Finding reliable resources and Chatting with Mira while considering emotions when the scenario is unscripted

by

Mohamad Ali Gharaat

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

University of Alberta

Abstract

The primary wellspring of information, the Internet, abounds with misinformation, particularly in the domains of mental health and psychology. This also affects the reliability and truth of responses of chatbots counting on unverified data.

To address this concern, the MIRA project started to create a reliable source of information and develop a chatbot that can deliver these verified resources of information. By respecting privacy, anonymity of user data, MIRA seeks to support individuals looking for a safe environment to find information and assistance in their mental health journey.

The MIRA Project consists of the MIRA resource library and the MIRA chatbot. In this thesis we develop the MIRA resource library in which our domain experts define, edit, and evaluate resources. This also includes a resource search functionality that allows the MIRA chatbot to search efficiently among the verified resources.

The MIRA chatbot employs scripted questions to detect user intents and entities. In this thesis we add Chatty MIRA for producing empathetic replies for scenarios in which the user is looking for just a conversation with the MIRA chatbot. Chatty MIRA facilitates a rule-based text generator and an emotion detection component, proficient in identifying emotions and producing empathetic replies. As users engage in casual conversations with Chatty MIRA, Chatty MIRA can simultaneously attempt to search for related resources based on detected intents and entities. This allows for a more engaging experience

for the end-user, where MIRA can provide empathetic responses while also offering relevant resources to support the users with relevant resources it can find.

We conduct a human evaluation to assess the superiority of Chatty MIRA over its rule-based and Large Language Model (LLM) competitors. As well as determine the gap between Chatty MIRA and advanced LLMs (i.e. GPT-3.5).

Preface

Ethics approval was granted on August 12, 2021, by the University of Alberta Health Research Board (case Pro00109148) and on April 21, 2022, by the Nova Scotia Health Authority Research Ethics Board (case 1027474). All data and computer code will be password-protected and stored on a secure server at the University of Alberta in Canada.

There are two papers that are related to the MIRA project:

- Noble J, Zamani A, Gharaat M, Merrick D, Maeda N, Lambe Foster A, Nikolaidis I, Goud R, Stroulia E, Agyapong V, Greenshaw A, Lambert S, Gallson D, Porter K, Turner D, Zaiane O. Developing, Implementing, and Evaluating an Artificial Intelligence–Guided Mental Health Resource Navigation Chatbot for Health Care Workers and Their Families During and Following the COVID-19 Pandemic: Protocol for a Cross-sectional Study [39]

- Zamani A, Reeson M, Marshall, T, et. al. "Intent and Entity Detection with Data Augmentation for a Mental Health Virtual Assistant Chatbot, 2023 Annual Conference on Intelligent Virtual Agents (ACM IVA 2023). Accepted and article to be published following conference.

To my lovely wife, Fatemeh

I found the meaning of my life with you.

Follow valuable goals, you are what you are looking for!

– Jalāl al-Dīn Muḥammad Rūmī, Persian Poet.

Acknowledgements

I want to express my deep appreciation for the invaluable support and guidance provided by Prof. Osmar R. Zaiane, my supervisor. I also extend my gratitude to Prof. Eleni Stroulia for her guidance and assistance. I thank the Mood Disorder Society of Canada (MDSC) for believing in this team and helping us gaining real world experience. Thank you Dave Gallson, and Ken Porter for giving us the freedom to do our research on this project. I do appreciate the help from Dr. Jasmine Nobel, who managed well the whole project and brought us to the point we are with her skills, hard work, and knowledge. I want to thank my friend and colleague, Ali Zamani, who designed and implemented the Mira core. I also thank other colleges and professors helping us in the Mira chatbot project. Without their contribution, constructive opinion, Mira would not be successful.

Contents

List of Tables

List of Figures

Glossary

AI
: Artificial Intelligence

API
: Application Programming Interface

BERT
: Bidirectional Encoder Representations from Transformers

CA
: Conversational Agent

CAMI
: Coaching Assistant for Medical Information

CNN
: Convolutional Neural Networks

CoLA
: Corpus of Linguistic Acceptability

CRUD
: Create, Read, Update, Delete

DEM
: Discrete Emotion Model

DiEM
: Dimensional Emotion Model

DIET
: Dual Intent and Entity Transformer

ELIZA
: ELIZA Chatbot

ELMo
: Embeddings from Language Model

EM
: Emotion Mining

IR
: Information Retrieval

ISEAR
: International Survey on Emotion Antecedents and Reactions

LLM
: Large Language Model

LM
: Language Model

LSTM
: Long Short-term Memory Networks

MDSC
: Mood Disorder Society of Canada

MELD
: Multimodal Emotion Lines Dataset

MIRA
: Mental Health Intelligent Information Resource Assistance

NLP
: Natural Language Processing

NLTK
: Natural Language Toolkit

PTSD
: Post-Traumatic Stress Disorder

QA
: Question and Answering

SVM
: Support Vector Machines

TF-IDF
: Term Frequency-Inverse Document Frequency

VA
: Virtual Assistants

Chapter 1

Introduction

In recent times, there has been a noticeable rise in the prevalence of mental health concerns [29]. Comparing fall 2020 with spring 2021, the frequency of adults experiencing major depressive disorder has increased by 4%[1] in Canada. Numerous services and programs for addressing mental health concerns are available throughout Canada. However, locating and trusting them might be challenging due to a substantial volume of unverified information.

Chatbots have the capacity to stimulate clients to engage in more dialogue through inquiry, identify user requirements, and furnish users with verified resources. They are accessible for consultation at any hour and do not pass judgment on their users.

The Mood Disorder Society of Canada (MDSC) has commissioned our research team to build a chatbot that can provide verified resources. In the context of this project, a resource encompasses any entity capable of disseminating valuable information. Resources may include literature, online platforms and websites, contact details, or a physical clinic. The Mental Health Intelligent Information Resource Assistance (MIRA) project requirements were defined in a way that we could launch the system in one year. This project includes the MIRA chatbot and the MIRA resource library.

Our research team had developed the Coaching Assistant for Medical Information (CAMI) chatbot[57]. CAMI is an information retrieval chatbot that uses its database of resources about neurodevelopmental disorders. Instead of

[1] `https://www150.statcan.gc.ca/n1/daily-quotidien/210927/dq210927a-eng.htm`

starting the MIRA project from scratch, it was suggested to clone the CAMI project or a part of it. However, CAMI exhibits certain limitations that we aimed to rectify in the MIRA project. This involved the development of new features and the exploration of non-trivial challenges.

- Resource formats: CAMI and MIRA both are chatbots that recommend resources to their end-users, but in CAMI, the resources are only web pages, whereas in MIRA, resources can encompass any services or programs offered by an organization that could be beneficial for mental health patients, spanning from clinics to books.

- Semantic-based search: CAMI has keyword-based search functionality to search over resources while we needed semantic-based search functionality. Semantic-based search takes into account the meaning or concept behind a query and retrieves and ranks resources based on their similarity to the query, even if the keywords from the query are not explicitly present in those resources. For instance, "ate" and "eat" have the same meaning, but a keyword matching method cannot identify their similarity.

- Tagging: CAMI does not support resource tagging, while in the MIRA project resources should be tagged with keywords. While tag categories are predefined, the tags themselves can encompass a wide range of terms within that category. In MIRA, we should be able to review tags and their categories for new submitted tags and approve them. For example, location is a tag category and a resource can be tagged with any tag under that tag category. If a resource is available in Canada, it is tagged with the "Canada" tag. If this is the first time we are adding the "Canada" tag to the database, it should get approved in a separate review process to make sure it actually belongs to its tag category.

- Review process: CAMI did not have a review process. However in MIRA, we require at least two reviewers to assess the submitted resources and

grant approval. Tags also are required to be approved if they have not been approved yet.

- Server limitations: For MIRA, we have a specific requirement requested by MDSC. The server on which we must deploy our system is restricted to utilizing only 4GB of RAM and four 1GHz CPU cores.

- Query to tag mapping: The query sent by the MIRA chatbot to the MIRA resource library should be mapped to tags that are available in the MIRA database. Query keywords might contain spelling errors or may not directly match any tags through keyword matching. However, the system should possess the capability to map each query term to an appropriate tag. By mapping query keywords to tags, it can find resources in association with these tags and sort those resources based on their similarity to the query.

- Query relaxation: MIRA is required to always return resources even if there is no exact matches for the input query. It should be able to relax queries and return the most relevant resources it has. Query relaxation reduces the restrictions that the query has over the resources until it can find enough resources for users. For example, if a user is seeking a resource related to depression and is located in Edmonton while there is no resource in Edmonton in the database; the system can check whether there are resources in the database in Alberta (instead of Edmonton). By searching for resources in Alberta instead of Edmonton, the system is relaxing the query.

Instead of using old school expert system CAMI has, We decide to use the Rasa framework[2] which utilizes sentence embedding to classify the intent of the users more accurately as the MIRA chatbot core. Ali Zamani implemented the MIRA chatbot core based on the Rasa framework as a part of his thesis [71]. He designed the core of the MIRA chatbot to enable it to ask scripted

[2]https://rasa.com/

questions and analyze end-user responses in order to detect intent and extract entities from user input.

The MIRA chatbot has been able to detect intents and entities only in a scripted dialogue flow. A series of dialogue flows were defined by experts. MIRA uses the flows to respond and jumps from one state to another based on the user's answers to the questions [71]. But in some scenarios, users just want to chat with the MIRA chatbot. In this thesis, we add this feature of chatting without scripted flows to the MIRA chatbot. We could use rule-based systems like ELIZA Chatbot (ELIZA) but the ELIZA chatbot was developed in 1966 and the wording of the responses is outdated. The language ELIZA has is direct without necessary empathy when responding to users' problems. In fact, ELIZA is lacking any empathy which needs to be addressed in MIRA when talking on issues related to mental health. ELIZA similar to other rule-based chatbots is dependent on text patterns. It detects the topic of the user message by parsing the message with its rules. No matter how close the semantics of the sentence is to a predefined rule in ELIZA. If ELIZA can not parse a sentence with the regular expression of that rule it does not generate the response in that topic.

On the other hand, new approaches based on Large Language Model (LLM) are able to generate desired text by prompt-engineering. Their short come is fabrication (known as hallucination), and generating harmful responses [26]. Which is unacceptable in the domain that MIRA is targeting.

1.1 Approach

The MIRA project is required to have a chatbot and a resource library. we discuss our approach for every

1.1.1 The MIRA Resource Library

In this work, we extend the framework that CAMI resource database was based on to have Create, Read, Update, Delete (CRUD) functionalities over resources from the framework and extend it to align it with our requirements

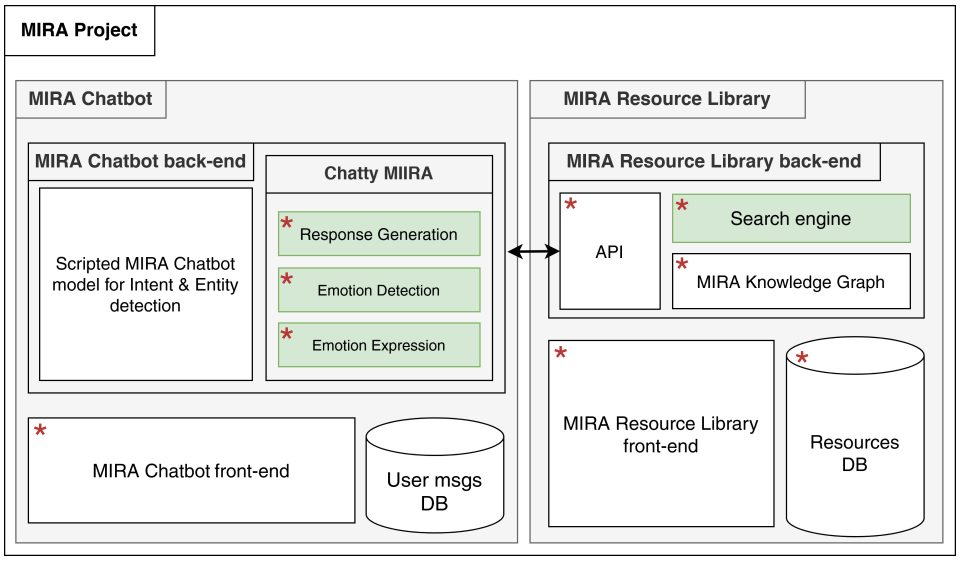

Figure 1.1: The MIRA project subsystems. Green components are components we researched on and evaluated in this thesis and stared components are components we developed for this project.

such as having supporting more types of resources, Tagging system, semantic-based search engine, query relaxation, and resource verification process.

1.1.2 Chatty MIRA

To support conversations when the scenario is unscripted, for example, when end-users just want to chat with the chatbot to feel heard. We needed a solution for unscripted scenarios in between rule-based systems and LLM that is not only empathetic but also can control the output from hallucination. In this thesis we propose Chatty Mira which is an extension of the ELIZA rule-based chatbot. Chatty Mira has all the attributes of ELIZA plus we updated its responses to make it close to what end-users expect from a mental health chatbot these days, by adding a semantic-based rule selection component to detect topics using a hybrid approach (keyword-based and semantic-based) to generate the response. To make chatty Mira emotionally intelligent we need to detect expressed emotion and convey empathy. We embed an emotion classifier into chatty MIRA and added a heuristic approach to generate an empathetic response having the topic and the emotion of the end-user message.

Fig. 1.1 depicts the subsystems of the MIRA project. In this thesis we

develop all the stared components. Our research mainly focuses on the green components namely: Chatty Mira Response generation and emotion detection/expression, and Resource library search engine. In other words, our research focuses on Information Retrieval (IR) in the MIRA resource library search engine, response generation and emotion detection and expression in Chatty MIRA are three main Natural Language Processing (NLP) tasks that we did research on to offer a solution based on our requirements.

1.2 Research Questions

In this research we would like to answer these research questions:

- RQ1) How to design a semantic-based search engine able to search among thousands of resources in an acceptable time for users, on a limited-resource small server? Chapter 3 covers all materials needed to answer this research question.
- RQ2) How can rule-based or AI-based approaches be used to generate responses when scenario is unscripted, on a limited-resource small server? Chapter 6 covers all materials needed to answer this research question.
- RQ3) How to detect emotion expressed by a user in the Chatty MIRA context, on a limited-resource small server? Chapter 4 covers all materials needed to answer this research question.
- RQ4) How responding based on user's emotion can show Chatty MIRA's empathy to users? Chapter 5 covers all materials needed to answer this research question.

