Integrating Conversational Pathways with a Chatbot Builder Platform

by

Varshini Prakash

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

Master of Science

Department of Computing Science

University of Alberta

Abstract

MIRA, a Mental Health Virtual Assistant, was developed during the COVID-19 pandemic to address rising mental health concerns and the demand for support and services. As a task-oriented chatbot, MIRA connected healthcare workers seeking mental health support with trusted services and programs. It achieved this by integrating Natural Language Processing (NLP) within a structured conversation flow, designed by experts in our multidisciplinary team to cater to healthcare workers. This work extends MIRA across diverse populations to improve accessibility to mental health services. The pilot version of MIRA was implemented in Python using an open-source chatbot development framework. Its conversational logic was hard-coded, and minor changes to the conversation flow necessitated extensive changes to the code base. Such a rigid system made it challenging to extend the functionalities of MIRA to cater to other languages or new populations, such as first responders, Veterans, Indigenous communities, or youth. In addition, building relevant and effective solutions that cater to community-specific needs requires insights from subject matter experts and close collaboration with a multidisciplinary team, including psychiatrists, domain experts and people with lived experience. Scaling MIRA's functionalities to serve diverse populations requires being able to easily integrate additional conversational pathways without changes at the code level. Hence, we introduce a chatbot builder platform and a conversation flow visualizer to simplify the creation and modification of conversational flows by non-technical experts. Our tools support multidisciplinary collaboration on

MIRA's development process by enabling iterative improvements and rapid modifications to the conversation flow. Our work has been successfully deployed and is available for public use `https://mymira.ca`.

Preface

Our paper, "Integrating Conversational Pathways with a Chatbot Builder Platform," is currently under review at the 26th International Conference on Information Integration and Web Intelligence (iiWAS2024) in Bratislava, Slovakia. The authors of the paper are Varshini Prakash, Alex Lambe Foster, Jasmine M. Noble and Osmar R. Zaiane.

To my parents, Prakash and Nithya,

For always being my guiding lights and encouraging me to explore.

The human spirit must prevail over technology.

– Albert Einstein

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Prof. Osmar Zaiane, for his tremendous support and guidance throughout my Master's journey. His encouragement gave me the confidence to overcome challenges that came my way. I am fortunate to have a mentor who is genuinely invested in the success of his students.

Thank you to the University of Alberta, Mood Disorders Society of Canada (MDSC), Mathematics of Information Technology and Complex Systems Accelerate, and Alberta Innovates for funding this research. I would also like to extend my heartfelt gratitude to my wonderful colleagues Alex Foster, Sandy Luu, Emilie Desnoyers, Dylan Merrick, and Matt Reeson for their invaluable contributions and support. I am deeply grateful to Ken Porter and Dr. Jasmine Noble, whose support made a huge difference in my journey.

I am grateful to Martha Steenstrup for organizing the communications workshop. Her teachings and feedback have greatly influenced how I articulate my ideas. I would like to thank Abhishek Naik for organizing the Making Minds Reading Group (MMRG) every month. Through MMRG, I had the opportunity to explore some wonderful books and engage in insightful discussions.

I am grateful for my friends Aristarchus Jim, Sheila Schoepp, Roberto Vega, AnaLi Vega, Farzane Aminmansour, and Katie Tran who turned grad school into a journey filled with cherished memories. Special thanks to my friends Sri Dharma, Suyash Pasari, Sarah Jean, and Elie Sarah, who stood

by me through the lows and celebrated my highs, even when separated by continents and time zones. Thanks to my childhood friend, Varsha Sridhar, for the lovely postcards from India that brought home closer to me.

My heartfelt gratitude goes to my father, mother, and sister, Shweta. Their unwavering faith in my dreams and constant encouragement to reach for the stars have been a driving force in my life. The kindness and warmth from my weekly conversations with my parents and grandparents made the distance from my family more bearable.

To Gautham, my partner and my biggest cheerleader, thank you for your endless love, kindness, and wisdom. I cannot thank you enough for all the insightful discussions, comprehensive feedback, and your relentless support. You are my rock, and I could not have done this without you.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Mental health awareness and access to mental health care services are crucial for building supportive environments and safe spaces that foster both individual and societal well-being. Poor mental health can have a substantial effect on various aspects of an individual's life, including their behaviour, relationships, and workplace performance. Recent studies reveal a concerning rise in mood disorders and mental health conditions [1]–[3]. The consequences can extend beyond individual well-being, affecting societal dynamics. If left unaddressed, individuals experiencing high levels of distress due to mental health conditions may be vulnerable to severe risks, including self-harm.

Increasing evidence shows that the COVID-19 pandemic led to a significant increase in the risks and burden of mental health illnesses [4]–[6]. The pandemic also exacerbated the adverse physical and psychological effects for those with pre-existing mental health conditions [7]. While this underscores an increased need for access to mental health services, several factors make it extremely difficult to seek help.

Some of the most common challenges in accessing mental health services are not knowing what resources exist, not knowing where to seek help, extended wait times, shortage of mental health providers, and financial constraints due

to inadequate coverage by insurance plans [8], [9]. Individuals struggling with poor mental health have reported they are reluctant to seek help due to the stigma attached to mental health, fear of judgement and potential negative impacts on their personal and professional lives [10]. Immediate access to information and support can be crucial in times of distress. It can be overwhelming for a person going through a crisis to find the right resources to seek help.

In this manuscript, we address the aforementioned challenges through the Mental Health Intelligent Information Resource Assistant (MIRA) chatbot. MIRA leverages the conversational capabilities of a chatbot to provide timely responses, making it a valuable tool for navigating mental health resources. Additionally, MIRA offers a private and anonymous platform, facilitating discussions on sensitive topics without fear of judgment or stigma.

1.2 MIRA

MIRA, the Mental Health Virtual Assistant, is an existing AI-based conversational agent that connects individuals seeking mental health support with trusted services and programs. Developed using the Rasa framework [11], the chatbot has been deployed[1] and is currently operational, launched publicly by our team and former members. As a task-oriented chatbot, MIRA interacts with the user through a conversation and presents them with relevant resources from the existing MIRA Resource Library - a comprehensive database of catalogued and expertly vetted resources.

