability ==1:
    return
```

Figure 6 25 Availability checker

```
maxValue = 0
thresholdValue = 0.8
maxValueSentence = ""
with open(file_to_read) as file:
    for line in file:
        # print(line)
        doc1 = nlp(line)
        doc2 = nlp(searchQuery + " " + subject + " " + predicate)
        if maxValue < doc1.similarity(doc2):
            maxValue = doc1.similarity(doc2)
            print(doc1.similarity(doc2))
            maxValueSentence = line.replace(" ","_").rstrip()
```

Figure 6 26 Similarity checker with current intent

```
if maxValue >= thresholdValue:
    print("Similarity value is "+str(maxValue)+" Threshold value is "+str(thresholdValue))
    info = ec.intentExampleList(subject, predicate)

    # add verticle bar
    class AsLiteral(str):
        pass

    def represent_literal(dumper, data):
        return dumper.represent_scalar(BaseResolver.DEFAULT_SCALAR_TAG,
                                       data, style="|")

    yaml.add_representer(AsLiteral, represent_literal)

    info_str = AsLiteral(yaml.dump(info))
```

Figure 6 27 Threshold value exceed checker

```
    with open('E:/University of Moratuwa/2nd year/research/final project/NEW/smart-bot/DataSet/intents.yml', 'rt',
              encoding='utf-8') as yamlfile:
        cur_yaml = yaml.safe_load(yamlfile)

        for i in range(len(cur_yaml)):
            if cur_yaml[i]["intent"] == maxValueSentence:

                cur_yaml[i]["examples"] = cur_yaml[i]["examples"] +  "\n- "+info[1] +info[2]

                if cur_yaml:
                    with open(
                            'E:/University of Moratuwa/2nd year/research/final project/NEW/smart-bot/DataSet/intents.yml
                            'w') as yamlfile:
                        yaml.safe_dump(cur_yaml, yamlfile, sort_keys=False)
    return 1
else:
    return 0
```

Figure 6 28 Example writer

The figures 6.25, 6.26, 6.27 and 6.28 illustrates how the similarity of the extracted intent is checked and updates the examples of existing intents if the threshold value exceeds.

The threshold value for the semantic similarity intent checker was computed using the ROC curve as illustrated below Figures 6.29 and 6.30.

Threshold Value = 0				
sentence	TP	FP	TN	FN
gallefort first baptism, "was at"	1			
world was, "was happened"		1		
sinharaja, "was at"		1		
sigiriya, "maximum height"		1		
gatabe temple, "located at"		1		
gallefort baptism, "took place"	1			
"took place", "was at"		1		
gallefort took place, "baptisam"	1			

Threshold Value = 0.1				
sentence	TP	FP	TN	FN
gallefort first baptism, "was at"	1			
world was, "was happened"		1		
sinharaja, "was at"		1		
sigiriya, "maximum height"		1		
gatabe temple, "located at"		1		
gallefort baptism, "took place"	1			
took place "was at"		1		
gallefort took place, "baptisam"	1			

Threshold Value = 0.2				
sentence	TP	FP	TN	FN
gallefort first baptism, "was at"	1			
world was, "was happened"		1		
sinharaja, "was at"		1		
sigiriya, "maximum height"		1		
gatabe temple, "located at"		1		
gallefort baptism, "took place"	1			
took place "was at"		1		
"held on", "baptisam"	1			

Threshold Value = 0.3				
sentence	TP	FP	TN	FN
gallefort first baptism, "was at"	1			
world was, "was happened"		1		
sinharaja, "was at"		1		
sigiriya, "maximum height"		1		
gatabe temple, "located at"		1		
gallefort baptism, "took place"	1			
took place "was at"		1		
gallefort took place, "baptisam"	1			

Threshold Value = 0.4				
sentence	TP	FP	TN	FN
gallefort first baptism, "was at"	1			
world was, "was happened"		1		
sinharaja, "was at"		1		
sigiriya, "maximum height"			1	
gatabe temple, "located at"		1		
gallefort baptism, "took place"	1			
took place "was at"		1		
gallefort took place, "baptisam"	1			

Threshold Value = 0.5				
sentence	TP	FP	TN	FN
gallefort first baptism, "was at"	1			
world was, "was happened"		1		
sinharaja, "was at"		1		
sigiriya, "maximum height"			1	
gatabe temple, "located at"		1		
gallefort baptism, "took place"	1			
took place "was at"		1		
gallefort took place, "baptisam"	1			

Threshold Value = 0.6				
sentence	TP	FP	TN	FN
gallefort first baptism, "was at"	1			
world was, "was happened"		1		
sinharaja, "was at"		1		
sigiriya, "maximum height"			1	
gatabe temple, "located at"		1		
gallefort baptism, "took place"	1			
took place "was at"		1		
gallefort took place, "baptisam"	1			

Threshold Value = 0.7				
sentence	TP	FP	TN	FN
gallefort first baptism, "was at"	1			
world was, "was happened"		1		
sinharaja, "was at"		1		
sigiriya, "maximum height"			1	
gatabe temple, "located at"			1	
gallefort baptism, "took place"	1			
took place "was at"		1		
gallefort took place, "baptisam"	1			

Threshold Value = 0.8				
sentence	TP	FP	TN	FN
gallefort first baptism, "was at"	1			
world was, "was happened"			1	
sinharaja, "was at"			1	
sigiriya, "maximum height"			1	
gatabe temple, "located at"			1	
gallefort baptism, "took place"	1			
took place "was at"		1		
gallefort took place, "baptisam"				1

Threshold Value = 0.9				
sentence	TP	FP	TN	FN
gallefort first baptism, "was at"	1			
world was, "was happened"			1	
sinharaja, "was at"			1	
sigiriya, "maximum height"			1	
gatabe temple, "located at"			1	
gallefort baptism, "took place"	1			
took place "was at"			1	
gallefort took place, "baptisam"				1

Threshold Value = 1				
sentence	TP	FP	TN	FN
gallefort first baptism, "was at"				1
world was, "was happened"			1	
sinharaja, "was at"			1	
sigiriya, "maximum height"			1	
gatabe temple, "located at"			1	
"gallefort baptism", "took place"				1
took place "was at"			1	
gallefort took place, "baptisam"				1

Figure 6 29 TP, FP, TN, FN values for different thresholds.

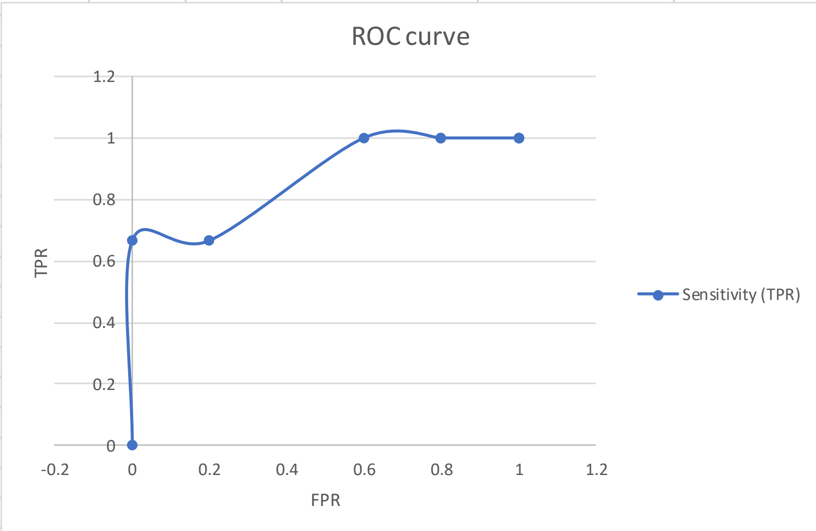
Threshold va	TP	FP	TN	FN	Specificity	FPR	Sensitivity (TPR)
0	3	5	0	0	0	1	1
0.1	3	5	0	0	0	1	1
0.2	3	5	0		0	1	1
0.3	3	5	0	0	0	1	1
0.4	3	4	1	0	0.2	0.8	1
0.5	3	4	1	0	0.2	0.8	1
0.6	3	4	1	0	0.2	0.8	1
0.7	3	3	2	0	0.4	0.6	1
0.8	2	1	4	1	0.8	0.2	0.666666667
0.9	2	0	5	1	1	0	0.666666667
1	0	0	5	3	1	0	0

TPR = Sensitivity

FPR = 1-Specificity

$$TPR = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$FPR = \frac{False\ Positive}{False\ Positive + True\ Negative}$$



Favour a low FPR, but don't want an abysmal TPR, so 0.8 is selected.

Figure 6 30 ROC Curve and threshold selection

In Figure 6.31 it illustrates how the intents and examples are written to the intent.yml file if there are no semantic similarity with the available intents and generated intent. The intent.yml file has predefined basic intents previously and then the auto generated ones will be attended to the same file.