1.3 Thesis Contributions

The primary contributions of our work is to extend **the MIRA resource library**, as well as **an emotionally intelligent response generator named Chatty MIRA**:

- the MIRA Resource Library: a website implemented to define, and evaluate resources. This website also is responsible for providing an internal Application Programming Interface (API) for the MIRA chatbot to search semantically among resources.
- Chatty MIRA: An efficient rule-based chatbot, capable of outperforming state-of-the-art Language Models in terms of user engagement, empathy, and contextually relevant response generation.

1.4 Thesis Organization

The rest of this manuscript consist of six chapters, background and related work, the MIRA project and its resource library, emotion detection, emotion expression, response generation, and conclusion and future work.

Chapter 2: Background and Related Work
In this chapter, AI and rule-based chatbots, particularly those designed for mental health applications are explored. Research related to the application of conversational agents in mental health reviewed, taking into account the perspectives of both younger and older users. Additionally, various methods to understand human emotions are discussed, as well as, emotion definitions.

Chapter 3: the MIRA project and its resource library
In this chapter attributes of the MIRA project and its components are discussed. As well as the MIRA resource library which is implemented to provide functionalities such as: CRUD and search. This chapter answers RQ1 and justifies how we designed and implemented a website to store resources and the search engine to rank resources for queries.

Chapter 4: Emotion Detection
In this chapter different machine learning approaches for classification task are compared. Including Naive Bayes, Logistic Regression, Support Vector Machine, Convolutional Neural Networks (CNN), Long Short-term Memory

Networks (LSTM), and Bidirectional Encoder Representations from Transformers (BERT). The BERT language model, which was finetuned on the selected emotions dataset, outperformed all other models with an accuracy of 83.45%. This chapter focuses on providing an answer to RQ3.

Chapter 5: Response Generation

In this chapter AI-based approaches as well as rule-based approaches are investigated for the response generation task in Chatty MIRA. For AI-based approaches we tested various prompts for every candidate Language Model (LM)s. On the other hand we used the original ELIZA as a starting point to develop a rule-based approach. Our evaluation showed that rule-based static response generator can achieve better results, while generating response in less amount of time with the same hardware. In this chapter, we address RQ2. Our final solution is a rule-based text generation method combined with a transformer model for similarity assessment. We enhanced a rule-based system from ELIZA and made it better than Falcon-7b, even though there's still a significant gap between it and state-of-the-art LLMs like GPT-3.5.

Chapter 6: Emotion Expression

This chapter depicts our evaluation results as well as what were the steps we took to align our response in a way that it could convey emotion and answers user's emotion in a proper way. We compared Artificial Intelligence (AI)-based models with rule-based approaches in the evaluation section of this chapter. This chapter answers RQ4 by conducting a human evaluation to compare the output of empathetic Chatty MIRA with the original ELIZA, and other AI-based generative models.

Chapter 7: Conclusion and Future work

In this chapter we reflect on the problems and challenges we had and our approaches to solve them. We also reflect on future plans to improve the MIRA system usability in different ways.

Chapter 2

Background and Related Work

In this chapter we start by defining background concepts such as natural language processing, as well as language modeling and chatbots. Then we review related works around chatbots in mental health and studies on chatbots in mental health from two perspectives. Finally we review related literature about modeling human emotions. We discuss related works of these three tasks in their coresponding chapters 4, 6, 5.

2.1 Natural Language Processing

Since the inception of AI in the 1950s and its rebirth in the 20th century, it has played a substantial role in offering effective solutions to significant human and societal challenges across various domains. This includes the field of NLP, which utilizes computational and linguistic techniques to assist computers in comprehending and, at times, generating human languages in the form of text and speech/voice [11].

Prominent research in NLP includes text generation, translation systems, IR, Question and Answering (QA), sentiment analysis, and text summarization, etc [11]. In this research we focus on three NLP tasks, namely, text generation, IR, and emotion detection which is sub category of sentiment analysis.

2.2 Language Models and Chatbots

Modeling a real world topic in general helps us predict. Model's job is to get context and output the prediction that could be the next event or action. For example, weather models can help us predict the next day weather temperature (prediction) having the history of temperatures (context).

Language Models are no exception, they can predict the next token given the previous tokens (context). Language models are based on simple statistical models or deep complex neural networks. Based on their architecture they can generate a vector as an output, or a text. When the output is text it can be next token that has highest probability or a list of next tokens and their probabilities. While, tokens can be a character (i.e. 'g'), a world (i.e. "goodness") or group of characters (i.e. "ness") that carrying a meaning. Language modeling can be done using old school approaches (i,e, Markov models), or more recent approaches such as Long Short-Term Memory (LSTM), and transformers. We will discuss about Text generation task and its background in chapetr 5.

Moving from language modeling into chatbots. The first chatbot (chatterbot) in the world is considered to be ELIZA [68]. It was developed by Joseph Weizenbaum at the Massachusetts Institute of Technology (MIT). ELIZA, a rudimentary software application, replicated conversational interactions by employing pattern-matching methods to generate replies based on user inputs. It adopted the persona of a Rogerian psychotherapist and participated in text-based dialogues, predominantly employing tactics of rewording and mirroring user utterances. Despite its simplicity when evaluated against contemporary benchmarks, ELIZA established the fundamental framework upon which numerous subsequent advancements in conversational AI technologies were built.

Since the advent of ELIZA in 1966, chatbots have evolved in many ways. Researchers explored integrating emotional intelligence [40], knowledge base [4], deep learning [50], reinforcement learning [14], and knowledge graphs [70] to build advanced chatbots.

Chatbots can be categorised into three categories [61]:

- **Rule-based:** Rule-based chatbots are constructed to follow a predefined

set of rules and patterns. However, the potential of rule-based chatbots is constrained by the quantity of rules they employ. While adding more rules can enhance their ability to generate outputs, it also introduces a drawback, namely, an increased potential for rule overlap and out of domain text generation.

- **Retrieval-based:** Chatbots which use retrieval mechanisms are trained to follow directed flows. The retrieval-based chatbots are programmed to find an answer from a set of predetermined responses in order to provide the best possible response.

- **Generative-based:** Advanced NLP algorithms are used by generative-based chatbots to interpret users' intentions from the words and sentences, and respond to them without human intervention. These chatbots are based on LMs and the state-of-the-art architecture that these LMs have are LSTM, or transformers.

Chatbots can also be a combination of these categories. the MIRA chatbot was a retrieval-based chatbot which walks thought a predefined flow of states like a state machine. It predicts the user's intent and extracts the user entity from the user's utterance in every state and decide which state is the next state based on the combination of the predicted intent and extracted entities. An example of training data for a flight finder chatbot is as follows:

Intent: Flying

- Example 1: I want to fly from [YEG (departure)] to [YUL (destination)].

- Example 2: I need a ticket to fly from [YUL (departure)] to [YEG (destination)].

- Example 3: Can you find [round trip (type)] a ticket from [YUL (departure)] to [YEG (destination)] for me?

By training a chatbot model with the above training data for flying intent (class). When end-user asks "I would like to fly from Edmonton to Toronto"

it detects "Flying" as the class or intent of the user message and fills the destination and departure slots with keywords that the user provided in the sentence, "Edmonton" and "Toronto" respectively. This sentence has both destination and departure slots filled, but the type of trip is unknown for the chatbot. It goes to an state where it asks "Is it a round trip or one-way?" to fill the trip type slot too and be able to search among the flights.

2.3 Chatbots in mental health from the Computer Science perspective

Rahman et al. [49] proposed a framework to design a healthcare chatbot to handle queries in the 'Bangla' language. Their approach is based on a Bangla knowledge base that is created by scrapping the internet on the basis of Bangla queries related to any disease. Oh et al. [40] designed a chatbot for psychiatric counselling. Their architecture uses morpheme embedding to comprehend the semantic information of words from a corpus of over 49 million Korean sentences and later uses a Gated Recurrent Unit to properly segregate the intention of an input sentence for maintaining a correct conversation. In many cases, the purpose of a medical chatbot is to extract a patient's through a conversation. This information can be extracted to perform various analyses on the patient or a group of patients. K-Bot [46], is built on a user-friendly interface that records the symptoms of a patient on the basis of binary inputs. The inputs are later used to train a decision tree model to predict a disease. Blanc et al. [5] studied different combinations of language models and neural networks for intent and slot prediction through a French medical chatbot. The study documented optimal results for a hybrid model consisting of FlauBERT and a linear neural network. Huang et al. [23] proposed a Hierarchical LSTMs for Contextual Emotion Detection model to detect the emotion of user utterances regarding the conversation context.

2.4 Chatbots in mental health from the psychology perspective

Recent research on mental health agents underscores the significance of studying chatbot users and their requirements. This is because the interaction patterns and needs of each generation vary significantly [27].

Koulouri et al. [27] investigated the acceptability of chatbots among young adults. Their primary focus was to identify the most beneficial features of mental health technology for youth. They found these four features: (1) self-help therapeutic techniques (e.g. meditation), (2) support availability, (3) Psycho-educational Content (e.g. information about mental health), (4) search functions. They were also eager to see more improvements in interactivity, adaptive content, professional support, and usability.

Brewer et al. [6] studied how older adults (ages +65) interact with voice assistants. A total of 201 men and women participated in the survey, revealing that older adults prefer voice assistants capable of addressing multi-query questions in a single response. Additionally, they expressed concerns regarding their limited control over the sources from which voice bots provide recommendations.

Maharjan et al. [32] developed a Conversational Agent (CA) designed for use with Google Nest smart speakers. They enlisted the participation of 20 individuals diagnosed with depression and bipolar disorder for their study, ultimately uncovering valuable insights:

- In cases where patients were presented with complex questions. Patients employed audible cues to signify their thought process (e.g., "eeeee" or "hmmm"), the CA often interrupted them, leading to difficulties in accurately capturing the user's response.

- Sometimes what is needed is someone or a CA to listen. Since human listeners often focus on "solving" issues instead of just listening, CAs are good company since they can respond with empathy and are always available. They will not judge and search for solutions only if patients

want to.

- People with depression have smaller social circles and they have less opportunity to talk to someone. They enjoy CAs companion more.

Razavi et al. [51] Enabled the operation of a rule-based chatbot named "LISSA," designed to engage in conversations with older adults who may be at risk of experiencing isolation or social anxiety, in 27 everyday topics. In their previous work [52] they described how LISSA was designed. The text generation mechanism employed by LISSA involves an initial step where it makes a pertinent remark regarding the user's input and then proceeds to ask a question. This method relies on keyword-sensitive rules, which unfortunately do not offer assistance in handling unseen examples, as there are no established patterns to process such cases. However, this approach was successful and can be used with rules being extended as LISSA evolves.

2.5 Emotion Mining

Emotion Mining (Emotion Mining (EM)) is a branch of text sentiment analysis that deals with the extraction and analysis of emotions. Emotions like happiness, sadness, fear, and anger, are derived from personal (subjective) experiences of individuals as well as their interactions with their surroundings (audio/visual signals) [22].

EM has been applied in various applications such as emotion mining from suicide notes [69], multimedia emotion tagging [65], emotion intensity prediction in tweets [21], detecting offensive language in conversations [2], and etc.

EM has been studied in various ways, they used different emotion models. Emotion models are systems that define how emotions are represented. Emotion models can be categorized into two main categories [1].

Discrete Emotion Models (DEMs)

Discrete Emotion Model (DEM)s put emotions into distinct categories. Based on the most commonly used models, primary emotion types are typically cat-

egorized into groups of five, six [18], [34], [44].

- Paul Ekman model [18] This model categorises all emotions in six primary emotions. Namely fear, anger, joy, sadness, disgust, and surprise. In 2011, Ekman et al. [19] added one more emotion (Contempt) to the previous Paul Ekman model for emotions.

  These seven emotions are defined as follows [19]:

  Anger: The reaction to any hindrance encountered while pursuing an important goal. Anger may also arise when someone endeavours to cause harm to either us (physically or emotionally) or to someone dear to us. In addition to stopping the harm, anger usually involves a desire to cause harm to the target.

  Fear: The reaction towards the potential danger, physical or psychological. Fear triggers impulses to freeze or flee. Fear often activates anger.

  Surprise: The response to an unexpected sudden event. It is the briefest emotion.

  Sadness: The response to the loss of an object or individual to which you hold a strong attachment. The classic example involves the passing of a child, parent, or spouse. In sadness there is resignation, But, it has the potential to transform into deep distress in which there is agitation and protest over the loss and then return to sadness again.

  Disgust: repulsion by the sight, smell, or taste of something; it can also be evoked by individuals whose behaviour is repulsive or by concepts that are objectionable.

  Contempt: Experiencing a sense of moral superiority over someone else.

  Happiness: Feelings that are enjoyed, that are sought by the person.

- Robert Plutchik model [44]

  This model categorizes all emotions in trust/acceptance anticipation in addition to six main emotions proposed by Paul Ekman. According to

Plutchik, these eight emotions can form four opposite pairs (namely: joy vs sadness, trust vs disgust, anger vs fear, and surprise vs anticipation).

Dimensional Emotion Models (DiEMs)

Dimensional Emotion Model (DiEM)s considers relations between emotions and how they can affect other emotions. It defines emotions based on predefined measures, usually include valence, arousal, and control.

Russel [56], proposed a circular two-dimensional model which distinguishes emotions in the Arousal - Valence domains with Arousal differentiating emotions using Activation and Deactivation, whereas Valence differentiates emotions by Pleasantness and Unpleasantness.

In this research, we opted for the Paul Ekman model [18] because of its simplicity and its self-contained definition of emotions.

Chapter 3

The MIRA Project and the MIRA Resource Library

The MIRA chatbot project has two main sub-systems (Fig. 3.1): MIRA chatbot, and MIRA Resource Library website to store, index, and rank resources for the MIRA chatbot.