MIRA uses a hybrid approach combining Machine Learning (ML) and Natural Language Processing (NLP) techniques with a rule-based conversation flow. We leverage NLP to detect intents (general meaning of the sentence) and extract entities (additional information in the sentence that provides con-

[1] `https://mymira.ca`

text) from the user's utterances.

MIRA was initially designed to cater exclusively to healthcare workers. We are now expanding its functionalities to serve equity-deserving populations and the general public. Generic mental health solutions are often ineffective because each community has unique needs and challenges that require tailored solutions. The obstacles faced by equity-deserving groups can be shaped by their history and circumstances. Experts who have studied these communities or belong to them provide valuable insights for designing effective interventions. With support from experts, we tailor MIRA to assist the general public, first responders, veterans, Indigenous communities, children and youth, and French-speaking Canadians. This approach ensures that MIRA meets the specific needs of diverse groups, providing effective and accessible mental health support.

Extending the pilot version of MIRA proved to be challenging due to its reliance on a rigid and hard-coded conversation flow structure. Altering this flow required direct changes to the underlying codebase. To make it easier to extend MIRA's functionalities and to include non-computer science experts in our team in the development and design process for MIRA, we created a chatbot builder platform. The chatbot builder simplifies the creation and modification of conversational flows. This meant that we could integrate additional conversational pathways without any changes to the code.

The conversational flow is segmented into modules on the platform, each starting with a chatbot-initiated utterance and linking to connected modules or user pathways based on conditions. The conversation flow can be visualized as a directed graph where each node or module defines the subsequent path based on specific conditions, with transitions or jumps between modules. The chatbot builder enables individual members of our multidisciplinary team to adapt and extend a chatbot's functionality to support general audiences or

particular groups of interest. Notably, while initially designed for MIRA, the platform may also be helpful for other Rasa-based chatbots that require rapid iterations and have continuously evolving needs.

1.2.1 MIRA as a tool

Our study aims to complement and augment traditional mental healthcare with a conversational guidance system. While Artificial Intelligence (AI) and digital mental health tools are not a replacement for traditional face-to-face care from mental health professionals, they can serve as valuable tools. MIRA is not intended to replace traditional therapy or counselling sessions with mental health professionals. It is a supplementary tool that can assist with resource navigation and mental health literacy. MIRA does not provide medical diagnoses or clinical assessments. It relies on user inputs and preferences to recommend appropriate resources.

MIRA is not a standalone solution for mental health care. Rather, it is designed to be a tool that complements other mental health services, resources, and support systems. Emerging mental health technologies can help bridge gaps in care, provide immediate support, and increase the overall accessibility of mental health resources, thus offering crucial assistance to those in need.

1.3 Thesis Statement

Our research focuses on developing a conversational guidance system to connect individuals seeking assistance with relevant mental health resources. We aim to extend the pilot version of MIRA, designed for healthcare workers, to the general public and other equity-deserving populations in order to improve accessibility to mental health services and resources.

To achieve this, we try to address the following research questions:

- How to make transitions between conversation modules reliable?

- How can we decouple the conversation flow from the codebase to extend to other audience groups without any changes to the code?
- How to scale MIRA to serve multiple audiences reliably without jeopardizing the existing functionalities?
- How can we extend MIRA to support bilingual capabilities in English and French?
- How can we visually represent complex conversation flows to make the underlying logic intuitive?

Addressing these research questions will enable our team of subject matter experts to seamlessly create new conversation flows and update existing ones.

1.4 Thesis Contributions

The initial version of MIRA was pivotal, as it laid the groundwork for the subsequent iterations. This included establishing the foundations for MIRA using the Rasa framework, creating a portal for cataloguing and annotating the database of resources, constructing conversational flows using Lucidchart (a flow chart tool) [12], meticulously curating extensive training data for each intent, exploring various architectures for intent detection and entity extraction and developing strategies for data augmentation. While these efforts provided a strong foundation for MIRA's functionality, they also revealed certain limitations that required further improvement.

MIRA's functionality was originally designed to cater to the needs of healthcare workers. The system employed a rigid, hard-coded conversation flow structure engineered to gather enough information from the user to formulate a query and retrieve relevant resources from the MIRA resource library. This hard-coded conversation flow catered to a specific audience. Expanding

MIRA to new audiences required significant time and effort to reconfigure the conversation flow. Altering this flow required direct changes to the underlying codebase and modifying custom actions in the Rasa framework. Firstly, the transitions or jumps between modules in the pilot version were occasionally unreliable, leading to errors and interruptions in the user experience. Secondly, extending MIRA to serve other audiences added challenges. The transitions between different conversational modules or segments limited flexibility and adaptability.

Scaling MIRA or modifying the existing flow became infeasible without a flexible way to dynamically modify the conversation flow. We subsequently had to restructure MIRA's fundamental architecture to enable these dynamic jumps. Therefore, our objective was to make the transitions reliable to establish a more agile and scalable system capable of rapid modifications, module redirections and dynamic flow adjustments that did not require extensive changes to the codebase. We subsequently had to restructure MIRA's fundamental architecture to enable these dynamic jumps.

To address these challenges, we abstracted the conversation flow and generalized the jumps in between the modules. This was crucial to enable seamless transitions between different parts of the dialogue. Since the jumps were now generalizable and reliable, we needed a tool that would allow us to modify the conversation flow more efficiently. The tool led to the development of a chatbot builder, which served as the foundation for our subsequent goals.

The chatbot builder provided a user-friendly interface for designing, updating and managing the conversation flow. It allows the team to make changes to the dialogue structures without the need for extensive coding. This tool enables our team to iterate quickly in response to evolving needs and to promptly address any bugs or issues that arise. Furthermore, as the complexity of the conversation flow increased, we identified the need for a visual representation

of the conversation structure. To address this, we implemented an automatic visualization of the conversation flow within the chatbot builder. This made it easier for the collaborators to visualize the structure and logic of the dialogue.