```python
class AsLiteral(str):
    pass

def represent_literal(dumper, data):
    return dumper.represent_scalar(BaseResolver.DEFAULT_SCALAR_TAG,
            data, style="|")

yaml.add_representer(AsLiteral, represent_literal)

info_str =  AsLiteral(yaml.dump(info))

#dictionary to be written
dict_file = [{'intent': subject +"_" + predicate,'examples' : info_str}]

with open(r'DataSet/intents.yml', 'a') as file:
    documents = yaml.dump(dict_file, file, sort_keys=False)
```

Figure 6 31 Intent writer

6.6 Knowledge graph and chatbot intent update module.

6.6.1Knowledge update scheduler

The knowledge scheduler will run on every Monday which will update the dynamic database. It will take the keyword list and run all the above-mentioned modules automatically according to the keyword list. Figure 6.32 will display the illustration of the implementation where it takes the keyword list and iterate and run it on every Monday.

```
    file1 = open('Dataset/keyword.txt', 'r')
    Lines = file1.readlines()

    count = 0
    # Strips the newline character
    for line in Lines:
        print(line)
        ys.youtubeScraper(line,False)

schedule.every().monday.do(job)


while True:
    schedule.run_pending()
    time.sleep(1)
```

Figure 6 32 Weight list

6.7 Summary

This chapter presented an overall detailed discussion on the implementation of this project. The next chapter will provide a detailed discussion on the results of the developed system.

7.0 EVALUATION

7.1 Introduction

Chapter 7 presents the evaluation. This provides the overall idea of the evaluation part by providing results of each section and also generating a simple chat using the outputs of the research project.

7.2 Results