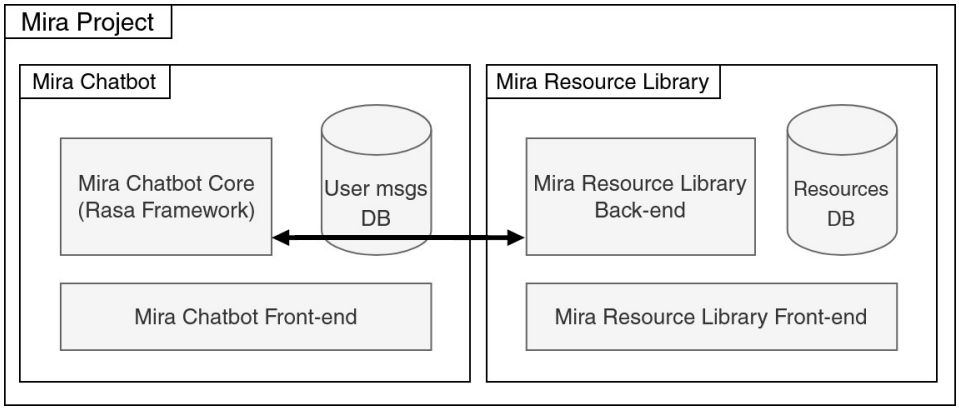

Figure 3.1: The MIRA project high level subsystems

The MIRA chatbot was developed using the Rasa[1] framework which utilizes the Dual Intent and Entity Transformer (DIET) architecture model to detect intents and entities in one model.

The MIRA chatbot front-end is a web-based GUI that connects to the MIRA chatbot Core using a restful API. Rather than relying on external Internet searches to address users' inquiries, MIRA utilizes its own database (MIRA resources) to provide information. The MIRA Resource Library is

[1]https://rasa.com/

a website to perform four operations on a resource namely, Create, Read, Update, and Delete (CRUD), as well as reviewing a resource and search among resources. It is forked from our previous research group chatbot project which is customized for the MIRA project.

The activity diagram in Fig. 3.2 shows, how these components collaborate to find a resource for the user. MIRA chatbot asks questions to extract enough keywords (intents, and entities) from the user input. This information engulfs different aspects of the user's needs and it is used to fill different slots such as "occupation", "health issue", and "location". When all slots are filled, MIRA chatbot sends a query to the resource portal to retrieve the 15 most relevant resources and show them to the user.

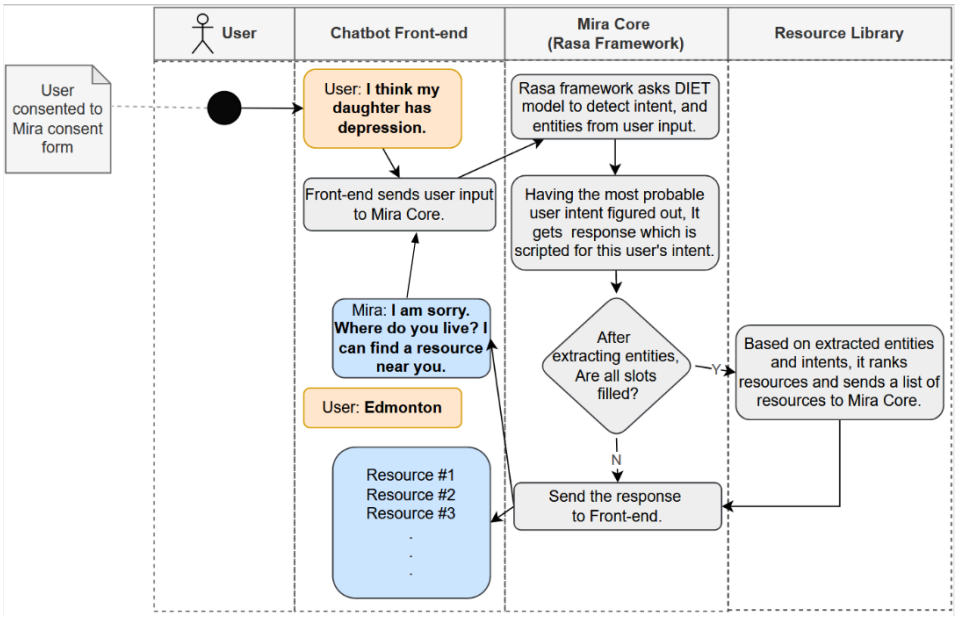

Figure 3.2: Interactions between MIRA modules for finding a resource.

3.1 The MIRA Chatbot

The MIRA chatbot is a retrieval chatbot created to provide resources to its users. It uses a scripted graph to ask a series of questions to fill slots of entities and retrieve the best resource according to filled slots (entities). For example, when the MIRA chatbot asks about the user's age, the response from a user fill the age slot. In some cases, the user asks a question in response to

the question that the bot asked. In those cases, the MIRA chatbot detects that the user is changing the topic and prioritizes the user's intent. The MIRA chatbot plays a crucial role in the MIRA project. Its tasks are intent detection, entity extraction, storing conversation logs in a database, housing the trained models for intent detection and entity extraction, response selection, and spell-checking functionality. Ali Zamani described the process of building MIRA chatbot in his thesis [71].

3.2 MIRA Resource Library

The MIRA resource library consists of both front-end and back-end components. The front-end is developed using the React[2] framework, while the back-end is designed and implemented using the Django[3] framework. The MIRA resource library consists of three different main components (1) CRUD operations and MIRA Database, (2) search engine and ranking function, and (3) an ontology of entities that specify the relation between popular entities in mental health.

The resource verification process involves two different experts reviewing each resource. If both experts agree to verify or reject a resource, the resource is approved or rejected accordingly. In cases where the two initial reviewers do not reach a consensus, a third reviewer makes the final decision.

The MIRA Resource Library holds the responsibility of managing and storing resource data, creating indexes for efficient retrieval, and offering the MIRA chatbot an API endpoint to facilitate resource searches.

The ontology plays a crucial role in the search engine by storing valuable domain-specific information. It aids the search engine in effectively mapping various keywords (entities) within the domain together and identifying parent-child relationships between them. This helps in improving the search engine's ability to consider hierarchies of the concepts within the domain, leading to more accurate and relevant search results. For instance, the fact that the

[2]https://react.dev/

[3]https://www.djangoproject.com/

anxiety and depression are not under the same category but anxiety and abuse are in the same category helps system have domain knowledge about mental health symptoms and causes.

3.2.1 MIRA Database

The MIRA Database is built on MYSQL, serving as its data management system, and it consists of 17 tables. Fig. 3.3 shows how these tables are designed to store various types of data, including information about resources, reviews, authentications, and Django admin logs.

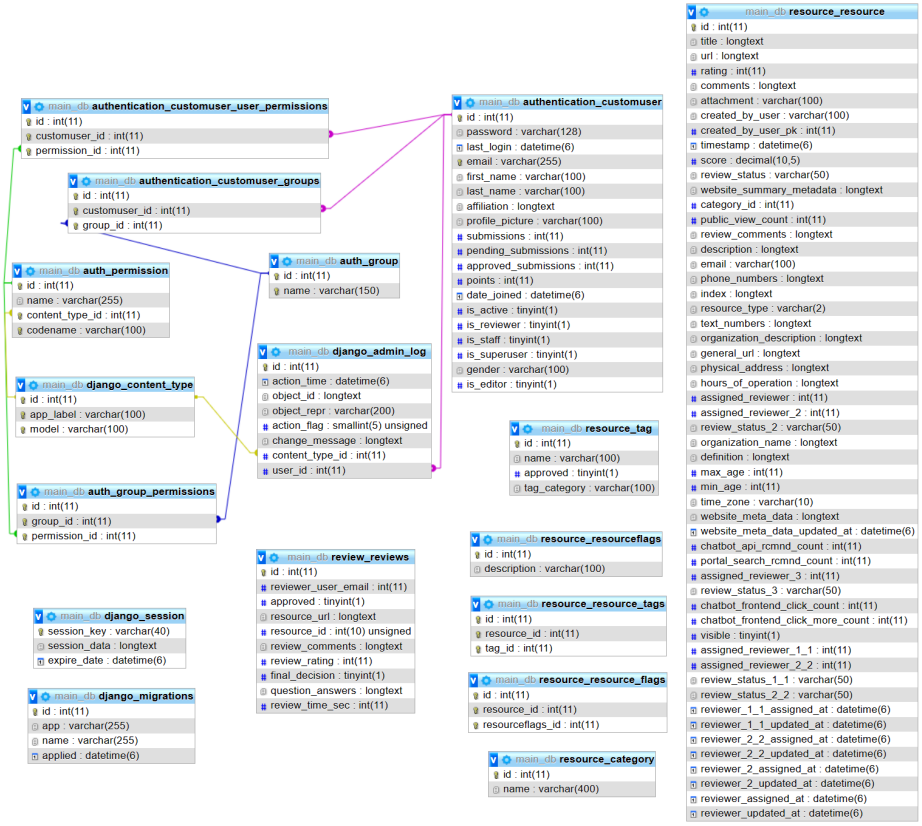

Figure 3.3: The MIRA database tables

In order to gather all the necessary information for each resource we store these data fields for each resource:

- **id:** Unique identifier for the resource.

- **title:** Unique title of the resource which should contains the organization of it.
- **URL:** Unique URL of the resource, it can be null.
- **attachment:** It can be null and specifies a file name. Files are uploaded to a folder on the server and can be accessed using the file name.
- **timestamp:** Time of creating the resource.
- **website_summary_metadata:** It saves all meta data of the resource website at the time of acceptance to periodical checking of whether a website has changed since approval or not.
- **website_meta_data_updated_at:** Last time the meta data was updated.
- **category_id:** Id of a row in "Resource category" table.
- **public_view_count:** Number of times that this resource has been sent to the resource library website to be shown via its API.
- **description:** Details about the resource.
- **email:** Email of the resource. It can be null.
- **phone_numbers:** An array of phone numbers of the resource. it can be null.
- **index:** A Json stored in a text data type. keys are tags and values are Term Frequency-Inverse Document Frequency (TF-IDF) value associated with each tag based on the resource title and descriptions. Tags with a TF-IDF value of zero are omitted from storage, and the dictionary may also have the possibility of being null.
- **resource_type:** Type of the resource. Is one of the RS, SR, and BT which are abbreviation of Resource, Service, and Both.

- **text_numbers:** An array of Text numbers (not phone numbers) of the resource. it can be null.
- **organization_name:** Name of the organization of the resource.
- **organization_description:** Description related to the organization of the resource.
- **general_URL:** Stores base URL of the resource.
- **physical_address:** Physical address of the location that the resource is being provided. Can be null.
- **hours_of_operation:** Hours of operation for each day of week. For example if a resource is only available at 8 AM from Monday to Friday, the value to store is "MON:8;TUE:8;WED:8;THU:8;FRI:8;SAT:;SUN:;". it can be null when it is not applicable.
- **time_zone:** Specifies the timezone of operation hour.
- **assigned_reviewer:** User id of the first reviewer.
- **assigned_reviewer_2:** User id of the second reviewer.
- **assigned_reviewer_3:** User id of the third reviewer (tie breaker).
- **review_status:** This specifies the review status of the resource from perspective of the first reviewer and can be "accepted", "rejected", "pending".
- **review_status_2:** This specifies the review status of the resource from perspective of the second reviewer and can be "accepted", "rejected", "pending".
- **review_status_3:** This specifies the review status of the resource from perspective of the third reviewer.
- **assigned_reviewer:** User id of the first reviewer.

- **definition:** For some resources definition is filled with the definition of the title. (e.g. definition of depression)
- **max_age:** Maximum age that this resource is suitable for.
- **min_age:** Minimum age that this resource is suitable for.
- **chatbot_api_rcmnd_count:** This refers to the count of how many times the search engine, through the chatbot API endpoint, has recommended this particular resource to users.
- **portal_search_rcmnd_count:** This refers to the count of how many times the search engine, through the Resource library API endpoint, has recommended this particular resource to users.
- **visible:** Specifies whether the resource should be visible to public or not.

3.2.2 Resource Library REST API

A RESTful API is a type of API that uses HTTP (or HTTPS) requests to GET, PUT, POST and DELETE data. It is based on the design principles of the REST architectural style, which is an architecture that uses a stateless communication protocol (such as HTTP) for creating web services. RESTful APIs are often used to create web services that are lightweight, scalable, and easy to maintain. They are typically used to expose data from a server to clients, such as web applications or mobile applications.

For every operation it has a different method:

- **GET:** To retrieve a resource.
- **POST:** To create a new resource.
- **PUT:** To edit or update an existing resource.
- **DELETE:** To delete a resource.

Within the resource library, we maintain 34 data points for each resource. Additionally, we conduct periodic meetings for our new volunteers to provide training on resource addition and editing procedures. To further assist them, we have implemented guides with a consistent question mark icon in the user interface that offers supplementary explanations for each field within the resource submission form. This can help them input 34 data points easier and faster.

The resource submission form was optimized to increase the productivity of users. We have implemented search functionality for tags which help them easily type a couple of character and choose from the suggestion list. Additionally, we also implemented a mechanism to hide/show some form data points when they are not applicable. These features helped our users to input more resources faster and easier.

After resources are inputted to the database. We needed to verify them and make sure that they were inputted correctly and were suitable for our patients. We implemented a process similar to the research paper review process. To verify a resource two editors should be assigned. if they both agree on rejecting or accepting a resource the resource ends up being stated as they wish. If they do not have the same idea. Super admin user can break the tie and make the final decision.

The resource library also captures statistics about resource editors. These include the number of resources assigned to an editor with their status (accepted, rejected, pending), and the average time of reviewing a resource. These statistics help the super admin user to assign fresh resources to those who are more available and productive.

After resources are added to the website and verified, there is a need to periodically check them for updates. If a verified resource undergoes any changes, the system automatically sends an email notification to inform our editors. This email reminds editors that the resource requires revision to ensure the information provided remains accurate and up-to-date. This approach helps maintain the quality and reliability of the resources available on the website.

3.3 MIRA Ontology and Search Engine

Taking "controlling depression of my daughter in Edmonton" as an example of user input, slots such as location and health issue are filled by **Edmonton** and **Depression**. Then, these keywords are sent to the search engine in the MIRA resource library.

The search can not be simply done by searching for queried keywords among titles or descriptions of resources and we need a semantic-based search engine with an ontology of popular mental-health entities. For the above query, the most relevant resource is titled "University of **Alberta** health service - **general mental health** clinic", a simple SQL query on the resource titles is incapable of finding the resource as the title words are different from the query keywords. In this case, we need to have the knowledge that Edmonton is a city in Alberta or depression is a general mental health issue. We extended an ontology of entities and mental health concepts, helping the system understand that hierarchy of entities and/or concepts and letting it relax queries based on the related results found.