Using the chatbot builder and the flow visualizer, we expanded the scope of MIRA beyond healthcare workers to include the general audience, youth, French-speaking populations and indigenous communities. Thus, broadening its accessibility and impact.

1.5 Thesis Outline

The following chapters of this manuscript include details about various aspects of our study. Chapter 2 discusses the mental health landscape in Canada, digital mental health tools, current developments in chatbots, and existing chatbot development platforms. Chapter 3 provides details about the implementation of MIRA, while Chapter 4 discusses the implementation of Chatbot Builder and the Flow Visualizer. In Chapter 5, we discuss the impact of the Chatbot builder on MIRA. Finally, Chapter 6 discusses the conclusions from this study and explores potential directions for the future of MIRA.

Chapter 2

Background

2.1 Mental Health Challenges in Canada

The prevalence of mental health concerns is striking, with statistics indicating that by the age of 40, one in two Canadians will have experienced or will be actively experiencing a mental illness [13]. A 2022 study conducted by Statistics Canada reported that over 5 million Canadians were affected by mental health conditions in the previous year [14]. The study also revealed that more than 36.6% of individuals with mood, anxiety, or substance use disorders reported that their health and mental health care needs were unmet or only partially met [14].

The social distancing and lockdown measures implemented during the COVID-19 pandemic intensified feelings of loneliness and social isolation, especially among vulnerable groups—those facing financial strain and struggling with poor mental health [15]. A meta-review of the longitudinal mental health impacts of the pandemic found significantly higher rates of depression and anxiety compared to pre-pandemic levels [16]. The study also indicated that adolescents, pregnant or postpartum people, and those hospitalized with COVID-19 dealt with worsening mental health [16]. These studies indicate the adverse effects of the pandemic on our long-term mental and physical well-being.

Concerningly, a more recent poll from Mental Health Research Canada (MHRC), conducted in January 2024, reveals the worst self-rated mental health indicators since the pandemic [17]. Over 10% of Canadians, particularly in Alberta and Ontario, reported high levels of anxiety or depression [17]. The results also showed a significant correlation between increased personal screen time and negative mental health indicators, with 26% of young Canadians spending 6+ hours on screens every day [17]. Multiple studies have provided evidence that excessive or problematic social media use is positively correlation with poorer overall mental health outcomes [18], [19]. These findings underscore the urgent need for improved mental health support amidst the aggravating effects of the pandemic and the pervasive use of social media.

2.1.1 Community-Centric Mental Health Support

With an increasing number of individuals struggling with mental health risks and rising demand for mental health care services, we address populations that are particularly vulnerable or identified as experiencing unmet needs in accessing care. Mental health challenges are complex and often unique to a particular population or community, based on their specific circumstances and histories. Effective solutions to support these populations are those that take their needs into consideration and create tailored pathways that address their distinct needs. Therefore, to improve accessibility for these groups, we develop extensions that can address the needs of first responders, Veterans, Indigenous communities, children and youth, and French-speaking Canadians.

French holds significance in Canada as one of the country's two official languages. Statistics on official languages in Canada reveal that French is the first official spoken language for 22% of the population, while 18% of the population is bilingual in both English and French [20]. These figures express the

French language's widespread use and cultural significance within Canada. A recent study in Quebec, the Canadian province that is primarily Francophone, reported a high level of depressive symptoms among respondents [21]. The survey also found that young adults (18-24 years) reported the highest prevalence of moderate to high depressive symptoms at 38% [21].

Youth (ages 15-24) are more likely to experience mental illness and substance use disorders than any other age group [22]. The youth, especially women, were more susceptible to mood and anxiety disorders, according to a recent study. In 2019, 5% of Canadian children between the ages 5-17 reported experiencing an anxiety disorder, while 2.1% reported experiencing a mood disorder [22].

Indigenous people in Canada are disproportionately affected by mental health challenges. This inequity results directly from the country's colonial past, persisting neo-colonial policies, and racism [23], [24]. A review of over two hundred papers on the mental health of Indigenous peoples in Canada stated that significant gaps remain in the research [25]. They found that suicide and substance use were overemphasized in the literature while lacking a critical approach to addressing historical trauma and colonialism [25]. Additionally, several groups like Métis and urban or off-reserve Indigenous communities were underrepresented in the research [25].

2.1.2 Digital Mental Health Interventions (DMHIs)

DMHIs aim to augment traditional care by improving access to mental health services and resources by adapting these interventions into digital formats [26]. They can encompass a range of applications, such as smartphone applications, web-based programs, online platforms, chatbots, virtual reality simulations, wearable devices or online support communities [27].

A key advantage of DMHIs is their scalability, allowing them to deliver effective research-backed treatments to a wider population [28]. DMHIs can substantially improve the accessibility of mental health treatments by addressing many of the barriers that prevent individuals from seeking help. Emerging technologies in mental health can be used to combat traditional barriers like staff shortages, cost and stigma [28]. An understaffed healthcare system leads to tightened workload, stress, and burnout among healthcare professionals [29].

Transportation challenges combined with limited clinic hours can hinder accessibility for patients from rural areas, the elderly, and individuals with disabilities [30]. For individuals with depression, DMHIs may be particularly useful as a home-based solution [30]. Symptoms of depression like fatigue, low energy and lack of motivation can make it difficult to travel for traditional treatment [30].

The COVID-19 pandemic led to a surge in demand for digital mental health resources due to their accessibility, cost-effectiveness, and personalization. Chatbots are widely used in DMHIs within the mental healthcare domain.

2.2 Chatbots

Chatbots are conversational agents that interact and engage with users through natural language conversations, either over text or voice-based interfaces, to perform various tasks or provide assistance. They have become ubiquitous in recent times and have become widely popular for their convenience and utility, owing to their versatility, scalability, and ability for customization.

There exists a broad range of chatbots, from customer service assistants that try to streamline customer interactions and provide customized support to voice-based assistants like Siri[31], and Alexa[32] to more cutting-edge innovations like ChatGPT[33], capable of engaging in more human-like conversations

with users.