```
keyword node created:gallefort
Keyword successfully written on keyword.txt
Fetched URL: https://www.youtube.com/watch?v=c1EjA4wJiZE
Fetched URL: https://www.youtube.com/watch?v=RHUbnsyQtHI
Fetched URL: https://www.youtube.com/watch?v=A0ZA8r9NcSA
Fetched URL: https://www.youtube.com/watch?v=u39jG_2XJVg
Fetched URL: https://www.youtube.com/watch?v=lWC6Z8HM2CM
Fetched URL: https://www.youtube.com/watch?v=0a50zk5XsOs
Fetched URL: https://www.youtube.com/watch?v=q-bk6cg8LXU
Fetched URL: https://www.youtube.com/watch?v=EkXadotx40Y
Fetched URL: https://www.youtube.com/watch?v=sRel3XmFve4
Fetched URL: https://www.youtube.com/watch?v=0indyQt7Uac
```

Figure 7 1 List of URL's

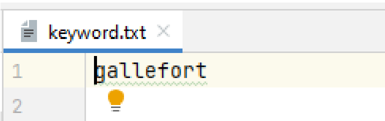


Figure 7 2 Keyword Text File.

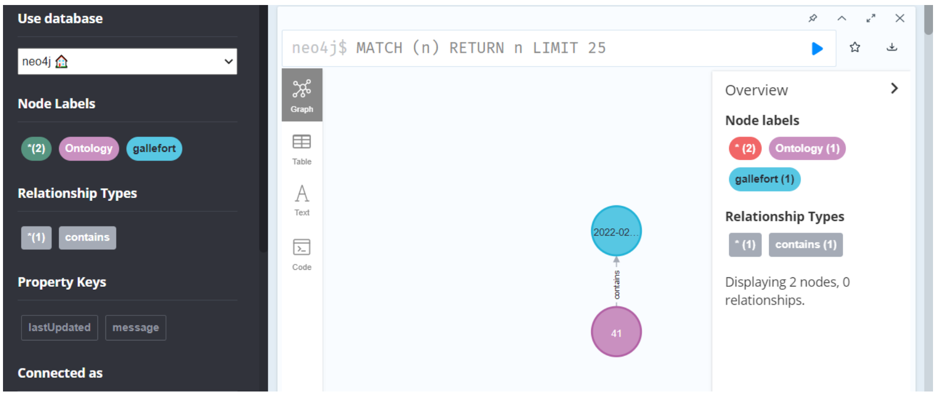


Figure 7 3 Initial keyword on graph

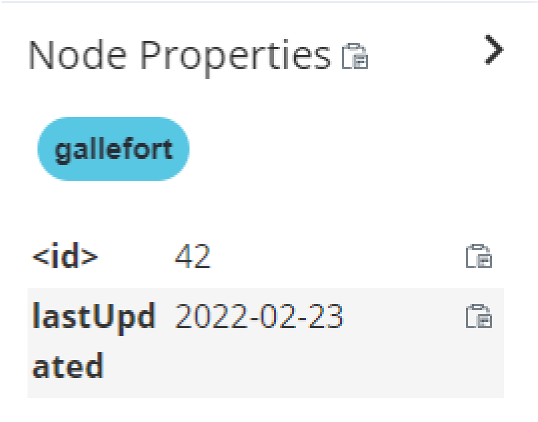


Figure 7 4 Initial node

Figures 7.1, 7.2, 7.3 and 7.4 displays when the keyword is given, how the keyword will be added to a keyword file for scheduler reference and how the keyword details are added to the graph. A list of URLs is printed which are retrieved URLs, which matches the keyword given from the user.

As shown in the below Figure 7.5 last updated date and max updated dates will be retrieved. These will be none if the knowledge graph doesn't contain a node with the user specified keyword.

```
<Record a.lastUpdated=None>
Last updated date: None
Max date: None
```

Figure 7 5 Dates

```
<class 'str'>
2014-03-27
<class 'str'>
Video URL: https://www.youtube.com/watch?v=RHUbnsyQtHI, Video created date: 2014-03-27
```

Figure 7 6 URL creation date

In the above Figure 7.6 it will display the created date of the specific video url which was retreived from the youtube data api.

```
.mp3 file downloaded: Audio/out1.mp3
[youtube] RHUbnsyQtHI: Downloading webpage
[download] Destination: Audio\out1.mp3
[download] 100% of 28.34MiB in 06:49
```

Figure 7 7 Audio downloading

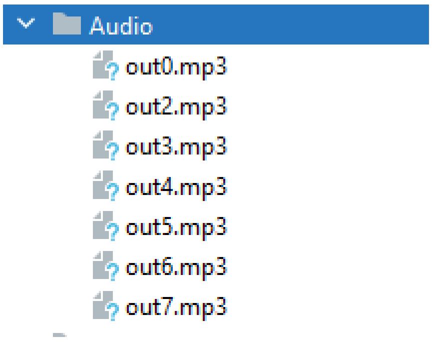


Figure 7 8 Downloaded mp3 files

```
[ffmpeg] Correcting container in "Audio\out1.mp3"
[ffmpeg] Post-process file Audio\out1.mp3 exists, skipping
.mp3 converted to .wav: 1.wav
```

Figure 7 9 .mp3 to .wav

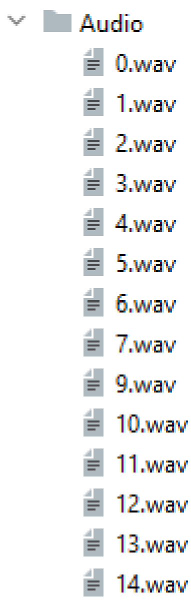


Figure 7 10 List of .wav files

Whem the video created date is greater than the last updated date the url will be sent to the video downloader and a list of .wav files are downloaded as ellobarated in Figure 7.7 and 7.8, it will be converted into .wav as ellobarated in Figures 7.9 and 7.10.