One additional advantage of employing a semantic-based search engine is its capability to comprehend similarities among synonym words. For instance, the terms "Alberta-wide" and "Alberta" may carry similar meanings, even though they do not precisely match in terms of keywords.

To add location knowledge to ontology we gathered list of all Canadian cities with their provinces and added them to the ontology under the super class of Location. We also added the terms we thought might be used by our end-users to our ontology as synonym of previous entities or new entity.

Fig.3.4 depicts the super classes of the MIRA ontology. If the user is searching for a service in the Red Deer city (a city next to Edmonton in Alberta) but we do not have any resource tagged with "Red Deer". The system should be able to relax the query and retrieve resources tagged with "Edmonton" and "Alberta" instead of "Red Deer".

Our search engine operates in two phases. Initially, it attempts to match each keyword in the query list with existing tags in the MIRA resource database

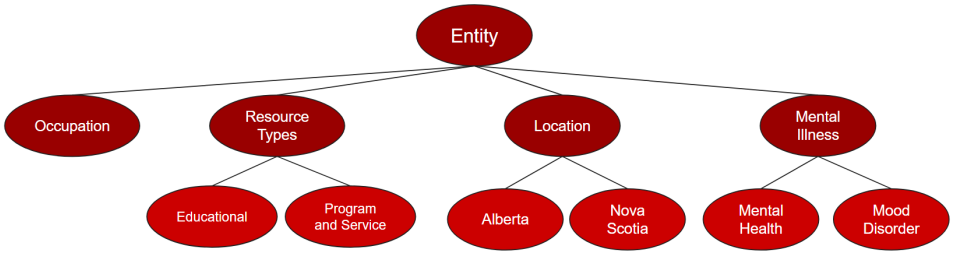

Figure 3.4: The MIRA ontology super classes and their children. Due to shortage of space we showed two first levels of the MIRA ontology.

using a combination of keyword search, edit distance search, and sentence transformer embedding similarity search. Subsequently, it retrieves all resources that have been tagged with at least one of the matching keywords and ranks them accordingly.

To enhance the precision of resource ranking, we implement a system where each resource is assigned a vector of size seven due to seven tag categories we have. Each item in the vector corresponds to a specific tag category and holds the resource's score in regard to that category. Similarly, we transform the query into a seven-item vector as well, representing its relevance to each tag category. In the final phase, resources are ranked based on the dot product of their vector and the query vector, allowing for a more precise ranking mechanism.

Our proposed similarity metric uses these rules to calculate resources score for a tag category based on these rules:

- The similarity between an ontology entity (or tag) and a resource is computed using a TF-IDF model which is trained on all the resources textual data.

- The similarity between an ontology entity and its parent entity is determined by a fixed value, represented as a hyper parameter (e.g., λ) typically set to 0.85 on a scale of 0 to 1. When we relax a query by including results related to the parent entities, we multiply the score by λ for resources tagged with parent entities. This adjustment is necessary because the original query does not explicitly include the parent entity,

and thus, these resources should receive a lower score compared to the main query's results to reflect their lesser relevance.

- The similarity between synonyms in ontology is 1 out of 1.
- We prioritize resources from certain organizations by adding a negligible score to resources' final similarity score.

Algorithm 1. describes the algorithm used in the MIRA resource ranking function.

Algorithm 1. The MIRA ranking function algorithm

Initialize:

1. Map all query keywords to a tag and store them in mapped_tags:list.

2. Find the parents of all items in mapped_tags and store them in query_relaxation_tags:list (according to the MIRA knowledge graph).

3. Store items in mapped_tags with category of "mental health" in VIP_tags:list.

Resource retrieve:

4. Get all Resources:dict that are at least associated with one of the tags in (query_relaxation_tags, mapped_tags) and has at least one association with one of the tags in VIP_tags.

Resource score:

for r=1 to len(Resources) **do**

TF-IDF for tags Step:

5. Find TF-IDF score of the resource for all tags in mapped_tags and put the score in the spot for coresponding category in the seven item embedding.

TF-IDF for query relaxation tags Step:

6. Find TF-IDF score of the resource for all query relaxation tags in query_relaxation_tags and put the score in the coresponding category in the seven item embedding of the resource.

Rules Step:

7. Increase the resource scores based on rules.

end **for**

8. Create a seven item embedding to represent the input query.

9. Calculate the dot product similarity between the embedding of the query and embeddings of all resources.

10. sort resources based on the dot product similarity.

11. return sorted resources.

To assess the effectiveness of our approach, we first conduct an automated evaluation to evaluate the effectiveness of the proposed approach. We also conduct a human evaluation to assess the search engine functionality in collaboration with other components in the MIRA chatbot.

3.4 MIRA Search Engine Evaluation

3.4.1 MIRA Search Engine Automated Evaluation

We employ two other methods for transforming both resources and queries into vectors. These methods were then evaluated using cosine similarity, thus our model's performance was compared to that of TF-IDF and a BERT-based model.

We prepare 50 test cases. Each test case consists of an input query and three resources that is most relevant to that query. The test is passed (hit) when one target resource is among the top five ranked resources by the approach. Table 3.1 shows the result of our evaluation. The BERT-based model was inaccurate in distinguishing similar resources correctly. TF-IDF is very sensitive to keywords and this makes it the worst approach among all. However TF-IDF run time is less than others. Our model uses the same BERT-based model and the same TF-IDF weights but since it uses BERT-based when TF-IDF failed to find the most similar resource, it can reduce the average calculation time in comparison with BERT-based model.

Method	AVG Response time	STD Response time	Hit No.
TF-IDF	**0.1** (s)	0.04	23/50
BERT-based	7 (s)	0.7	35/50
Our Model	1 (s)	0.6	**41/50**

Table 3.1: Comparison of AVG response time and count of hits for search approaches.

3.4.2 MIRA Search Engine Human Evaluation

In order to test the search engine in real-world scenarios. We arranged three internal workshop to test the chatbot. In the MIRA resource library we have resources for different audiences. We divide team members between audience types we had a resource for and asked them to mimic what a person in that audience group will do. The workshop participants did what was requested by interacting with the MIRA chatbot and following the conversation until they were provided with the top 5 related resources. Then there was a shared file in which they log their interaction with the system and add their thoughts about the provided resources. Among 111 tests that were logged only seven of them had issues, those issues were related to inconsistency with the Mira ontology. For example in the ontology, "Children" was one of the sub entities of "Youth" entity. Those issues were fixed by updating the Mira ontology.

Chapter 4

Emotion Detection

Human communication includes effective messages conveyed through the use of emotionally coloured words. In this chapter, we facilitated machine learning approaches, including Naive Bayes, Logistic Regression, Support Vector Machine (SVM), two neural network models consisting of Convolutional Neural Network, Long short-term memory, and Bidirectional Encoder Representations from Transformers (BERT). pre-trained BERT LM, which was fine-tuned on the dataset, outperformed other models with an accuracy of 84.45%.

4.1 Background

In the healthcare domain, there are many chatbots extracting emotions from different channels to detect users' mental health issues. Sawalha et al. [59] use the text of interviews of participants with a virtual character (Ellie) to predict Post-traumatic stress disorder (PTSD) and they achieved a balanced accuracy of 80.4% on a data set used. DeVault et al. [16] and Stratou et al. [62] incorporated audio, and motion tracking, in addition to text data to predict Post-Traumatic Stress Disorder (PTSD).

4.2 Data

Datasets for EM

Acheampong et al. [1] surveyed the state-of-the-art literature in textual EM. Using transformers-based approaches showed a great improvement in captur-

ing contextual information. They used different sources of emotionally rich data.

- **ISEAR** The Swiss National Centre of Competence in Research created The International Survey on Emotion Antecedents and Reactions (International Survey on Emotion Antecedents and Reactions (ISEAR)) database [60]. Seven emotion labels (happiness, sadness, fear, joy, guilt anger, and surprise) can be found in this dataset that is collected from 3000 participants living in 37 countries.

- **SemEval** SemEval is a series of international NLP research workshops in which different tasks are defined. Each task usually includes reasonable amount of prepared related data (e.g. SemEval-2018 Task 1[37]).

- **EMOBANK** More than 10,000 sentences in this dataset [8] have been annotated using the Valence-Arousal-Dominance (VAD) model of emotion representation. This database come from a wider range of sources, including news headlines, essays, blogs, newspapers, fiction, letters, and travel guides. A portion of the dataset has also been categorically annotated using Ekman's basic emotion model.

- **WASSA-2017 Emotion Intensities (EmoInt)** The Workshop on Computational Approaches to Subjectivity, Sentiment, and SocialMedia Analysis (WASSA-2017) [38] data was organized in order to detect the intensity of emotions in tweets. It includes annotations for four discrete emotions: joy, anger, fear, and sadness.

- **Cecilia Ovesdotter Alm's Affect data** The dataset [3] was built from stories and annotated categorically for fear, anger, disgust, sadness, happiness, and surprise. It also includes feeler, intensity, and lists of emotion words used for helping with emotion annotations.

- **Daily Dialog** This dataset [28] was created by crawling regular human conversations in dialogues. It has 13,118 sentences that have been anno-

tated for neutral, anger, disgust, fear, happiness, sadness, and surprise discrete emotion labels.

- **AMAN's Emotion dataset** The dataset [10] detects emotional content in blog posts. It has 1466 sentences with emotion labels. The emotions associated with these sentences were broken down into joy, sorrow, disgust, anger, fear, surprise, mixed emotion, and no emotion. The mixed emotion category includes sentences that express two or more emotions at once or that cannot be assigned to a single emotion. Sentences that depict neutral emotions are defined as having no emotion.

- **Grounded Emotion data** Grounded emotion data [31] investigates the effects of five external factors on tweeters' emotional states. They investigated five types of external factors and demonstrated their impact and correlation with a user's emotional state through extensive analyses. A total of 2557 tweets were annotated for happy and sad categorical emotions, with 1525 tweets being happy and 1032 tweets being sad.

- **MELD dataset** Multimodal EmotionLines Dataset (Multimodal Emotion Lines Dataset (MELD)) [45] for Emotion Recognition in Conversation is a multimodal dataset that includes audio, video, and text. It includes around 1400 dialogues and 13 000 utterances from the Friends television show, with utterances categorised as anger, disgust, sadness, joy, surprise, fear, and neutral.

- **Emotion-stimulus** The Emotion-Stimulus Dataset was created in 2015 by Ghazi et al. [20] and annotated with both the emotion and the stimulus using FrameNet's emotions-directed frame. There are 820 sentences containing both cause and emotion, and 1594 sentences with the emotion tag. Categories in English include: happiness, sadness, anger, fear, surprise, disgust, and humiliation.

In this project, a multi-class emotion detection model module was required to classify a user's utterance into five emotion categories: joy, sadness, anger,

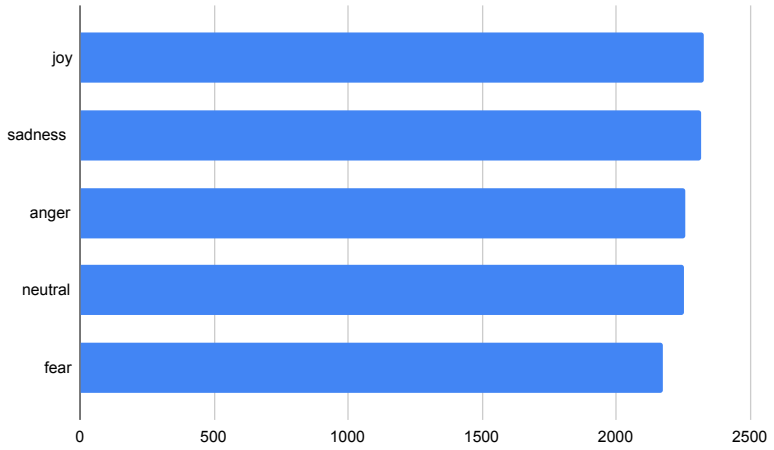

Figure 4.1: Number of items per classes in the merged dataset

fear, and neutral. We consider merging three datasets ISEAR, Daily Dialog, and Emotion-stimulus that are shown in Table 4.1. To the best of our knowledge, there were no balanced dataset that has all emotions we needed. There is also class imbalance problem in two of the databases. Thus we decid to combine all these databases.

Dataset	**Content**	**Size**	**Emotion Categories**	**Balanced**
Daily Dialog	dialogues	102k sentences	neutral, joy, surprise, sadness, anger, disgust, fear	No
Emotion-stimulus	dialogues	2.5k sentences	sadness, joy, anger, fear, surprise, disgust	No
ISEAR	dialogues	7.5k sentences	joy, fear, anger, sadness, disgust, shame, guilt	Yes

Table 4.1: Datasets merged to be used in this project.

We decided to choose Ekamn's basic model which is a discrete emotion model. The fact that emotions are distinct in this model makes the emotion classification problem simpler in comparison with dimensional emotion models.

As shown in Table 4.1, ISEAR[1] has balanced data for each class of emotions. But Daily Dialog[2] and Emotion-stimulus[3] suffer from class imbalance. To cope with the class imbalance, we merged the three datasets and removed some sentences. As a result, a dataset with balanced classes was created depicted in Fig. 4.1.

Each dataset record consists of one sentence with a label specifying the

[1] `http://www.affective-sciences.org/index.php/download_file/view/395/296/`
[2] `http://yanran.li/dailydialog.html`
[3] `http://www.site.uottawa.ca/~diana/resources/emotion_stimulus_data`

emotion expressed in the sentence. Table 4.2 shows three records of the mixed dataset.

Sentence	**Emotion**
When I failed the M.S.C exams	anger
Exactly	neutral
When our dog died	sadness

Table 4.2: Three records of the mixed dataset.

After having data balanced and ready, the next step was applying several prepossessing techniques to the sentences (e.g., lower case conversion, removing punctuation, and stemming) to decrease the amount of noise and vocabulary size and to prepare the data for the classification task.

We chose our baselines to be (1) Support Vector Machine (Support Vector Machines (SVM)), (2) Naive Bayes, (3) Random Forrest, (4) Logistic regression, and two approach based on (5, and 6) neural network. We used BERT sentence embedding as our approach and trained a classifier model based on that. We showed that a BERT-based model can be more accurate and we used that model in our system.