Typically, chatbots fall into three categories: those engaging in open-ended conversations or task-oriented interactions [34], or a combination of both [35]. Rule-based chatbots rely on pre-defined rules, responses, and actions [36].

2.2.1 Generative Chatbots

Advanced chatbots employ generative models, creating natural real-time dialogues while executing tasks. Large Language Model (LLM) based chatbots like ChatGPT[33], Copilot[37], and Perplexity AI[38] excel in real-world scenarios given their ability to deal with noisy input and execute human instructions well. LLMs are fine-tuned using Reinforcement Learning from Human Feedback (RLHF) [39], a technique that leverages human feedback to refine the LLM model through reinforcement learning, optimizing it to make predictions that maximize reward. There have been rapid advancements in the field of LLMs, including breakthroughs such as multi-modal capabilities [40] and multi-agent collaboration [41], to name a few. Researchers have developed LLMs in the fields of medicine [42], finance [43], and programming [44].

While LLM research is impressive and LLM-based chatbots have surged in popularity in recent years, they have limitations. A notable concern is their potential to generate biased responses if the data used to train these LLMs is biased [45]. This can lead to discriminatory and harmful behaviour, posing ethical challenges. They also tend to hallucinate to fill in gaps in their knowledge. Hallucinations are bugs in LLMs that occur when they generate factually incorrect or purely fictional information to fill gaps in their knowledge [46]. This can lead to erroneous decisions, spread misinformation, and reduce trust in AI. Furthermore, as these models grow in size and complexity, characterized by billions of parameters trained on massive amounts of data, their internal workings become increasingly opaque [45].

This makes it challenging to incorporate LLMs in domains like healthcare, where the accountability and transparency of decisions are crucial. Considering the sensitive nature of mental health, these limitations have the potential to be dangerous. With caution, we plan to explore using generative chatbots in the future once we enforce robust guardrails to ensure their reliability and safety.

2.2.2 Conversational Information Retrieval Chatbots

Conversational information retrieval chatbots assist users in navigating large archives of information or extensive databases, retrieving relevant answers to user queries using search algorithms. Their application as FAQ chatbots is particularly popular, where they simplify and streamline the time-consuming task of sifting through information to find relevant content [47]. Through immediate responses and accurate search results, IR chatbots can increase efficiency for customers by facilitating informed decision-making and increased customer satisfaction.

In retrieval tasks, user questions are matched with large archives of text using an inverted index [48]. The widely used method to retrieve information is by evaluating the similarity between a user's query (q) and an existing question (Q) or a question-answer (QA) pair in the database [49]. This evaluation employs various similarity measures, such as Term Frequency–Inverse Document Frequency (TF-IDF) [50], cosine similarity [51] or semantic embeddings [52]. TF-IDF measures the importance of a term within a document relative to a collection of documents.

While traditional information retrieval chatbots that rely on similarity measures to match answers from an existing database are useful, they struggle with understanding extended dialogues and maintaining conversational con-

text. In contrast, more recent chatbots incorporate deep learning techniques to interpret user intents, perform pattern matching, apply relevance feedback to refine search results, and use query expansion to capture the broader context of user queries [53], [54]. Using similarity metrics derived from transformer-based embeddings has led to more accurate and context-aware matches [55], [56].

Conversational information retrieval is possible through LLMs. However, as discussed earlier, they have the drawback of hallucinating. One technique that has been introduced to mitigate hallucinations is Retrieval-Augmented Generation (RAG), an approach that integrates an information retrieval task prior to LLM's generation task. RAG retrieves documents relevant and augments them as additional context into the prompt [57]. This method provides the LLM with external knowledge beyond its original training data, helping to fill gaps and reducing hallucinations, especially for knowledge-intensive queries [57]. While RAG can reduce hallucinations, it does not eliminate them. The compiled information relies heavily on the quality of the database.

2.2.3 Task-oriented Chatbots

Task-oriented chatbots assist users in performing domain-specific tasks, in contrast to open-domain chatbots. These chatbots generally employ a structured conversation flow using predefined dialogue management rules [58], [59]. The rule-based flow is an effective solution to steer the conversation within the boundaries of the intended task or domain and ensure a focused interaction. There is evidence that rule-based guidance generally leads to improvement in task performance for such chatbots [60]–[62].

These chatbots incorporate natural language understanding through intent detection and entity extraction. They typically leverage slot-filling techniques to prompt the user for specific information required to complete the task [63].

Another key feature of task-oriented chatbots is the integration with external systems or databases to complete domain-specific actions such as providing customer support, booking reservations, placing orders, or scheduling appointments [63].

An example of a task-oriented chatbot developed for survivors of sexual violence uses a hybrid approach, combining rule-based and ML-based approaches [64]. Another example of a task-oriented chatbot named *CHAI*, combines offline Reinforcement Learning (RL) with fine-tuned language models to interact with users in a goal-directed dialogue [65]. They evaluate CHAI on a negotiation task where the bot assumes the role of a seller and negotiates with the user, who assumes the role of a buyer [65]. A domain-specific and focused interaction style, combined with natural language capabilities makes task-oriented chatbots intelligent within their specialized domain. They also enhance the user experience by making task completion more convenient and efficient.

While task-oriented chatbots can offer a focused user experience, they are known to have difficulties in handling natural language and struggle to comprehend the complexity and variability that is inherent to human language [66]. Inadequate understanding of natural language can have adverse consequences resulting in recognition errors and breaks in the conversations [67]. Numerous studies have shown that such conversation breakdowns can lead to user frustration and abandonment of the service [34], [68]. This work tries to address this limitation through the development of a framework to transition conversations from open-domain chatting to task-oriented interactions [69]. They also released a large-scale annotated dataset to support research in this direction [69].