```
Audio-chunks
    chunk01.wav
    chunk11.wav
    chunk12.wav
    chunk13.wav
    chunk14.wav
    chunk15.wav
    chunk16.wav
    chunk17.wav
    chunk18.wav
    chunk19.wav
    chunk110.wav
    chunk111.wav
    chunk112.wav
    chunk113.wav
    chunk114.wav
```

Figure 7 11 List of chunks

```
Audio file chunked: Audio-chunks\chunk11.wav
Error:
Audio file chunked: Audio-chunks\chunk12.wav
Error:
Audio file chunked: Audio-chunks\chunk13.wav
Audio chunk text: August 1755 the first baptism took place at the tort reform church in the golf 4th
Detected language: en
```

Figure 7 12 Audio chunk text

```
Text
    1.txt
    4.txt
    5.txt
    6.txt
    10.txt
    11.txt
    12.txt
    13.txt
    14.txt
```

Figure 7 13 List of text files

```
1.txt
August 1755 the first baptism took place at the tort reform church in the golf 4th and since then him a
On the orders of goa portuguese viceroy francisco de almeida his son was trailing and arab ship in the indian oce
1 oak is known as galo by the portuguese and it was after this incident the portuguese invincible arms cock on th
Something worth fighting for.
On the 13th of march 1640.
The dutch east india company of v o c.
Write the galle fort.
After defeating the portuguese.
Use the most advanced technology and strengthen the forth impeccably.
On the 23rd of february 1796.
Hero of the portuguese ended at the golf 4 things as well as the british try to capture this significant place bu
Sri lanka played a pivotal role during mediaeval i am i became a meeting an exchange point especially for merchan
Prithvi and events south india.
When the portuguese arrived in 15 of 5 to become harder sri lanka history this by the dutch in the british invasi
The ways of life which also inculcated in a communities that exist even to this day in 1988 unesco listed thrivin
Mainly because of its historical significance.
Let me show you why is it plays continues to attract thousands of people from across the globe and harbour in goa
```

Figure 7 14 Audio converted text

Figure 7.11 ,7.12 ,7.13 and 7.14 ellobarates how the audio is chunked to separate chunks and extracted texts stored to the text files after checking the language.

```
Final predicate with the longest length: took place at
Extracted subject: the first baptism
Extracted predicate: took place at
Extracted object:  the tort reform church
the first baptism 0took place at  the tort reform church
```

Figure 7 15 Extracted triples

```
weight assigned for feature 5: 0.43
August 1755 the first baptism
test.txt
0
  SPACE
the DET
tort NOUN
reform NOUN
church NOUN
weight assigned for feature 11
the DET
first ADJ
baptism NOUN
weight assigned for feature 12: 0
weight assigned for feature 13: -0.3
August PROPN
1755 NUM
weight assigned for feature 15: 0.01
weight assigned for feature 16: 0
weight list for the extracted text:
[0, 0, 0, 0, 0.43, 0, 0, 0, 0, 0, 0, 0, -0.3, 0, 0.01, 0]
```

Figure 7 16 List of weights assigned

Figures 7.15 and 7.16 elaborates the output of the text extraction with subject,predicate and the object and the weights assigned for the features.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11	f12	f13	f14	f15	f16	Correctness
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0.43	0	0.39	0	0	0	0	0	-0.3	0	0.01	0	1
0	0	0	0	0.43	0	0.39	0	0	0.21	0	0	-0.3	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0.43	0	0	0	0	0	0	0	-0.3	0	0	0	0
0	0	0	0	0.43	0	0	0	0	0.21	0.16	0.01	-0.3	0	0	0	0
0	0	0	0	0.43	0	0	0	0	0	0.16	0.01	-0.3	0	0.01	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0.43	0	0	0	0	0.21	0.16	0	-0.3	0	0	0	1
0	0	0.49	0	0.43	0	0	0	0	0.21	0	0.01	-0.3	0	0	0	1
0	0	0	0	0.43	0	0	0	0	0	0.16	0	-0.3	0	0	-0.81	0
0	0	0	0.46	0.43	0	0	0	0	0.21	0	0.01	-0.3	0	0	0	0
0	0.5	0	0	0.43	0	0	0	0	0	0	0	-0.3	0	0.01	0	0

Figure 7 17 Feature dataset

Figure 7.17 displays a snapshot of the generated dataset from the web text data extraction for the training purpose of the model.

```
      f1   f2   f3    f4    f5    f6   f7  ...   f10  f11   f12  f13  f14   f15  f16
90   0.0  0.0  0.0  0.00  0.43  0.00  0.0  ...  0.21  0.0  0.01 -0.3  0.0  0.00  0.0
254  0.0  0.0  0.0  0.00  0.00  0.00  0.0  ...  0.00  0.0  0.00  0.0  0.0  0.00  0.0
283  0.0  0.0  0.0  0.46  0.43  0.00  0.0  ...  0.00  0.0  0.00 -0.3  0.0  0.01  0.0
445  0.0  0.5  0.0  0.00  0.00  0.00  0.0  ...  0.00  0.0  0.00 -0.3  0.0  0.00  0.0
461  0.0  0.0  0.0  0.46  0.43  0.43  0.0  ...  0.00  0.0  0.00 -0.3  0.0  0.00  0.0

[5 rows x 16 columns]
[1. 0. 1. 0. 1. 1. 1. 0. 0. 1. 0. 0. 1. 1. 0. 0. 1. 1. 1. 1. 1. 1. 1. 0.
 1. 1. 1. 1. 1. 0. 1. 0. 0. 1. 1. 1. 1. 1. 1. 1. 0. 1. 1. 0. 1. 0. 0. 1.
 1. 1. 0. 1. 0. 1. 0. 1. 1. 0. 1. 1. 1. 0. 0. 1. 0. 1. 0. 0. 1. 0. 0. 1.
 1. 0. 0. 0. 1. 0. 1. 1. 0. 0. 1. 0. 0. 1. 1. 0. 1. 1. 1. 0. 0. 0. 1. 1.
 1. 1. 1. 1. 0. 1. 1. 1. 1. 1. 1. 0. 1. 1. 0. 1. 1. 0. 1. 1. 1. 0. 0. 1.
 1. 1. 1. 1. 1.]
1.0
```

Figure 7 18 Logistic model training

Figure 7.18 ellobarates the output for the weightlist assigned and as the extraction is correct an output of 1 is given.