4.3 Traditional Machine Learning Algorithms

As a baseline, we implemented Naive Bayes, Random Forrest, Logistic Regression, and SVM with Term Frequency-Inverse Document Frequency (TF-IDF) [15] input representations to have a taste of the data complexity. In the following, we describe implemented machine learning models.

- **Naive Bayes**: Naive Bayes is a statistical classifier that is based on Bayes' theorem which assumes all features are independent. Naive Bayes uses prior probabilities and conditional probabilities to assign a label to the input records.

- **Random Forrest**: The random forest is a classification technique that uses numerous decision trees to classify data. When creating each individual tree, it employs bagging and feature randomization in order to

generate an uncorrelated forest of trees whose committee prediction is more accurate than that of any one tree.

- **Logistic Regression**: Logistic Regression is another statistical model that is based on the Sigmoid function which is suitable for binary classification tasks.

- **Support Vector Machine**: SVM is another robust machine learning algorithm that maps the training data to points in space and tries to classify the points using a hyperplane in a way that the distance between the closest points of the two classes is maximized.

As an emotion detection task with a unique emotion for each sentence is a straightforward task, we do expect the machine learning models to perform well. However, using complex approaches may increase the performance so we have employed two powerful neural network algorithms to tackle the problem. For implementing traditional machine learning algorithms, Scikit-learn was used.

4.4 Neural Network

The next step is using more sophisticated approaches for solving problem so we employed Conventional Neural Network (CNN), and LSTM for doing the emotion detection task. A pretrained 300 dimension word2vec [35] is used for word embedding and the output of word2vec is used as an input for CNN and LSTM. In the following a brief description of CNN and LSTM is presented.

- **Conventional Neural Network (CNN)**: A CNN is a form of artificial neural network that is especially intended to analyse pixel input and is used in image recognition and processing.

- **Long short-term memory (LSTM)**: LSTM networks are a sort of recurrent neural network that may learn order dependency in sequence prediction problems.

As CNN and LSTM have more complexity in comparison to the traditional machine learning algorithms, we expect that they achieve better accuracy. However, the state of the art approaches use language models for solving natural language processing task. So, in the following, we use a language model. for implementing CNN, and LSTM, Keras framework and Natural Language Toolkit (NLTK) library were employed.

4.5 BERT Language Model

Pretrained language models are trained on large corpora and usually have millions of parameters. we employed BERT pretrained language model to have an embedding of sentence and use it as an input for a fully connected layer. The last layer is a Softmax layer, which specifies a probability for each emotion. We utilized 'BERT-base,' the smallest version of BERT with 12 layers and 110 million parameters, trained on Books Corpus **Zh-2015-ICCV** and English Wikipedia. We also fine-tuned the 'BERT-base' language model on the dataset to achieve robust performance. We expect that BERT model achieve the best performance in comparison to the traditional machine learning algorithms and neural network models since BERT model is trained in a large corpora and yield a better representation of context. For implementing this approach, we used Ktrain framework. We fine-tuned the model using AdamW optimizer [41] with a learning rate of 5e-5 and batch size of 32 for two epochs on 90% of the training dataset and testing the validation performance on the 10% held-out set.

4.6 Classification Evaluation Metrics

classification problems are assessed using precision, recall, and the F1-score.

- Precision: It is calculated by dividing the number of true positives (TP) by the total number of positive predictions, where positive predictions encompass both true positives (TP) and false positives (FP).

- Recall: It is calculated by dividing the number of true positives (TP) by the sum of true positives (TP) and false negatives (FN).

- F1-score: It is calculated as the harmonic mean of precision and recall. F1-score is better to check the overall performance, which fall on a scale from 0 to 1. A score of 1 signifies flawless precision and recall, while 0 denotes the poorest performance.

4.7 Evaluation

To evaluate and compare the performance of the previously mentioned models. We employed 5-fold cross-validation to ensure that the prediction results are robust and accurately represent the model's truth. The result of the traditional machine learning algorithms, CNN, LSTM, and BERT model are shown in Table 4.3. It is worth mentioning that, macro-averaged precision and recall have been reported.

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Naïve Bayes	67.02	69.12	66.95	68.01
Random Forest	64.28	64.55	64.51	62.99
Logistic Regression	69.35	69.55	69.48	69.51
SVM	72.71	72.84	70.02	71.40
CNN	78.80	77.86	78.82	78.24
LSTM	73.95	73.96	71.97	72.95
BERT	**83.45**	**89.09**	**83.05**	**85.96**

Table 4.3: Performance on the test data

We assessed the BERT model using a test dataset containing 3393 test cases. The confusion matrix is depicted in Fig. 4.2. The confusion matrix of all other six method are depicted in Fig 4.3. We observed that "joy" was frequently confused with "neutral", while "sadness" and "fear" were often confused with "anger".

Among traditional machine learning algorithms, SVM achieved the highest accuracy. Among neural network models, CNN outperformed LSTM and overall the BERT language model outperformed all other methods with scoring a higher accuracy.

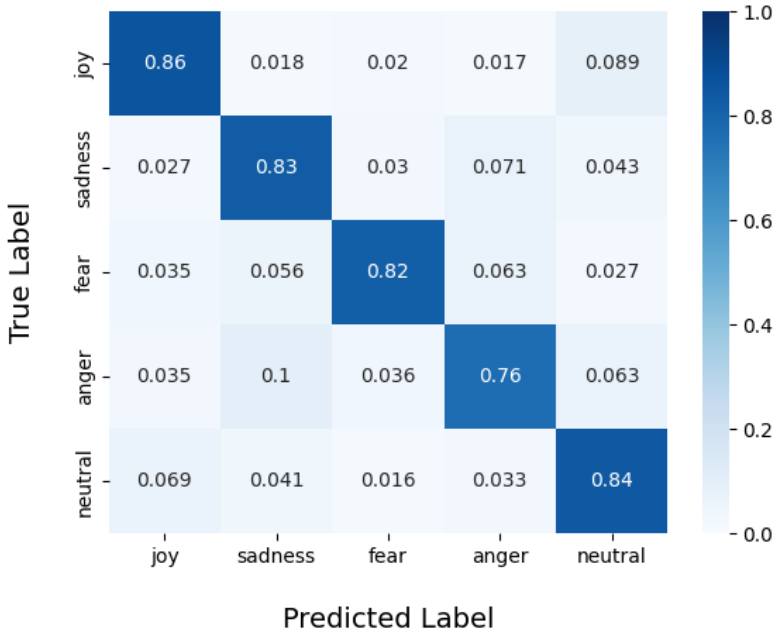

Figure 4.2: Confusion matrix for the BERT model

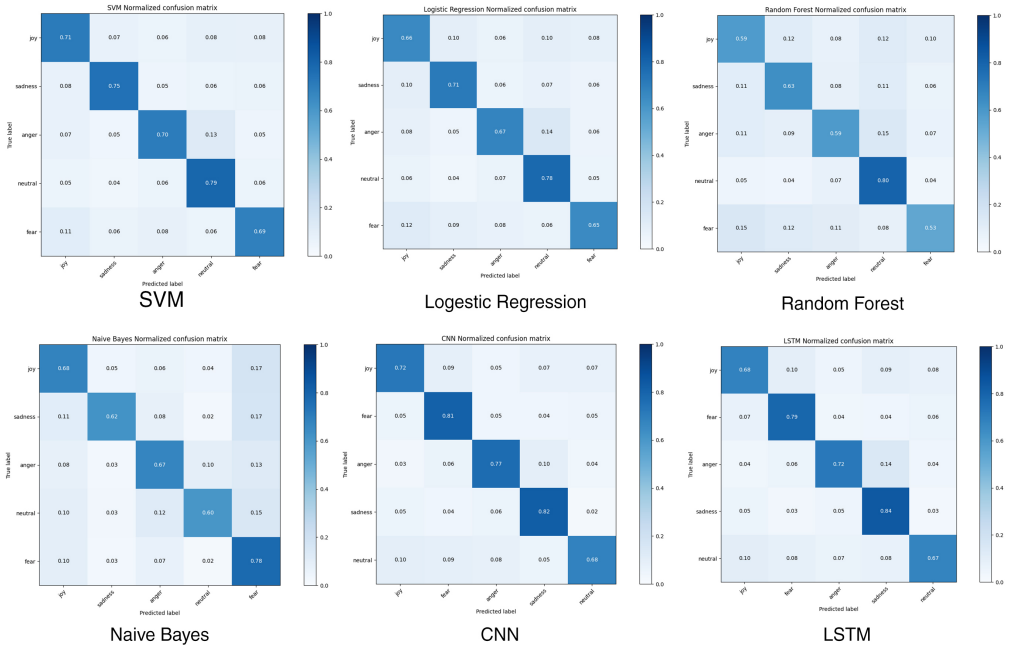

Figure 4.3: Confusion matrix for All methods except BERT

According to the results, fine-tuned BERT is used in our system for emotion detection.

Chapter 5

Response Generation

Text generation is the task of creating text that is indistinguishable from human-written text. Text generation approaches include AI-based or Rule-based approaches. In section 5.1 we reviewed the steps we've undertaken and the insights we've gained during our efforts to address the challenges associated with establishing a rule-based response generation system capable of rivalling medium-sized language models. In section 5.2 we discussed AI-based approaches as well as discussion about why AI-based approaches were not applicable in our sensitive domain.

5.1 Rule-based response generation

Rule-based chatbots have a long history starting from Eliza [68] in 1966. In 1972 an improvement over Eliza was a chatbot named Parry [13] with a personality. In 1996 Alice [64] was developed based on a pattern-matching (rule-based) mechanism. Alice used the AI Markup Language (AIML) which allows chatbot developers and conversation designers to define conversational rules in XML format. Fast forward to the present, and there are multiple Virtual Assistants (Virtual Assistants (VA)s) like Microsoft Cortana[1], Apple Siri[2], Amazon Alexa[3], Google Assistant[4], IBM Watson[5]. These VAs are not completely based

[1]https://www.microsoft.com/en-us/cortana
[2]https://www.apple.com/siri/
[3]https://developer.amazon.com/en-US/alexa
[4]https://assistant.google.com/
[5]https://www.ibm.com/watson

on AI and still use rule-based approaches in some cases.

5.1.1 From ELIZA to Chatty MIRA

According to Wikipedia[6], the ELIZA chatbot was originally implemented in MAD-Slip programming language. Currently, different implementations in various programming languages are available such as Python [63] and Java. We implement our version based on the Java version.
Response generation in ELIZA has the following steps:

- All sentences are segmented into words using spaces as separators. Subsequent processing occurs on the list of words, rather than on the individual characters within each word.
- A series of pre-substitutions occur (e.g. changing "you're" to "you are").
- It processes the input, creating a list of all associated keywords it detects. Subsequently, these keywords are sorted based on their weight in descending order.
- It iterates through all decomposition rules associated with a specific key, attempting to parse the input sentence with it. This process continues until a successful parsing and output generation occur. If none of the decomposition rules linked to the keywords yield a match, the bot responds with a default message.
- It uses an assembly pattern to generate the response. Each decomposition rule could have more than one assembly pattern. It reduces the chance of producing duplicate responses.
- A set of post-substitutions takes place (e.g. changing "I" to "You") and the response is returned.

Although these steps work for most of the cases, one can easily make it fail by understanding its bottlenecks. These bottlenecks are mostly because

[6]https://en.wikipedia.org/wiki/ELIZA

this method relies only on lexical-based approaches which could fail just by a typo error, to tackle this issue, we substituted this module with a hybrid module which has a Sentence to Vector (Sen2Vec) model based on BERT LM (specifically, all-MiniLM-L6-V2). This new approach involves assessing the semantic similarity between the embeddings of the topic and the input. The topics are subsequently sorted based on their resemblance to the input sentences. This sorting is determined by combining both semantic similarity and the topic's rules capability to break down and reconstruct the sentences. If one decomposition rule for a topic can parse an input sentence and also this topic is the most similar topic to the input. This topic is chosen and the response is generated using decomposition rules. But if the user says something that is very close to a topic but not parsable by any decomposition rule. This topic is still selected but the response can not be produced by the decomposition rules instead we use static responses defined to respond to user inputs about this topic.

To solve this, we used Sen2Vec BERT-based model (all-MiniLM-L6-V2) and semantic similarity between topic embedding and the input embedding. If the similarity was more than a threshold, despite the fact that no decomposition rule can parse it. We still use a static response.

Even with this enhancement, the system still depends on predefined patterns to decompose the input and reassemble the output from decomposed sentence parts. However, with the integration of a Sen2Vec model, it is not able to use decomposition and assembly rules. Instead, it employs pre-defined static responses to answer the user's input.

Based on this Benchmark[7] the all-MiniLM-L6-v2[8] model has the best accuracy with high speed and relatively low size. Better sentence encoders exist, but they increase our response time and are not that much different. We used this model without fine-tuning (zero-shot learning). This model's output is a 384-dimensional dense vector. Cosine similarity calculation between sentence

[7] `https://www.sbert.net/docs/pretrained_models.html#sentence-embedding-models/`

[8] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

Model Name	Performance Sentence Embeddings (14 Datasets)	Performance Semantic Search (6 Datasets)	Avg. Performance	Speed	Model Size
all-mpnet-base-v2	69.57	57.02	63.30	2800	420 MB
multi-qa-mpnet-base-dot-v1	66.76	57.60	62.18	2800	420 MB
all-distilroberta-v1	68.73	50.94	59.84	4000	290 MB
all-MiniLM-L12-v2	68.70	50.82	59.76	7500	120 MB
multi-qa-distilbert-cos-v1	65.98	52.83	59.41	4000	250 MB
all-MiniLM-L6-v2	68.06	49.54	58.80	14200	80 MB
multi-qa-MiniLM-L6-cos-v1	64.33	51.83	58.08	14200	80 MB
paraphrase-multilingual-mpnet-base-v2	65.83	41.68	53.75	2500	970 MB
paraphrase-albert-small-v2	64.46	40.04	52.25	5000	43 MB
paraphrase-multilingual-MiniLM-L12-v2	64.25	39.19	51.72	7500	420 MB
paraphrase-MiniLM-L3-v2	62.29	39.19	50.74	19000	61 MB
distiluse-base-multilingual-cased-v1	61.30	29.87	45.59	4000	480 MB
distiluse-base-multilingual-cased-v2	60.18	27.35	43.77	4000	480 MB

Figure 5.1: Pretrained Sen2vec models ranking. (Sorted by average performance of semantic search and sentence embedding tasks.)

embedding of the topic of rules and the input sentence improved our rule selection accuracy. Table 5.1 shows the impact of using sentence embedding in reducing errors while choosing the right rules. Our test cases were taken from conversations that our testers had with it. After removing duplicates, 30 different questions were chosen to be used in comparison as test cases.