Another study that proposes solutions to this challenge, analyzed the natural use of a task-oriented banking chatbot to gain interesting insights into how

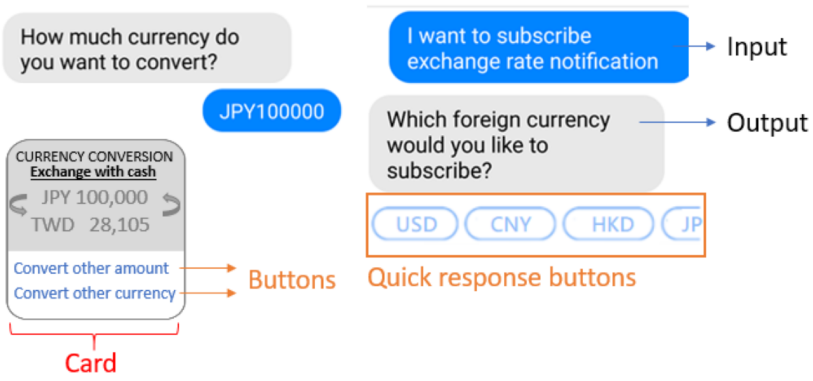


Figure 2.1: Original figure from a study[34] suggests that offering alternative input modalities, such as buttons or quick responses, in addition to typing, resulted in more satisfactory user interactions for task-oriented chatbots.

users navigated obstacles in the conversation or conversational non-progress (NP) during their interactions [34]. In their analysis of conversation logs from over a thousand users, they found that users were more likely to abandon the chatbot altogether in the event of NP, rather than opting to switch subjects or reformulate their message. [34]. They found that users may express themselves more effectively when they were offered options and alternate channels [34]. Interestingly, providing options such as buttons, shown in Fig. 2.1 proved to be effective and reduced incidences of NP [34]. This aligns with the findings from another study which suggests that that providing guesses of user intentions and offering possible options could make it explicit to the users that the chatbot was trying to recognize and understand their intent [68]. Additionally, users were less likely to leave a conversation when presented with alternative channels [34]. Exiting the service because they have been directed to a better alternative is a much more favourable outcome than due to a negative experience [34].

In summary, enhancing natural language understanding capabilities through advanced techniques like contextual reasoning, intent recognition, error handling, and effective fallback mechanisms is crucial for task-oriented chatbots

to provide a seamless and satisfactory user experience, especially in the face of natural language complexities and deviations from the predefined conversational flow.

2.3 Existing Chatbot Builders

As chatbots became ubiquitous, there has been a surge in chatbot builders, more commonly known as Chatbot Development Platforms (CDP). CDPs are specialized tools used to create and develop various types of chatbots. These platforms provide a comprehensive set of features that allow users to build chatbots without needing extensive programming skills, typically through a user-friendly graphical user interface (GUI). The commonly used CDPs are proprietary ones developed by technology giants, including *Dialogflow* by (Google), *Microsoft Bot framework*, *Amazon Lex* and *IBM Watson*. Open-source frameworks like *Rasa*, *Botkit*, *Botpress*, and *DeepPavlov* offer alternatives to the aforementioned proprietary solutions.

CDPs help reduce the complexity of the chatbot development process for individuals and businesses. This accessibility is particularly beneficial for non-coders, newer developers or interns, and new team members, allowing them to contribute to the chatbot development process and foster collaboration. Additionally, these platforms may provide functionalities that assist with deploying, maintaining, and scaling chatbots. This allows developers to focus solely on the development of the chatbot logic rather than the underlying infrastructure. Their biggest advantages are that they enable faster development cycles and significantly help reduce the development costs that come with building chatbots from scratch.

To select the right CDP for an objective, it is important to ensure the platform's capabilities align with the specific needs and requirements of the project. Academics have outlined key aspects of effective chatbot design, in-

cluding functional capabilities such as integration, analytics and quality assurance [70], [71]. There are several blogs that compare and highlight the strengths of various CDPs [72], [73].

A previous study evaluated existing CDPs, including Amazon LEX, Dialogflow CX/ES, LUIS, Conversational language understanding (CLU) by Microsoft, Rasa, and IBM Watson Assistant to help users select the most suitable CDP for their needs [71]. They specified the criteria and assigned scores based on requirement assumptions. This study offered valuable insights relevant to our work. They found that the more recent versions of CDPs aimed to facilitate chatbot development without the requirement of coding. For example, Dialogflow introduced the CX edition featuring a visualized state machine known as the visual builder, while Microsoft developed Bot Framework Composer to build conversational AI applications [71]. In contrast, Rasa does not support any visual dialogue management flow but offers stories, a text-based tool that replaces visual managers for designing dialogue flows [71]. Despite this limitation, they claim Rasa is the most configurable and extensible among the CDPs due to its open-source nature. Additionally, Rasa's NLU implementation is transparent, allowing developers to make informed decisions about their choice of ML techniques [71]. While other commercial CDPs do not offer this level of transparency.

Another study analyzed three prominent CDPs: Google Dialogflow, IBM Watson Assistant, and Amazon LEX [74]. This work identifies a comprehensive set of desirable features and evaluates the utility of various chatbot development platforms. By comparing these platforms, the study provides insights to inform decisions on selecting the most suitable chatbot builder for specific use cases and requirements [74].

While these CDPs are extremely useful tools, they are not without limitations. A major challenge in using CDPs is the risk of vendor lock-in, where

Frameworks	Open-source	Flexibility	Cost-effective	Ease of Integration	Transparency	Community support
Dialogflow						
Amazon Lex						
IBM Watson						
Microsoft Bot Framework	✓	✓				
Rasa	✓	✓	✓	✓	✓	✓
Botkit	✓	✓		✓		
DeepPavlov	✓	✓		✓		

Table 2.1: Comparing features across Open-Source and Proprietary Chatbot Frameworks

platform-based approaches bind chatbots to a vendor's ecosystem [75]. This limits the flexibility in adopting new technologies or integrating with platforms that may better align with the chatbot's needs [75]. Vendor lock-in is often implicit and can pose difficulties in transitioning out of the ecosystem. For instance, Watson offers a separate pricing plan for deploying chatbots on third-party clouds, and currently, there are no migration tools that allow neutral exchange formats between platforms [75].