```
Writing intents...
who is taking place at in the first baptism?
created intent example
who is taking place at in the first baptism?
what is taking place at in the first baptism?
created intent example
what is taking place at in the first baptism?
when is took place at in the first baptism
created intent example
when is took place at in the first baptism
where is took place at in the first baptism
created intent example
where is took place at in the first baptism
What is taking place at in the first baptism?
created intent example
What is taking place at in the first baptism?
how is took place at in the first baptism
created intent example
how is took place at in the first baptism
```

Figure 7 19 Intents

```
    - sad
    - very sad
    - unhappy
    - not good
    - not very good
    - extremly sad
    - so saad
    - so sad

- intent: bot_challenge
  examples: |
    - are you a bot?
    - are you a human?
    - am I talking to a bot?
    - am I talking to a human?

- intent: the_first baptism_took_place_at
  examples: |
    - who is taking place at in the first baptism?
    - what is taking place at in the first baptism?
    - when is took place at in the first baptism
    - where is took place at in the first baptism
    - What is taking place at in the first baptism?
    - how is took place at in the first baptism
```

Figure 7 20 nlu file with updated intents and examples if no semantic similarity with available intents

```
0.642422114518568
0.8799302148400052
Sentence with maximum similarity gallefort_the_first_baptism_took_place_at
Similarity value: 0.8799302148400052
Similarity value is 0.8799302148400052 Threshold value is 0.8
```

Figure 7 21 Similarity checker between available intents and created intent

```
- intent: gallefort_the_first_baptism_took_place_at
  examples: |
    - who is taking place at in the first baptism?
    - what is taking place at in the first baptism?
    - when is took place at in the first baptism
    - where is took place at in the first baptism
    - What is taking place at in the first baptism?
    - how is took place at in the first baptism
    - what is baptism in first held?
    - when is a baptism in first held'
```

Figure 7 22 nlu file with updated examples if semantic similarity with available intents

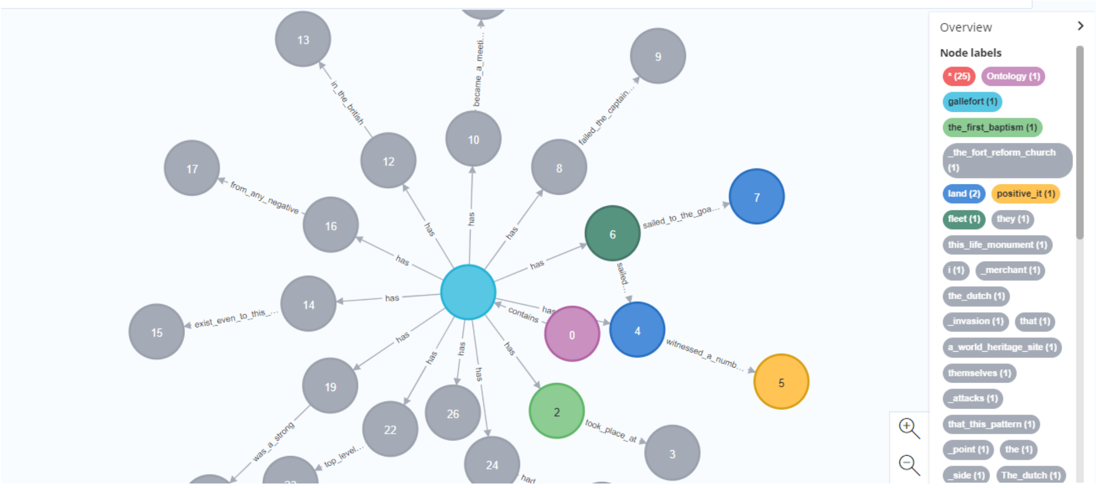


Figure 7 23 Knowledge graph

Above figures, Figure 7.19 , Figure 7.20 ,Figure 7.21 and Figure 7.22 displays the created intents and there examples from the extracted text. Figure 7.23 represents the generated knowladge graph.

```
['keyword', 'subject', 'predicate', 'object', 'Outcome']
    keyword            subject  ...                   object  Outcome
0   keyword            subject  ...                   object  Outcome
1 gallefort  the_first_baptism  ...  _the_fort_reform_church      Yes
2 gallefort  the_first_baptism  ...  _the_fort_reform_church      Yes
3 gallefort  the_first_baptism  ...  _the_fort_reform_church      Yes
4 gallefort  the_first_baptism  ...  _the_fort_reform_church      Yes
```

Figure 7 24 Categorical data

```
     keyword_1  keyword_2  subject_1  ...  object_1  object_2  object_3
109          1          0          1  ...         1         0         0
261          1          0          1  ...         1         0         0
791          1          0          0  ...         0         1         0
101          1          0          1  ...         1         0         0
648          1          0          1  ...         1         0         0

[5 rows x 11 columns]
     keyword_1  keyword_2  subject_1  ...  object_1  object_2  object_3
575          1          0          1  ...         1         0         0
693          1          0          1  ...         1         0         0
79           1          0          1  ...         1         0         0
577          1          0          1  ...         1         0         0
667          1          0          1  ...         1         0         0
```

Figure 7 25 Numerical data