Model	F1-Score
Original Eliza	61%
Chatty MIRA	92%

Table 5.1: F1-Score comparison between the original Eliza and Chatty MIRA.

After having a good similarity metric for ranking "decomposition rules" now it is duty of "assembly rules" to generate response.

Text generation using regular expressions was also another weak point of the original ELIZA. To show the problem of using regular expressions for text generation, take this sentence as an input "I am worried it's because I'm too stressed out." ELIZA responded with "Do you believe it is normal to be worried it's because you're too stressed out?". Regular expressions cannot split sentences correctly because rules are not 100% accurate. Using rules for

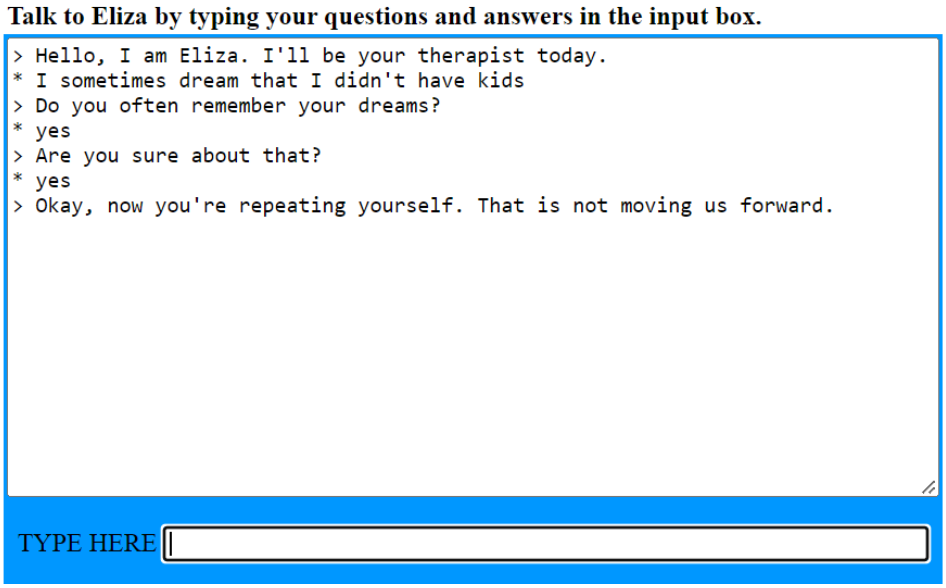

Figure 5.2: Example of inappropriate wording and how yes/no questions can make problems for the original Eliza response generation model.

assembling the words and generating responses could also lead to poor results due to the same reason that rules are not always accurate. To solve this issue in Chatty MIRA we first developed static responses for every rule instead of dynamic regular expression-based rules.

To have a good static response for every topic, we replaced assembly rules with a more general static response. We revised the text used in the original Eliza rules to make sure words are appropriate and questions that our version of Eliza can generate are not informal. For example, we removed these sentences from the response list "Do you feel strongly about discussing such things?" or "Can you be more positive?". These are yes/no questions and are against our goal. In MIRA the main goal is to ask more questions and encourage users to express more which can not be achieved with yes/no questions. Because the answer to a yes/no question is one word which makes the next response generation hard. We also observed that some versions of the Eliza have a policy to prevent users from saying something twice. Fig 5.2 shows how this policy can be destructive when it is coupled with yes/no questions. We also removed yes/no questions and used elaboration and explanation questions as much as possible.

5.2 AI-based response generation

Unlike rule-based models, AI-based models are designed machine learning-based programs that learn from data. The use of ML algorithms eliminates the need to manually define and implement new pattern matching rules, allowing chatbots to be more flexible and less reliant on domain-specific knowledge [9].

Using pretrained models gained popularity after the publication of static word representations like Word2vec [36] and GloVe [42]. Embeddings from Language Model (ELMo) [43] model was able to create contextualized word representations. This was grounded in a bi-directional LSTM capable of producing context-aware embedding, marking the first instance of such an achievement. Next year in 2019 BERT model [17] which was based on transformer architecture showed state-of-the-art results. After that generative LMs became more popular like GPT model [47].

AI models used in chatbots can be divided into two types [9]: Retrieval and generative models.

- **Retrieval:** Information retrieval models are created based on a dataset of textual information. The model can retrieve the information required based on the user's input.

- **Generative:** As the name implies, these models generate new responses, based on the user's input. These models must trained over large amounts of data in order to understand sentence structure and syntax. However, after training over a large amount of data, these models are still able to output inconsistent or low-quality text.

The GPT-3 [7] paper changed the prevailing approach from fine-tuning pretrained models for individual tasks to utilizing a single large model for almost every task, achieved by only modifying the prompts given to the model.

Unlike traditional supervised learning methods, which train a model to take in an input x and predict an output y as $\mathrm{P}(y|x)$, prompt-based learning is based on LM which models the probability of next token directly. Liu et al. [30] surveyed and organized research in the prompt-based learning paradigm. They

defined prompt-based learning as a paradigm which uses models to perform every prediction task. In this paradigm. the original input x is modified using a template into a textual string prompt x' that has some unfilled slots, and then the language model is used to fill the slots to obtain a final string xˆ, from which the final output y can be derived. Since language models are able to understand the context and fill a slot, almost any language model can be used with this paradigm. However, to have the best results there are considerations.

5.2.1 Design Consideration for Prompt-based Learning

Considerations have an effect on the model we choose for prompting and our strategy. Wang et al. [66] studied the impacts of pretraining objective and architecture choices on the zero-shot generalization abilities of the LLMs. They used the T5X library [53] which simplifies the process of building and training LLMs at large scale with different training objectives. They defined three primary language model training objectives:

- Full Language Modeling: all tokens in the training are used.
- Prefix Language Modeling: approximately, half of the tokens are used for training and prefix size is randomly selected. Loss is not calculated for prefix tokens in the training dataset.
- Masked Language Modeling: a part of the training dataset is masked (e.g. 15%) and in a span of a couple of words (e.g. 3 words) and the loss is calculated for the masked words.

They conducted a systematic review of how a model's performance can change when it is "pretrained" and/or "finetuned" by an objective. They trained different model architectures with different pretraining and fine-tuning objectives while keeping the computation resources needed to train all models the same. They found that a casual "decoder-only" pretrained with "full language modeling" objective performs best if evaluated immediately after pretraining, whereas when adding a multitask fine-tuning step, an "encoder-decoder" pretrained with "masked language modeling" performs best. They

proposed T0 for prompt-based learning. T0 [58] is a multipurpose LM with 3-11 billion parameters in different versions. It is based on T5 LM [48] which is an encoder-decoder transformer, and pretrained with a masked language modeling-style objective on C4 dataset[9].

In the MIRA project, we had limitations that prevented us from using any LLM (i.e. T0). It consumes 44 Gigabyte of hard drive and needs about 100 Gigabytes of RAM at run time. Since we were not able to serve many customers from around the world with this LLM we decided to test this paradigm with other smaller LMs such as T5, T5 v1.1, and GPT-2 without finetuning only by prompting.

5.2.2 Refining Text Generation Task for the prompt-based approaches

For prompt-based learning, it is essential to refine the task in a way that the LM could be used the best. Every LM requires a different form of prompting. LMs are generally able to predict the next token/word/sentence when they are given a text prompt. A good Prompt for a language model is the one that helps the model understand the context and is similar to the form of what the model was originally trained on. For a single label binary classification task, the text prompt could be "[INPUT], pick a class which is more related to the previous sentences. [CLS A], [CLS B]". In this example, the model should be able to read the input and generate either "CLS A" or "CLS B".

In our domain, user utterance is the input (X), it detects the emotion of the user's input (F). We changed the ELIZA assembly rules in a way that they are able to get the X and F and generate a prompt (X') for our language model. We customized our dynamic rule-based response generator model to generate prompts for our language model. Then the prompt is sent to the LM model, which generates the response of the language model (Y). We used improved ELIZA rules for prompting T5, T5 version 1.1 and GPT-2, while using a simple template for Falcon-7b, and GPT-3.5.

During our feasibility study, we observed that medium-sized LM such as

[9]https://huggingface.co/datasets/c4

GPT-2, and T5, are very limited in prompt-based generation in comparison to LLMs like GPT-3.5. So instead of asking humans to evaluate all these models. ran an automatic evaluation to remove some of the approaches.

We decided to conduct two evaluations. A human evaluation study and an automated evaluation among LMs and rule-based models to find the best out of them. All candidates in this evaluation are options we are able to utilize in our client's server. After comparing candidates, the best candidate will be chosen to be used in the Chatty MIRA text generation module. To compare its capabilities more, we compare it with Falcon-7b and GPT-3.5 in our human evaluation study.

5.3 Automated evaluation

In this section, we compare rule-based to AI-based models in generating a good response according to the Corpus of Linguistic Acceptability (CoLA) benchmark [67]. After finding the highly ranked models of each category, we discuss how the calculation time was for both top candidates in every category (rule-based and AI-based).

5.3.1 Acceptability

We used CoLA benchmark to automatically evaluate which model can generate responses that are grammatically acceptable in English. According to its website[10], "The Corpus of Linguistic Acceptability (CoLA) in its full form consists of 10657 sentences from 23 linguistics publications, expertly annotated for acceptability (grammatically) by their original authors." We built a classifier with this dataset.

We gathered 20 test cases from the user messages logs and we used those test cases for all models in rule-based and AI-based category to generate the response. We implement a tester program that runs all models for all test cases and calculates CoLA score for the generated output. Rule-based models do not need a prompt, but AI-based models need a prompt generator. We used one of

[10]https://nyu-mll.github.io/CoLA/

our rule-based model as the prompt generator for all AI-based models, instead of implementing a separate prompt generator for generative LMs, we used the "dynamic rule-based" model as a prompt generator for T5, T5 v1.1, and GPT-2. The prompt template that we use is: "rephrase: [dynamic rule-based output]". If AI-based models achieve the same score as "dynamic rule-based" models. It means that the AI-based model did not improve the prompt and it could not make the prompt better.

Model	**Average CoLA Score**
Dynamic rule-based	0.970
GPT-2	0.970
Static rule-based	0.961
T5	0.959
T5 v1.1	0.798

Table 5.2: Average CoLA Score (out of 1)

As depicted in Table 5.3, the scores are quite comparable, with only T5 v1.1 lagging behind. Among AI-based methods, GPT-2 emerged as the top-performing model with a score of 0.970 out of 1, while dynamic and static rule-based approaches exhibited very similar performance, achieving average scores of 0.970 and 0.961 out of 1, respectively.

5.3.2 Execution Time

Because the "dynamic rule-based" model is the prompt generator for GPT-2, the execution time of the "dynamic rule-based" model should be definitely less than the AI-based model. The execution time is shown in Table 5.3 in Seconds.

Model	**Avg execution time (Second)**
GPT-2	8.5
dynamic rule-based	4.5

Table 5.3: Average run time in milliseconds for GPT-2 and "dynamic rule-based"

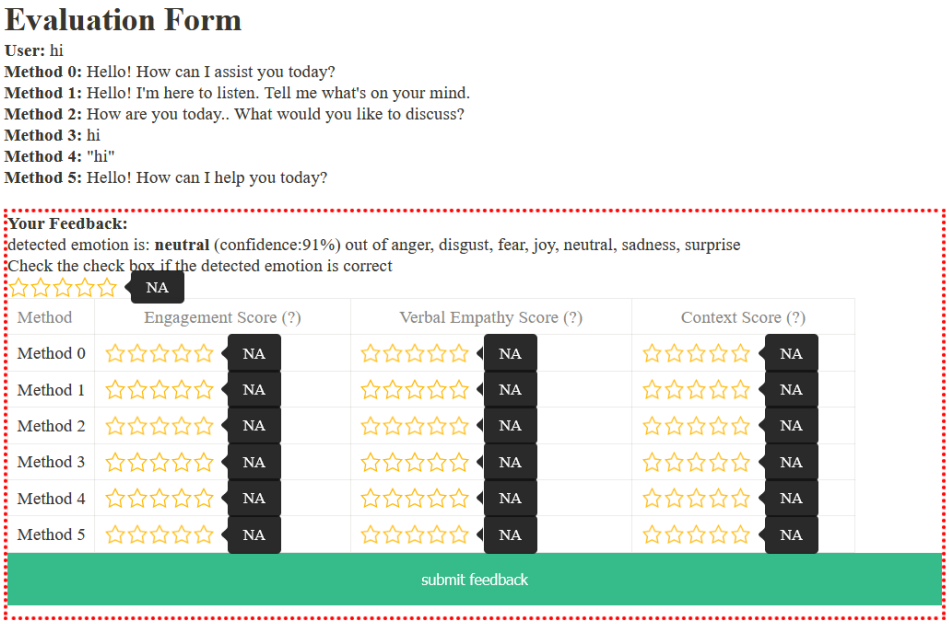

Figure 5.3: Evaluation form which shows responses of the different models to the user input and gets user feedback.

5.4 Human evaluation

We developed Chatty MIRA by employing a hybrid rule-based approach that combines both dynamic and static response generation methods.

We built an HTML page for comparing Chatty MIRA with ELIZA (Java Script version by George Dunlop[11]), Falcon-7b[12], MPT-30b[13], and GPT-3.5-turbo[14].

Fig 5.3. illustrates the output of the five models: GPT-3.5, ELIZA, Chatty MIRA, and Falcon-7b and MPT. After getting input from user the evaluation form transforms to Fig 5.3. We asked evaluators to rate methods for these measurements: Engagement, Verbal empathy, and contextual relevance. if any measurements not applicable the user can rate it as zero star. "NA" scores does not included in the average score calculation.

Our measurements were defined as follows.

- User Engagement: Ability of the system to capture and maintain the

[11] www.peccavi.com
[12] https://huggingface.co/tiiuae/falcon-7b
[13] https://www.mosaicml.com/blog/mpt-30b
[14] https://platform.openai.com/docs/models/gpt-3-5

attention or interest of the user. For example "What else can you recall from that dream" could be an engaging response to "I saw a bad dream last night".