Additionally, while these platforms offer informal guidelines for chatbot design, there is a lack of established design patterns and standardized quality metrics tailored for chatbots [75]. In the future, these platforms should develop comprehensive guidelines and integrate design principles to standardize best practices for chatbot development [75].

As an effort against vendor lock-in, the World Wide Web Consortium (W3C)'s Conversational Interfaces Community Group strives to unify dialogue systems across domains and proprietary formats [76]. It enables developers to collaborate and share experiences across various domains, integrating proven techniques from existing specifications and design standards [76].

2.3.1 Mental Health Chatbots

A 2020 study found that 39% of health chatbots were mental health chatbots [77]. This number continues to grow over time. They have been integrated to serve users with functions ranging from providing support, conducting screen-

ing, delivering therapeutic intervention, monitoring behaviour changes, and preventing relapses [78]. Efforts have been made to develop specialized chatbots to support individuals with specific mental health concerns such as depression and anxiety [79], autism [80], Attention-deficit/hyperactivity disorder (ADHD) [81], Post-traumatic stress disorder (PTSD), suicide prevention [82], mindfulness [83] and life coaching [84].

Previous research indicates that chatbots can improve mental health outcomes, with users reporting their experience with mental health chatbots mostly positively [85], [86]. This is supported by the ability of chatbots to offer increased accountability, such as daily check-ins [87], and behavioural intentions [88], promoting higher user engagement and lower dropout rates [89]. Towards improving care access, mental health chatbots have the benefit of being accessible round-the-clock and the ability to cater to the needs of users irrespective of their time zone or physical location [90]. Moreover, evidence suggests that the anonymity offered by chatbots, coupled with the absence of direct human interaction, allows users to express themselves more freely without the fear of judgment or stigma [91].

The findings from a comprehensive scoping review that evaluated 41 mental health chatbots indicate that the primary applications of chatbots in the mental health domain are for the delivery of therapy, followed by their use as training and screening tools [92]. Notably, the study revealed that a significant majority of these chatbots (92.5%) employed rule-based architectures, while only 7.5% were ML-based [92]. The study suggests the preference for rule-based frameworks is due to their ability to perform simple, well-defined, and structured tasks [93]. Additionally, rule-based chatbots are simpler to develop and prone to fewer errors.

2.4 Takeaways

There has been a notable increase in demand for mental health support and a decline in mental well-being, exacerbated by the pandemic. To address this, we must ensure that mental health services are accessible to a wide range of individuals by developing inclusive solutions that cater to community-specific needs. Through MIRA, we ensure our approach remains transparent, inclusive, and culturally relevant to the audiences we cater to.

Therefore, subject matter experts specializing in youth, French, and Indigenous research oversee their respective extensions for MIRA. Indigenous researchers and advisors control and lead their extension. They are actively involved in community engagement to incorporate the voices of Indigenous communities into the evaluation and development stages.

To build solutions, we explore existing DHMIs, particularly chatbots, given their widespread use within the mental health domain. Subsequently, we discuss generative chatbots, information retrieval and task-oriented chatbots in detail. We draw insights from existing literature to design MIRA with a hybrid approach. This combines the following:

1. Rule-based approach to maintain a structured conversation flow and leverage our team's expertise in psychiatry for guidance

2. Task-oriented design to focus solely on mental health topics rather than open-ended dialogue

3. Information-retrieval methods to fetch relevant resources using extracted entities that match tags in the database

Additionally, we incorporate solutions from existing work to prevent conversational breaks by including fallback rules, communicating alternative channels to users, such as directing them to the MIRA resource library and offering

buttons to guide users through resource selection to enhance engagement and user experience.

The focus of this manuscript is to develop a chatbot builder for MIRA, a chatbot built using the Rasa framework. A chatbot builder facilitates the integration of new conversational pathways for non-computer science experts, allowing easy extension of existing chatbot frameworks. The platform simplifies the incorporation of insights for extensions to specific groups without rebuilding the entire system. Therefore, we developed a chatbot builder with a user-friendly interface for designing, updating, and managing the conversation flow. To achieve this, we evaluate existing chatbot builders, both proprietary and open-source solutions.

Commercial CDPs such as Dialogflow [94], Amazon Lex [95], and IBM Watson[96] have drawbacks such as cost inefficiency, high software and maintenance expenses, and limited flexibility in customizing platform features to align with our needs. While Microsoft Bot Framework [97] is an open-source solution, it still stores data and individual user interactions on the cloud, a practice also common for the platforms mentioned above. We chose Rasa due to its flexibility, ease of integration, community support and high level of customization compared to other open-source platforms like Botkit [98] and DeepPavlov [99].

We extend MIRA to communities mentioned in Section 2.1.1 using our chatbot builder. However, it is a versatile tool that can be utilized beyond MIRA to design conversation flows for other Rasa-based chatbots.

Chapter 3

Implementation of the Pilot Version of MIRA

3.1 Overview

This chapter outlines the implementation details of the pilot version of MIRA and its components [100], [101]. The chapter is structured as follows: Section 3.2 outlines the resource library and the resource review process, Section 3.4 explains our reasons for choosing Rasa over other chatbot frameworks, Section 3.4.1 provides a detailed overview of the Rasa architecture and its components, including how they work together. Section 3.4.2 covers intent detection and entity extraction, while Sections 3.4.3 and 3.4.4 discuss the training data and the dialogue management system. Finally, we address the limitations of the pilot version of MIRA in Section 3.5 and propose solutions to overcome them in Chapter 4, which constitutes the main contribution of this manuscript.

3.2 Resource Library

MIRA Resource Library is a database of resources, with over 1500 resources catalogued by members of the Mood Disorders Society of Canada (MDSC). To ensure the quality and reliability of the resources, they are vetted and approved by an Expert Advisory Committee (EAC) using an evaluation assessment form developed using a hybrid of validated tools [102]. EAC members are subject

Figure 3.1: This figure outlines the resource gathering and review process. Research associates and volunteers catalogue resources with appropriate tags, which the EAC then reviews.