```
 'Yes' 'Yes' 'Yes' 'Yes' 'Yes' 'Yes' 'Yes' 'Yes' 'Yes' 'Yes' 'Yes' 'Ye
 'Yes' 'Yes' 'Yes' 'Yes' 'Yes' 'Yes' 'Yes' 'Yes' 'Yes' 'Yes' 'Yes' 'Ye
 'Yes' 'Yes' 'Yes' 'Yes' 'No' 'Yes' 'No' 'Yes' 'Yes' 'Yes' 'Yes' 'Yes'
 'Yes' 'Yes' 'Yes' 'Yes' 'Yes' 'Yes' 'Yes' 'No' 'Yes' 'Yes' 'Yes' 'Yes
 'Yes' 'Yes' 'Yes' 'Yes' 'Yes' 'Yes' 'Yes' 'Yes' 'Yes' 'Yes' 'Yes' 'Ye
 'Yes' 'Yes' 'Yes' 'Yes' 'Yes' 'Yes' 'Yes' 'Yes' 'Yes' 'No' 'Yes' 'Yes
 'Yes' 'Yes' 'Yes' 'Yes' 'Yes' 'Yes' 'Yes' 'Yes' 'Yes' 'Yes' 'Yes' 'Ye
y value
Yes
```

Figure 7 26 Training data and predicted value

Figures 7.24 , 7.25 and 7.26 displays the final classification of the Naïve Bayers classifier as it is a good extraction according to the user's point of view

```
Bot loaded. Type a message and press enter (use '/stop' to exit):
Your input ->  hi
Hey! How are you?
Your input ->  where is the firstbaptism took place
the fort reform church
Your input ->  ok
```

Figure 7 27 Sample chat

The above, Figure 7.27 illustrates the simple chat produced by a chatbot using the intents and the graph generated.

7.2 Evaluation using matrices

This chapter presented an overall detailed discussion on the evaluation of this research. A logistic regression model is used to classify the correct classifications and the below Figure 7.28 which displays the matrices of the model.

```
Logistic Regression Accuracy: 0.664
Logistic Regression Precision: 0.6375
Logistic Regression Recall: 0.796875
```

Figure 7 28 Logistic regression matrices

The below Figure 7.29 displays the matrices of the Naïve Bayes classifier which is used for node classification.

```
Model accuracy score: 0.9667
Classification accuracy : 0.9667
Precision : 0.6000
]n
Recall or Sensitivity : 1.0000
              precision    recall  f1-score   support

          No       1.00      0.60      0.75        20
         Yes       0.96      1.00      0.98       220