- Verbal Empathy: Verbal empathy is meant to capture the two goals of understanding another's feelings and then reflecting that understanding via dialogue, conversation cues, acknowledgement, or other ways of saying "I hear you" (you can think of this as the best verbal empathetic response). For example "That sounds difficult..." could be a good empathetic response to "I had a hard exam".

- contextual relevance: Ability to understand and maintain an understanding of the conversation and the information that has been shared during that conversation. For example "Hello! how can I help you?" is not a in-context response to "Do you like me?".

Table 5.5 illustrates the outcomes of the human evaluation. Chatty Mira shows significant enhancements compared to its base model, ELIZA. It even outperformed Falcon-7b, highlighting the potential of a rule-based approach. Nevertheless, it is important to note that Chatty Mira still has a considerable distance to match the performance level of GPT-3.5 and MPT-30b.

A glance at the data represented in Table 5.5 demonstrates that a well-constructed rule-based model can excel and even surpass the capabilities of LMs. Nevertheless, GPT-3.5 and MPT-30b performance remains beyond the reach of rule-based systems.

Test Data	**S**	**D**	**T5**	**T511**	**GPT2**
that is my desire to find someone who loves me	0.92	0.98	0.98	0.96	0.98
I do not feel good right now	0.94	0.98	0.88	0.86	0.97
I do not like my brother	0.98	0.98	0.98	0.89	0.98
they are bad	0.89	0.98	0.98	0.33	0.98
I remember it was cold the first Winter of the Edmonton	0.97	0.97	0.97	0.78	0.97
my childhood, I remember it was very good	0.97	0.97	0.97	0.78	0.97
I had a better life if I was still a baby	0.98	0.98	0.98	0.92	0.97
I hate my family, because I do not like how they treat me.	0.95	0.95	0.94	0.80	0.96
I hate my husband, because he is not responsible.	0.95	0.95	0.94	0.80	0.96
sorry	0.95	0.98	0.98	0.89	0.98
I am very sorry for my neighbour	0.97	0.97	0.96	0.86	0.97
I'm sorry to keep you waiting	0.97	0.95	0.92	0.15	0.95
He suddenly felt sorry for her and was vaguely conscious that he might be the cause of the sadness her face expressed	0.98	0.97	0.97	0.90	0.97
I remember my childhood was so beautiful	0.97	0.97	0.97	0.78	0.97
I also remember the beach, where for the first time I played in the sand	0.98	0.97	0.97	0.78	0.97
I will kill you if you come to me again	0.97	0.97	0.97	0.78	0.97
I dreamed about Annie all last night.	0.97	0.95	0.90	0.92	0.97
I dreamed about him every night.	0.97	0.97	0.97	0.89	0.95
are you a good bot?	0.97	0.98	0.98	0.95	0.98
I really thought that I am sick when I found out Alex was a Mexican	0.97	0.98	0.98	0.94	0.98

Table 5.4: Test cases and the CoLA score of each model's response. S: rule-based static, D: rule-based dynamic, T5: T5 model (AI-based), T511: T5 v1.1 (AI-based), GPT2: GPT-2 (AI-based)

Model	**Engagement Score**	**Verbal Empathy Score**	**Context Score**
ELIZA	1.69	1.58	2.04
Chatty MIRA	2.43	2.20	2.09
GPT-3.5	4.84	4.58	4.84
Falcon-7b	1.82	1.74	1.86
MPT-30b	4.46	4.34	4.76

Table 5.5: Human evaluation scores of User Engagement, Being in context. Scores are out of 5.

Chapter 6

Emotion Expression

In previous chapters, we develop an emotion detection model that uses both static and dynamic response generation methods for text generation. Also in order to detect the topic that the user is concerned about it uses decomposition rules as well as a sentence transformer to find semantic similarity of the input to topics we support.

Then we train a BERT-based model to detect emotions which had good enough accuracy based on F1-score we reported in the evaluation. Having these components ready we can implement a component to generate empathetic responses based on the user's detected emotions. In this Chapter we discuss how we addressed this in a way that fits in our customers limited server.

6.1 Empathy

The Western conception of empathy has its origins in two sources (English and Germany). The term empathy in English is derived from the Greek root pathos, denoting emotions, feelings, suffering, or pity (there is also a connection to a German word) [33].

Empathy defined as feeling or understanding what an individual is experiencing[54]. For patient care, showing greater understanding and being better at addressing the specific needs of each person is very important. And empathy has been identified as the main factor to achieve this [25]. Empathy can be expressed in verbal and nonverbal way [24]. In this project, our aim is to concentrate on verbal empathy.

Verbal Empathy consists of a) letting patients know they are not alone by normalizing the situation (i.e. "We all struggle with this"), b) Acknowledging emotions (i.e. "you are feeling really frustrated") c) noticing strength points (i.e. "thanks for sharing this to me"), and d) making sure there is no judgment [24].

6.2 Verbal Empathy in Chatty MIRA

Chatty MIRA without emotion detection/expression, uses dynamic or static response generation for responding to the user. This response was not aligned with verbal empathy definition.

In our prototypes, we tried duplicating all existing responses including dynamic response generation rules and static predefined responses to have an empathetic version of them as well as a neutral version of them. This was not successful, because to be empathetic we should respond to the sadness of the user with sadness and happiness with happiness, etc. So for every emotion, we needed a duplicate version of all responses that are aligned to be a candidate as a response to that emotion. It was not scalable and the amount of repeated words among the empathetic-happy version and neutral was very high. So we changed our approach to keep neutral responses (both dynamic and static) and added another sentence making them more empathetic.

Instead we created two bags of sentences for considering the emotion of the user. One bag has sentences good for responding to negative emotions. The other bag of sentences has sentences good for responding to positive emotions.

By adding one sentence from these bags to the beginning of the neutral response we were able to mimic that we understand the user's feelings and concerns for that. In some cases, the emotion detection component does not predict an emotion with confidence. We need to show that we are not pretty sure but we have detected an emotion but we are not sure. In addition to the negative feeling and positive feeling response bags, we also introduced the Clarification Bag. Which is used for confirming both positive and negative emotions. We also added the sense of the popularity of an issue and showed the

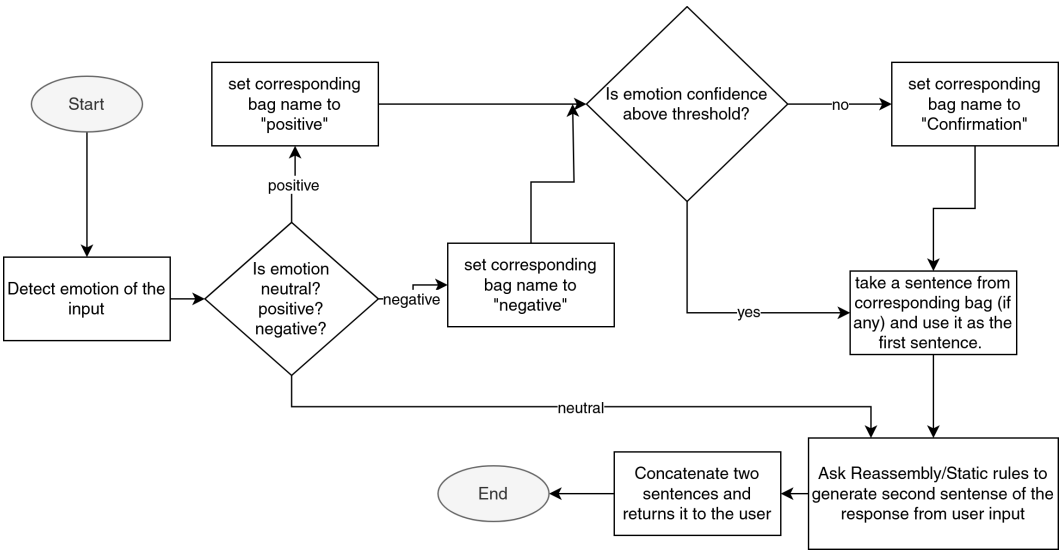

Figure 6.1: Flowchart of the empathetic response generation.

strength of the patient in the dynamic response generation rules and predefined static sentences based on the topic and how we were able to do that. By doing so we were able to generate an Empathetic response that has all four characteristics of an empathetic response.

There are weaknesses that using a pattern-based approach brings with it which is generating sentences that are not grammatically correct or odd to say in that way due to the decomposition rules. so it is possible to generate odd sentences in corner cases when the model is using dynamic decomposition rules.

Fig 6.1 shows the flowchart for adding verbal empathy to our response generator component, we opted to append an opening sentence to the previous response. This sentence aims to demonstrate our comprehension of the user's emotions by utilizing the emotion detection component. We built three predefined sentence bags including Clarification Bag, Negative Feeling bag, Positive Feeling bag. Based on the emotion detected by the emotion detection model and the confidence of it we chose a sentence from the corresponding bag. Then we use Chatty MIRA text generation component which we have updated its reassembly rules to use wordings that are showing chatbot understands strength points of the user and the user is not alone having an issue.

6.3 Evaluation

During the Human evaluation, we asked our participants to evaluate the "Verbal Empathy" as one of the metrics for every conversation turn. The models were rated on a scale of zero to five, where a score of zero indicated inapplicability. Out of the 59 conversational exchanges participants had, GPT-3.5 and MPT-30b emerged in the top two positions, achieving an average score of 4.58, and 4.34 out of 5 respectively. Following was Chatty MIRA, with an average score of 2.2, trailed by Falcon-7b at 1.625, and Eliza at 1.58.

For examples in one of the conversation turns the user said: "Work has been really tough :(". Bellow is the responses of each model to it:

- ELIZA: I see.
- Chatty MIRA: That can't be easy to sit with. Let's discuss further. Tell me more about that.
- Falcon: I'm sorry to hear that. What's been going on?
- MPT-30b: I'm sorry to hear that work is hard for you. Can you tell me more about what's making it difficult? Maybe I can offer some suggestions or support.
- GPT-3.5: I'm sorry to hear that work has been tough for you. It can be really challenging when we feel overwhelmed or stressed out. Is there anything specific that's been bothering you?

The emotion that was detected for this input was sadness and the evaluator gave it 5 out of 5 since the detected emotion is accurate. ELIZA was not able to communicate any words to show empathy. Chatty MIRA was confident about the sadness and it started the response with "that can't be easy to sit with" from the negative response bag. Then it asks to discuss further to get more info about it. Both Falcon versions generated a response containing empathy followed by a sentence asking for more info. GPT-3.5 and MPT-30b were able to show empathy by repeating the user input and also describing

how it feels with words such as "overwhelmed" and "stressed out" which gives a sense of being listened to by the user better than other models.

Eliza wordings was not appropriate at all. "Say, do you have any psychological problems?" was what Eliza responded to "I would like counselling services". Chatty MIRA responses looks artificial. For example, "I can see how that would be difficult. I want to know why sad?" was what Chatty MIRA generated for "I am feeling worried and sad". It is asking in context questions but the question is not what a consultant would ask. Falcon-7b usually returned with short responses, for example it responded with "I'm here for you" as the response to "I could really use a friend".

To summarise, in this chapter we defined and evaluated verbal empathy and showed how responding according to the user's emotion (one of the main factors in verbal empathy) can make the chatbot be more empathetic from the user's point of view.

Chapter 7

Conclusion and Future Work

Our motivation to conduct this research was to (1) provide users with resources and (2) make them able to chat about their issues. We investigate these research questions in this research:

- RQ1) How to design a semantic search engine able to search among thousands of resources in an acceptable time for users, on a limited-resource small server?
- RQ2) How can rule-based or medium-sized AI-based approaches be used to generate responses when scenario is unscripted, on a limited-resource small server?
- RQ3) How to detect emotion expressed by a user in the Chatty MIRA context, on a limited-resource small server?
- RQ4) How responding based on user's emotion can show Chatty MIRA's empathy to users?

7.1 Research Question 1

MIRA was able to detect intents and entities and make a query to resource library we created. We build a search engine which uses a knowledge graph to relax queries and to sort resources verified on the resource library database. We proposed a model which was fast, semantic-based, able to achieve high accuracy in our evaluation test (Chapter 3).

In chatty MIRA we implemented two modules for detecting user's emotion and expressing appropriate emotion.

7.2 Research Question 2

For expressing appropriate response, we compared rule-based approaches as well as AI-based approaches. We suffered from lack of computation resources, and having real consulting dialogues to fine tune language models on them. In general supervised learning for response generation was not applicable and we were only able to use generative AI. We tested prompt-based response generation which does not riles heavily on data. Among all models we were able to deploy on our servers or buy their subscriptions we could not generate satisfactory results and chose to use rule-based response generation (Chapter 5).

7.3 Research Question 3

For detecting user's emotion we used popular datasets available for emotions we needed and built different classifiers. After evaluation we chose BERT-based classifier for this task 4. Our main challenge was finding datasets that we can trust.

7.4 Research Question 4

We assessed the empathetic responses of various models through a human evaluation. Our main challenge was to develop a heuristic that could be universally applied to all scenarios. We endeavored to mimic the user's emotions in order to enhance the empathetic quality of Chatty Mira responses, and the results of the human evaluation validate our progress 6.

7.5 Future Work

We started research in text generation for unscripted scenarios in 2022. In 2022 Ruis et al. [55] stated that language models are not zero-shot language

communicators. However, in the same year there were predictions by Google researchers [12] showing that T5-based model checkpoints are evolving dramatically in the past years and will evolve even more for tasks they have not been trained on which makes them appropriate for response generation in a domain with poor training data. In 2023, we observed an increase in the introduction of new LLMs to the public. However, despite the progress in research, the industry remains cautious and skeptical about the technology's potential problems rather than its capabilities. To encourage industry adoption of the latest generative AI, we initially began with straightforward and cost-effective rule-based approaches.

We dedicated all our time to enhance Chatty MIRA, which relies on a rule-based model. This marked our initial step: establishing a model capable of engaging in conversations and producing reasonably good responses. With increased time and funding at our disposal, we are now able to handle extensive research on generative AI and its applications in this sensitive domain.