Resource Category	Approved Resources	Resources Logged
Indigenous	148	197
Youth	497	590
French	231	322
Healthcare Workers	106	155
Veterans	69	81
General Public	123	200
Total	1174	1345

Table 3.1: This table lists the approved resources and the total number of resources logged among specific resource categories as of May 2024.

matter experts from MDSC's network of partners. They assist with tasks such as beta testing MIRA and its new features, reviewing resources in the MIRA Resource Library, and providing advice on related topics. At least two EAC members vet each resource before approval, as illustrated in Fig. 3.1. The resource library currently has over 1000 fully vetted resources. The resource gathering and review process follows the steps depicted in Fig 3.1. At the end of an interaction with MIRA, the user is recommended resources from the MIRA resource library relevant to the context of the conversation.

3.3 Components of MIRA

MIRA's chat interface is a web-based GUI that connects to the Rasa-based MIRA using a Representational State Transfer Application Programming In-

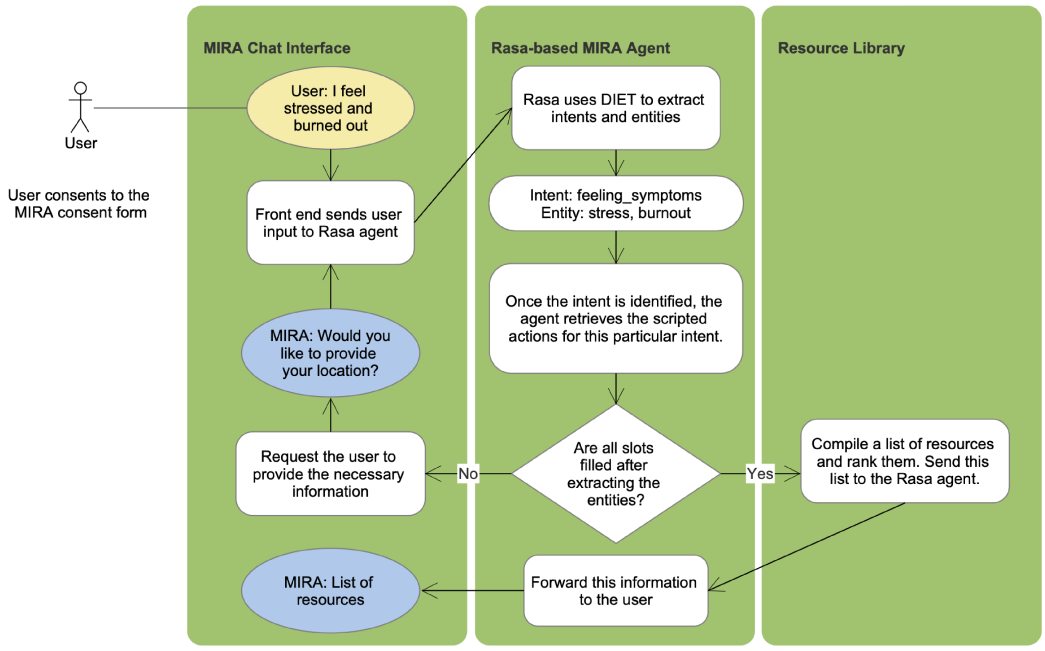


Figure 3.2: Activity diagram illustrating the collaboration between the user, MIRA Chat Interface, Rasa-based MIRA, and Resource Library to find a resource for the user.

terface (REST API). REST API transfers a representation of the requested resource's state to the client system. The Rasa-based agent is trained on the Dual Intent and Entity Transformer (DIET) architecture to perform intent detection and entity extraction tasks simultaneously. Section 3.4.2 explains the DIET architecture in detail.

This activity diagram in Fig. 3.2 illustrates the interactions between various components of the pilot version of MIRA in the process of finding a resource for a user. These components include the user, MIRA Chat Interface, Rasa-based MIRA agent, and the Resource Library.

The process begins with the user consenting to the MIRA consent form. The user's utterances are sent to the Rasa-based agent. Subsequently, the Rasa framework uses the DIET model [103] to detect the user's intent and extract relevant entities from the input. The system determines the most probable user intent based on the detected intent and retrieves a response per

the rule-based flow. The chatbot might request additional information from the user, such as their location.

The system verifies whether all the required slots have the necessary information. If they are, the system queries the resource library. The resource library ranks resources based on the extracted entities and intents and sends a list of resources back to the Rasa agent. Finally, the compiled response, including the list of relevant resources, is sent back to the chat interface, and the user receives the final list of resources.

3.4 Rasa Framework

Rasa is an open-source conversational AI framework used to build chatbots [104]. Rasa's customization capabilities allowed the integration of advanced AI and NLP functionalities incrementally, fitting our project objectives [102]. This approach supported the ongoing development and tests to continually enhance the chatbot's capabilities and adapt to evolving needs.

Considering the importance of privacy and security, particularly in the context of a mental health care chatbot such as MIRA, we are hesitant to entrust our data to cloud storage. Therefore, we run Rasa on our local machines to train and deploy models on the University of Alberta servers. Moreover, since MIRA is a free service available to the public, affordability was taken into account and free platforms were prioritized. Therefore, Rasa became an affordable solution that offered control over software flexibility and aligned with our data security and privacy needs.