    accuracy                           0.97       240
   macro avg       0.98      0.80      0.87       240
weighted avg       0.97      0.97      0.96       240
```

Figure 7 29 Naive bayes matrices

7.7 Summary

This chapter presented an overall detailed discussion on the evaluation of this research. The next chapter provides a detailed discussion on the conclusion of this project and the future work planned.

Chapter 8

8.0 Conclusion and Further Work

8.1 Introduction

The evaluation of the knowledge graph, chatbot intent generation and their results were presented in Chapter 7. In this chapter the overall achievements of this research will be presented. It will explain the overall objectives and the achievements of the objectives.

The problems encountered in this research and the limitations are mentioned in this chapter. The next steps of the research are also discussed.

8.2 Conclusion

The goal of our world is to reach out audience with minimal number of human interactions through a chatbot. Chatbot communication is more effective than human interaction. Therefore, as a helping hand for a good chatbot communication a knowledge graph generation model will be modified with extending it by obtaining information from video content. At present volume of video content creation is surpassing text content creation with the advancement of technology and as an effective mode of communication therefore this method will create more value in the future.

This research thesis covers the overall creation of the chatbot ontology creation and intent creation from video content. This knowledge extracting mechanism will also extract nlp intents for the chatbot communication. This research has the capability to dynamically extract data and also has the capability to update the knowledge graph and intents once a week when new video contents are uploaded. It also takes the history of user and improves the extracted knowledge graph and intents.

In this thesis the existing approaches for knowledge graph generation for chatbots were critically reviewed. An in-depth study was conducted on the technologies that are used to generate chatbots and knowledge graphs. An in-depth discussion of information on the design and development algorithms to generate the knowledge graph and chatbot intents was conducted. This evaluates the development approach and simulates and compares and contrast with the existing approach.

The approach in this thesis doesn’t support for different languages as it has the ability to only extract knowledge from English language videos. As a future work a feature will be introduced to support extracting knowledge from different languages. This doesn’t support nested question nodes or continuous chat with the chatbot and the nested node extraction will be done as future work.

8.9 Summary

Chapter 8 concludes the thesis and provides a clear understanding on the objectives, achievements and the future work of the research.

References

[1] A SURVEY To Chatbot System With Knowledge Base Database By Using Artificial Intelligence & Expert Systems".

[2] A.Ait-Mlouk and L. Jiang, "KBot: A Knowledge Graph Based ChatBot for Natural Language Understanding Over Linked Data," IEEE Access, vol. 8, pp. 149220–149230, 2020, doi: 10.1109/ACCESS.2020.3016142.

[3] About Watson," Jul. 29, 2021. https://www.ibm.com/watson/about (accessed Mar. 02, 2022).

[4] Adamopoulou and L. Moussiades, "An Overview of Chatbot Technology," in Artificial Intelligence Applications and Innovations, vol. 584, I. Maglogiannis, L. Iliadis, and E. Pimenidis, Eds. Cham: Springer International Publishing, 2020, pp. 373–383. doi: 10.1007/978-3-030-49186-4_31.

[5] AI & Graph Technology: What Are Knowledge Graphs? - Neo4j Graph Data Platform." https://neo4j.com/blog/ai-graph-technology-knowledge-graphs/ (accessed Mar. 03, 2022).

[6] Alan Turing | Biography, Facts, Computer, Machine, Education, & Death | Britannica." https://www.britannica.com/biography/Alan-Turing (accessed Mar. 02, 2022).

[7] An and Y. B. Park, "Methodology for Automatic Ontology Generation Using Database Schema Information," Mobile Information Systems, vol. 2018, pp. 1–13, 2018, doi: 10.1155/2018/1359174.

[8] Avram, "Empirical Study on Knowledge Based Systems," vol. 8, no. 1, p. 11, 2005.

[9] Cahn, "CHATBOT: Architecture, Design, & Development," p. 46.

[10] Caldarini, S. Jaf, and K. McGarry, “A Literature Survey of Recent Advances in Chatbots,” Information, vol. 13, no. 1, p. 41, Jan. 2022, doi: 10.3390/info13010041.

[11] Carlander-Reuterfelt, A. Carrera, C. A. Iglesias, O. Araque, J. F. S. Sanchez Rada, and S. Munoz, “JAICOB: A Data Science Chatbot,” IEEE Access, vol. 8, pp. 180672–180680, 2020, doi: 10.1109/ACCESS.2020.3024795.

[12] Curry and D. J. D. O’Shea, “The implementation of a storytelling Chatbot,” p. 8.

[13] D. Martinez, M. Taboada, and J. Mira, “Knowledge Base Development,” in Knowledge-Based Intelligent Information and Engineering Systems, vol. 2774, V. Palade, R. J. Howlett, and L. Jain, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 1373–1380. doi: 10.1007/978-3-540-45226-3_186.

[14] E. Adamopoulou and L. Moussiades, “Chatbots: History, technology, and applications,” Machine Learning with Applications, vol. 2, p. 100006, Dec. 2020, doi: 10.1016/j.mlwa.2020.100006.

[15] Ferrucci et al., “Building Watson: An Overview of the DeepQA Project,” AIMag, vol. 31, no. 3, p. 59, Jul. 2010, doi: 10.1609/aimag.v31i3.2303.

[16] G. Weikum, L. Dong, S. Razniewski, and F. Suchanek, “Machine Knowledge: Creation and Curation of Comprehensive Knowledge Bases,” arXiv:2009.11564 [cs], Mar. 2021, Accessed: Mar. 03, 2022. [Online]. Available: http://arxiv.org/abs/2009.11564

[17]HeidiSteen, “Azure Cognitive Search Service REST.” https://docs.microsoft.com/en-us/rest/api/searchservice/ (accessed Mar. 02, 2022).

[18] International Journal of Scientific Research in Computer Science, Engineering and Information Technology," vol. 4, no. 6, p. 7, 2018.

[19] J. Schuurmans, F. Frasincar, and E. Cambria, "Intent Classification for Dialogue Utterances," IEEE Intell. Syst., vol. 35, no. 1, pp. 82–88, Jan. 2020, doi: 10.1109/MIS.2019.2954966.