At the time we required it, there weren't open-source and promising LLMs like Llama 2 available to us. Medium-sized LMs weren't justified by the computational resources they demanded. This is why we view rule-based text generation as a preliminary step towards employing new open-source LLMs like Llama 2 , Falcon, MPT for this purpose. We prefer employing a rule-based text generation approach combined with transformers to identify the conversation's topic. The current Chatty MIRA model can serve as a baseline for future research on LLM-based generative AI. Further research is required to establish safeguards for LLMs to ensure they will not produce harmful responses.

In Chapter 3 we discussed how we built resource library and the ranking function. LLMs such as Llama 2 can also find a valuable place in the MIRA resource library. Instead of just presenting the top five resources to users after ranking them for a query, LLMs can be utilized to summarize the content of the highest-ranked resources and rephrase it to align more effectively with the user's specific needs.

This project is also planned to be extended to support more provinces

in Canada and more languages (i.e French) as well as aligning with different cultures. This needs using multi language AI models to be able to understand mixture of two languages in a sentence.

References

[1] F. A. Acheampong, C. Wenyu, and H. Nunoo-Mensah, "Text-based emotion detection: Advances, challenges, and opportunities," *Engineering Reports*, vol. 2, no. 7, e12189, 2020.

[2] M. Allouch, A. Azaria, R. Azoulay, *et al.*, "Automatic detection of insulting sentences in conversation," in *2018 IEEE International Conference on the Science of Electrical Engineering in Israel (ICSEE)*, IEEE, 2018, pp. 1–4.

[3] E. C. O. Alm, *Affect in* text and speech*. University of Illinois at Urbana-Champaign, 2008.

[4] G. Battineni, N. Chintalapudi, and F. Amenta, "Ai chatbot design during an epidemic like the novel coronavirus," in *Healthcare*, Multidisciplinary Digital Publishing Institute, vol. 8, 2020, p. 154.

[5] C. Blanc, A. Bailly, É. Francis, *et al.*, "Flaubert vs. camembert: Understanding patient's answers by a french medical chatbot," *Artificial Intelligence in Medicine*, p. 102 264, 2022.

[6] R. Brewer, C. Pierce, P. Upadhyay, *et al.*, "An empirical study of older adult's voice assistant use for health information seeking," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 12, no. 2, pp. 1–32, 2022.

[7] T. Brown, B. Mann, N. Ryder, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[8] S. Buechel and U. Hahn, "EmoBank: Studying the impact of annotation perspective and representation format on dimensional emotion analysis," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, Valencia, Spain: Association for Computational Linguistics, Apr. 2017.

[9] G. Caldarini, S. Jaf, and K. McGarry, "A literature survey of recent advances in chatbots," *Information*, vol. 13, no. 1, p. 41, 2022.

[10] S. Chaffar and D. Inkpen, "Using a heterogeneous dataset for emotion analysis in text," in *Canadian conference on artificial intelligence*, Springer, 2011, pp. 62–67.

[11] G. G. Chowdhury, "Natural language processing," *Annual Review of Information Science and Technology*, vol. 37, no. 1, pp. 51–89, 2003.

[12] H. W. Chung, L. Hou, S. Longpre, *et al.*, "Scaling instruction-finetuned language models," *arXiv preprint arXiv:2210.11416*, 2022.

[13] K. M. Colby, S. Weber, and F. D. Hilf, "Artificial paranoia," *Artificial Intelligence*, vol. 2, no. 1, pp. 1–25, 1971, ISSN: 0004-3702.

[14] H. Cuayáhuitl, D. Lee, S. Ryu, *et al.*, "Ensemble-based deep reinforcement learning for chatbots," *Neurocomputing*, vol. 366, pp. 118–130, 2019.

[15] B. Das and S. Chakraborty, "An improved text sentiment classification model using TF-IDF and next word negation," *CoRR*, vol. abs/1806.06407, 2018. arXiv: 1806.06407.

[16] D. DeVault, K. Georgila, R. Artstein, *et al.*, "Verbal indicators of psychological distress in interactive dialogue with a virtual human," in *Proceedings of the SIGDIAL 2013 Conference*, 2013, pp. 193–202.

[17] J. Devlin, M.-W. Chang, K. Lee, *et al.*, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186.

[18] P. Ekman, "Basic emotions," *Handbook of cognition and emotion*, vol. 98, no. 45-60, p. 16, 1999.

[19] P. Ekman and D. Cordaro, "What is meant by calling emotions basic," *Emotion review*, vol. 3, no. 4, pp. 364–370, 2011.

[20] D. Ghazi, D. Inkpen, and S. Szpakowicz, "Detecting emotion stimuli in emotion-bearing sentences," in *International Conference on Intelligent Text Processing and Computational Linguistics*, Springer, 2015, pp. 152–165.

[21] P. Goel, D. Kulshreshtha, P. Jain, *et al.*, "Prayas at emoint 2017: An ensemble of deep neural architectures for emotion intensity prediction in tweets," in *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, ACL, 2017, pp. 58–65.

[22] H. Gunes, B. Schuller, M. Pantic, *et al.*, "Emotion representation, analysis and synthesis in continuous space: A survey," in *2011 IEEE International Conference on Automatic Face & Gesture Recognition (FG)*, IEEE, 2011, pp. 827–834.

[23] C. Huang, A. Trabelsi, and O. Zaıane, "ANA at SemEval-2019 task 3: Contextual emotion detection in conversations through hierarchical LSTMs and BERT," in *Proceedings of the 13th International Workshop on Semantic Evaluation*, Minneapolis, Minnesota, USA: Association for Computational Linguistics, Jun. 2019, pp. 49–53.

[24] care innovations, *Www.careinnovations.org*, Care innovations, Aug. 2023.

[25] P. Irving and D. Dickson, "Empathy: Towards a conceptual framework for health professionals," *International Journal of Health Care Quality Assurance*, vol. 17, no. 4, pp. 212–220, 2004.

[26] W.-J. Ko, T.-y. Chen, Y. Huang, *et al.*, "Inquisitive question generation for high level text comprehension," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, Nov. 2020, pp. 6544–6555.

[27] T. Koulouri, R. D. Macredie, and D. Olakitan, "Chatbots to support young adults' mental health: An exploratory study of acceptability," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 12, no. 2, pp. 1–39, 2022.

[28] Y. Li, H. Su, X. Shen, *et al.*, "Dailydialog: A manually labelled multi-turn dialogue dataset," *arXiv preprint arXiv:1710.03957*, 2017.

[29] D. Liu, X. L. Feng, F. Ahmed, *et al.*, "Detecting and measuring depression on social media using a machine learning approach: Systematic review," *JMIR Mental Health*, vol. 9, no. 3, e27244, 2022.

[30] P. Liu, W. Yuan, J. Fu, *et al.*, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *arXiv preprint arXiv:2107.13586*, 2021.

[31] V. Liu, C. Banea, and R. Mihalcea, "Grounded emotions," in *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*, 2017, pp. 477–483.

[32] R. Maharjan, K. Doherty, D. A. Rohani, *et al.*, "Experiences of a speech-enabled conversational agent for the self-report of well-being among people living with affective disorders: An in-the-wild study," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 12, no. 2, pp. 1–29, 2022.

[33] K. McLaren, *The meaning of empathy and einfühlung*, Karla McLaren, Jan. 2013.

[34] A. Mehrabian, "Comparison of the pad and panas as models for describing emotions and for differentiating anxiety from depression," *Journal of psychopathology and behavioral assessment*, vol. 19, pp. 331–357, 1997.

[35] T. Mikolov, E. Grave, P. Bojanowski, *et al.*, "Advances in pre-training distributed word representations," in *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[36] T. Mikolov, I. Sutskever, K. Chen, *et al.*, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, 2013.

[37] S. Mohammad, F. Bravo-Marquez, M. Salameh, *et al.*, "Semeval-2018 task 1: Affect in tweets," in *Proceedings of the 12th international workshop on semantic evaluation*, 2018, pp. 1–17.

[38] S. M. Mohammad and F. Bravo-Marquez, "Wassa-2017 shared task on emotion intensity," *arXiv preprint arXiv:1708.03700*, 2017.

[39] J. M. Noble, A. Zamani, M. Gharaat, *et al.*, "Developing, implementing, and evaluating an artificial intelligence–guided mental health resource navigation chatbot for health care workers and their families during and following the covid-19 pandemic: Protocol for a cross-sectional study," *JMIR Research Protocols*, vol. 11, no. 7, e33717, 2022.

[40] K.-J. Oh, D. Lee, B. Ko, *et al.*, "A chatbot for psychiatric counseling in mental healthcare service based on emotional dialogue analysis and sentence generation," in *2017 18th IEEE international conference on mobile data management (MDM)*, IEEE, 2017, pp. 371–375.

[41] A. Paszke, S. Gross, F. Massa, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, Curran Associates, Inc., 2019, pp. 8024–8035.

[42] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[43] M. E. Peters, M. Neumann, M. Iyyer, *et al.*, "Deep contextualized word representations," in *Proceedings of NAACL-HLT*, 2018, pp. 2227–2237.

[44] R. Plutchik, "A general psychoevolutionary theory of emotion," in *Theories of emotion*, Elsevier, 1980, pp. 3–33.

[45] S. Poria, D. Hazarika, N. Majumder, *et al.*, "Meld: A multimodal multi-party dataset for emotion recognition in conversations," *arXiv preprint arXiv:1810.02508*, 2018.

[46] R. Pradhan, J. Shukla, and M. Bansal, "'k-bot'knowledge enabled personalized healthcare chatbot," in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, vol. 1116, 2021, p. 012 185.

[47] A. Radford, K. Narasimhan, T. Salimans, *et al.*, "Improving language understanding by generative pre-training," 2018.

[48] C. Raffel, N. Shazeer, A. Roberts, *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.

[49] M. M. Rahman, R. Amin, M. N. K. Liton, *et al.*, "Disha: An implementation of machine learning based bangla healthcare chatbot," in *2019 22nd International Conference on Computer and Information Technology (ICCIT)*, IEEE, 2019, pp. 1–6.

[50] A. B. Rakib, E. A. Rumky, A. J. Ashraf, *et al.*, "Mental healthcare chatbot using sequence-to-sequence learning and bilstm," in *International Conference on Brain Informatics*, Springer, 2021, pp. 378–387.

[51] S. Z. Razavi, L. K. Schubert, K. van Orden, *et al.*, "Discourse behavior of older adults interacting with a dialogue agent competent in multiple topics," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 12, no. 2, pp. 1–21, 2022.

[52] S. Z. Razavi, L. Schubert, M. Ali, *et al.*, "Managing casual spoken dialogue using flexible schemas, pattern transduction trees, and gist clauses," in *Annual Conference on Advances in Cognitive Systems (ACS)*, 2017.

[53] A. Roberts, H. W. Chung, A. Levskaya, *et al.*, "Scaling up models and data with t5x and seqio," *arXiv preprint arXiv:2203.17189*, 2022.

[54] B. Rothschild, *Help for the helper: The psychophysiology of compassion fatigue and vicarious trauma*. WW Norton & Company, 2006.

[55] L. Ruis, A. Khan, S. Biderman, *et al.*, "Large language models are not zero-shot communicators," *arXiv preprint arXiv:2210.14986*, 2022.

[56] J. A. Russell, "A circumplex model of affect.," *Journal of personality and social psychology*, vol. 39, no. 6, p. 1161, 1980.

[57] T. Sajed, "Building an expert-system based conversational agent to provide personalised resources about neurological disorders," M.S. thesis, University of Alberta, 2021. [Online]. Available: `https://era.library.ualberta.ca/items/03ba002e-0b00-479c-9b00-b12d84ac6b2c`.

[58] V. Sanh, A. Webson, C. Raffel, *et al.*, *Multitask prompted training enables zero-shot task generalization*, 2021. arXiv: `2110.08207 [cs.LG]`.

[59] J. Sawalha, M. Yousefnezhad, Z. Shah, *et al.*, "Detecting presence of ptsd using sentiment analysis from text data," *Frontiers in Psychiatry*, vol. 12, 2022, ISSN: 1664-0640.

[60] K. R. Scherer and H. G. Wallbott, "Evidence for universality and cultural variation of differential emotion response patterning.," *Journal of personality and social psychology*, vol. 66, no. 2, p. 310, 1994.

[61] S. Singh and H. Beniwal, "A survey on near-human conversational agents," *Journal of King Saud University-Computer and Information Sciences*, 2021.

[62] G. Stratou, S. Scherer, J. Gratch, *et al.*, "Automatic nonverbal behavior indicators of depression and ptsd: Exploring gender differences," in *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*, IEEE, 2013, pp. 147–152.

[63] Wadetb, *Wadetb/eliza: Python implementation of the eliza chatbot*. [Online]. Available: `https://github.com/wadetb/eliza`.

[64] R. S. Wallace, "The anatomy of a.l.i.c.e.," in *Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer*, R. Epstein, G. Roberts, and G. Beber, Eds. Dordrecht: Springer Netherlands, 2009, pp. 181–210, ISBN: 978-1-4020-6710-5.

[65] S. Wang, G. Peng, Z. Zheng, *et al.*, "Capturing emotion distribution for multimedia emotion tagging," *IEEE Transactions on Affective Computing*, vol. 12, no. 4, pp. 821–831, 2019.

[66] T. Wang, A. Roberts, D. Hesslow, *et al.*, "What language model architecture and pretraining objective work best for zero-shot generalization?" *arXiv preprint arXiv:2204.05832*, 2022.

[67] A. Warstadt, A. Singh, and S. R. Bowman, "Neural network acceptability judgments," *arXiv preprint arXiv:1805.12471*, 2018.

[68] J. Weizenbaum, "Eliza a computer program for the study of natural language communication between man and machine," *Communications of the ACM*, vol. 9, no. 1, pp. 36–45, 1966.

[69] H. Yang, A. Willis, A. De Roeck, *et al.*, "A hybrid model for automatic emotion recognition in suicide notes," *Biomedical informatics insights*, vol. 5, BII–S8948, 2012.

[70] H. Q. Yu, "Dynamic causality knowledge graph generation for supporting the chatbot healthcare system," in *Proceedings of the Future Technologies Conference*, Springer, 2020, pp. 30–45.

[71] A. Zamani, "Developing a mental health virtual assistance (chatbot) for healthcare workers and their families," M.S. thesis, University of Alberta, 2022. [Online]. Available: `https://era.library.ualberta.ca/items/1601a6f0-cf44-43b1-ba99-2ae7efbbcd9b`.