[20] J. Weizenbaum, "ELIZA—a computer program for the study of natural language communication between man and machine," Commun. ACM, vol. 9, no. 1, pp. 36–45, Jan. 1966, doi: 10.1145/365153.365168.

[21] Jurafsky, D. and Martin, J., 2020. Speech and Language Processing. 3rd ed.

[22] K. Fitzpatrick, A. Darcy, and M. Vierhile, "Delivering Cognitive Behavior Therapy to Young Adults With Symptoms of Depression and Anxiety Using a Fully Automated Conversational Agent (Woebot): A Randomized Controlled Trial," JMIR Ment Health, vol. 4, no. 2, p. e19, Jun. 2017, doi: 10.2196/mental.7785.

[23] Kudumula, "The Architecture of Bot Framework," IJCTT, vol. 68, no. 1, pp. 1–4, Jan. 2020, doi: 10.14445/22312803/IJCTT-V68I1P101.

[24] M. Rahman, A. A. Mamun, and A. Islam, "Programming challenges of chatbot: Current and future prospective," in 2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC), Dhaka, Dec. 2017, pp. 75–78. doi: 10.1109/R10-HTC.2017.8288910.

[25] M. Turing, "I.—COMPUTING MACHINERY AND INTELLIGENCE," Mind, vol. LIX, no. 236, pp. 433–460, Oct. 1950, doi: 10.1093/mind/LIX.236.433.

[26] Melina, E. K. Putra, W. Witanti, Sukrido, and V. A. Kusumaningtyas, “Design and Implementation of Multi Knowledge Base Expert System Using the SQL Inference Mechanism for Herbal Medicine,” J. Phys.: Conf. Ser., vol. 1477, p. 022007, Mar. 2020, doi: 10.1088/1742-6596/1477/2/022007.

[27] Mgr. T. Zem čík, “A Brief History of Chatbots,” dtcse, no. aicae, Oct. 2019, doi: 10.12783/dtcse/aicae2019/31439.

[28] Microsoft Bot Framework.” https://dev.botframework.com/ (accessed Mar. 02, 2022).

[29] Mirzal, “Design and Implementation of a Simple Web Search Engine,” International Journal of Multimedia and Ubiquitous Engineering, vol. 7, no. 1, p. 9, 2012.

[30] Much more than a chatbot: China’s Xiaoice mixes AI with emotions and wins over millions of fans - Microsoft Stories Asia.” https://news.microsoft.com/apac/features/much-more-than-a-chatbot-chinas-xiaoice-mixes-ai-with-emotions-and-wins-over-millions-of-fans/ (accessed Mar. 03, 2022).

[31] Patil, M. K, N. R. A, and N. R, “Comparative study of cloud platforms to develop a Chatbot,” IJET, vol. 6, no. 3, p. 57, Jun. 2017, doi: 10.14419/ijet.v6i3.7628.

[32] Q. Yin, S. Co, and M. Satar, “ENGLISH AS A FOREIGN LANGUAGE LEARNER INTERACTIONS WITH CHATBOTS: NEGOTIATION FOR MEANING,” p. 22, 2020.

[33] R. Lee, “Chatbot ESL Robot - English Tutor Mike, Ron C. Lee | Virtual Assistant ESL Robot - English Tutor Mike | Virtual agent ESL Robot - English Tutor Mike | Chat bot ESL Robot - English Tutor Mike | Conversational agent ESL Robot - English Tutor Mike | (10775),” chatbots.org. https://www.chatbots.org/chatbot/mike2/ (accessed Mar. 02, 2022).

[34] S. A. and Dr. John, “Survey on Chatbot Design Techniques in Speech Conversation Systems,” ijacsa, vol. 6, no. 7, 2015, doi: 10.14569/IJACSA.2015.060712.

[35] S. A. Salloum, M. Al-Emran, A. A. Monem, and K. Shaalan, “Using Text Mining Techniques for Extracting Information from Research Articles,” in Intelligent Natural Language Processing: Trends and Applications, vol. 740, K. Shaalan, A. E. Hassanien, and F. Tolba, Eds. Cham: Springer International Publishing, 2018, pp. 373–397. doi: 10.1007/978-3-319-67056-0_18.

[36] S. A. Thorat and V. Jadhav, “A Review on Implementation Issues of Rule-based Chatbot Systems,” SSRN Journal, 2020, doi: 10.2139/ssrn.3567047.

[37] S. Elnagar, V. Yoon, and M. A. Thomas, “An Automatic Ontology Generation Framework with An Organizational Perspective,” p. 10.

[38] S. Nagdewani and A. Jain, “A REVIEW ON METHODS FOR SPEECH-TO-TEXT AND TEXT-TO-SPEECH CONVERSION,” vol. 07, no. 05, p. 6, 2020.

[39] S. Yoo and O. Jeong, “Auto-Growing Knowledge Graph-Based Intelligent Chatbot Using BERT.” ICIC International 学会, 2020. Accessed: Mar. 03, 2022. [Online]. Available: https://doi.org/10.24507/icicel.14.01.67

[40] Salhi, K. E. Guemmat, M. Qbadou, and K. Mansouri, “Towards developing a pocket therapist: an intelligent adaptive psychological support chatbot against mental health disorders in a pandemic situation,” IJEECS, vol. 23, no. 2, p. 1200, Aug. 2021, doi: 10.11591/ijeecs.v23.i2.pp1200-1211.

[41] T. Lampoltshammer and S. Wiegand, “Improving the Computational Performance of Ontology-Based Classification Using Graph Databases,” Remote Sensing, vol. 7, no. 7, pp. 9473–9491, Jul. 2015, doi: 10.3390/rs70709473.

[42] T. Um, T. Kim, and N. Chung, “How does an Intelligence Chatbot Affect Customers Compared with Self-Service Technology for Sustainable Services?,” Sustainability, vol. 12, no. 12, p. 5119, Jun. 2020, doi: 10.3390/su12125119.

[43] The History Of Chatbots - From ELIZA to Alexa,” AI Chatbots software from Onlim, Oct. 12, 2017. https://onlim.com/en/the-history-of-chatbots/ (accessed Mar. 02, 2022).

[44] Types of Chatbot Technology. This article is an excerpt of… | by Artificial Solutions | Voice Tech Podcast | Medium.” https://medium.com/voice-tech-podcast/types-of-chatbot-technology-72d095df2540 (accessed Mar. 03, 2022).

[45] W. Maroengsit, T. Piyakulpinyo, K. Phonyiam, S. Pongnumkul, P. Chaovalit, and T. Theeramunkong, “A Survey on Evaluation Methods for Chatbots,” in Proceedings of the 2019 7th International Conference on Information and Education Technology - ICIET 2019, Aizu-Wakamatsu, Japan, 2019, pp. 111–119. doi: 10.1145/3323771.3323824.

[46] W. Shalaby, A. Arantes, T. GonzalezDiaz, and C. Gupta, “Building chatbots from large scale domain-specific knowledge bases: challenges and opportunities,” in 2020 IEEE International Conference on Prognostics and Health Management (ICPHM), Detroit, MI, USA, Jun. 2020, pp. 1–8. doi: 10.1109/ICPHM49022.2020.9187036.

[47] What is the Turing Test?,” SearchEnterpriseAI. https://www.techtarget.com/searchenterpriseai/definition/Turing-test (accessed Mar. 02, 2022).

[48] X. Verduijn, “Chatbot Albert One, Robby Garner | Virtual Assistant Albert One | Virtual agent Albert One | Chat bot Albert One | Conversational agent Albert One | (4537),” chatbots.org. https://www.chatbots.org/chatbot/albert_one/ (accessed Mar. 02, 2022).

[49] Zhou, J. Gao, D. Li, and H.-Y. Shum, “The Design and Implementation of XiaoIce, an Empathetic Social Chatbot,” arXiv:1812.08989 [cs], Sep. 2019, Accessed: Mar. 03, 2022. [Online]. Available: http://arxiv.org/abs/1812.08982