3.4.1 Components

Rasa follows a scalable and modular architecture depicted in Fig 3.3. At the core of MIRA is a Rasa-based agent responsible for key functionalities, including training the model, handling messages, loading the dialogue model,

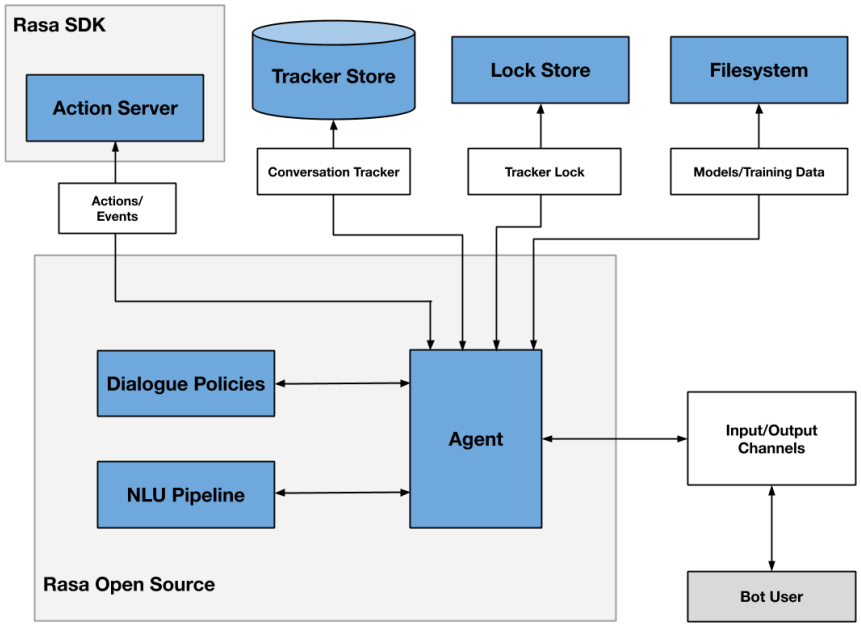


Figure 3.3: Figure from Rasa[105] depicts an overview of Rasa's internal architecture, which consists primarily of NLU and dialogue management. Dialogue Policies select the next action in a conversation, and NLU processes user inputs to extract structured information.

and fetching the next action. Below is a list of key concepts and components within the Rasa pipeline, with its two primary components being Natural Language Understanding (NLU) and Dialogue Management.

- **Action**: A single activity or function that the chatbot performs in a conversation; this can be responding to the user or sending an API call.
- **Rasa Software Development Kit (SDK)**: A Python SDK used to develop custom actions.
- **Custom Actions**: Custom actions can be programmed to implement desired tasks for the chatbot. We typically use custom actions to interact with external systems and APIs. For a generic chatbot, this can include everything from API calls, database queries, responses to the user, controlling lights, adding a reminder, checking back balances, and more.

- **Action Server**: Action Server is within the Rasa SDK and separate from Rasa Open Source, and is responsible for executing these custom actions.

- **Rasa Open Source**: Previously, Rasa Core and Rasa NLU were two separate components. The two are now integrated into a single package called Rasa Open Source.

- **Dialogue Management**: The dialogue management system uses policies to determine the appropriate action at each step of a conversation. Based on the context, the dialogue management component selects the next course of action. In the pilot version of MIRA, this system followed a handcrafted conversation flow structure, guiding the user through predefined paths.

- **NLU**: This component is responsible for extracting structured information from user messages through tasks such as intent detection, entity extraction, and response retrieval. We employed a pre-trained DIET model to process user messages. We also used regular expressions and lookup tables in our training data to ensure the DIET model identifies intents and entities accurately.

- **Intent**: Intent refers to the underlying objective in the user's message. The goal of intent detection is to accurately identify the intention within user's messages, which could range from seeking information, needing assistance, and, requesting information about treatment or symptoms. Accurately identifying these intents is crucial for a positive user experience.

- **Entity**: Entities are structured information within user messages. They are annotated in training data to process specific information from the

user input (slot filling). We use lookup tables, a list of pre-defined values to extract entities with known values from user inputs. For instance, consider the entity `mental_health_concern`. This entity in included in our training data as a comprehensive list of mental health conditions.

- **Training Data**: Composed of a list of utterances annotated with intents and entities. We specify the training data in markdown format, easy to read and is curated by members of our team on Github.

For MIRA, we employ a rule-based policy that follows the conversation flow developed using our chatbot builder. We collect the necessary information from users through entities using the slot-filling process to serve resources. Slot mappings are defined for each slot that our form needs to complete, and the slot values are determined by detected entities. Additionally, our training data samples include these entities to ensure accurate entity detection.

3.4.2 Intent Detection and Entity Extraction

The chatbot employs Rasa's Dual Intent and Entity Transformer (DIET) architecture for both intent detection and entity extraction tasks [103]. The intents and entities extracted with the DIET classifier are used to trigger various functionalities, such as replying with a predefined message, writing to a database, or accessing an API to complete the user's request. This allows MIRA to both easily drive the conversation based on user queries, and to extract relevant mental health concerns it can later use to serve the user resources. A visual representation of an interaction between a user and MIRA along with the detected intent and extracted entity is depicted in Figure 3.4.

We choose DIET among other large-scale pre-trained models for several reasons.

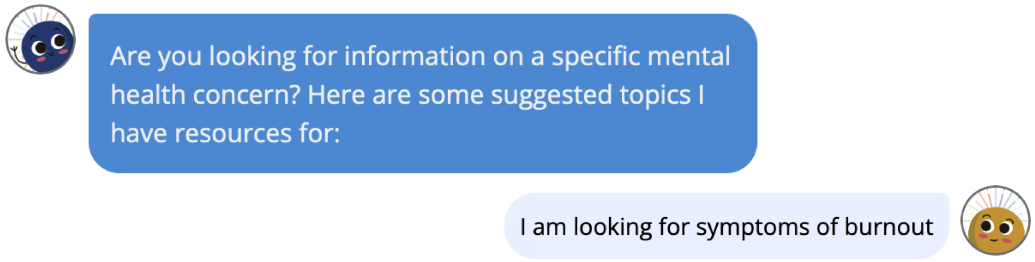


Figure 3.4: An interaction between a user and MIRA might go as follows: the user expresses wanting to know about the symptoms of burnout. MIRA should recognize the user's intent (seeking information about symptoms) and identify the relevant entity ("burnout"). This allows MIRA to provide an appropriate and informative response regarding burnout symptoms.

1. It features a modular architecture that seamlessly integrates into a software development workflow.

2. It matches LLMs models in both accuracy and performance and is faster to train.

3. Previous research conducted by our team found that DIET outperformed other predictive models and architectures for both augmented and pre-augmented datasets [106].

DIET is a multi-task transformer-based architecture designed to leverage the inter-dependencies between intents and entities for simultaneous detection [103]. The key components of the DIET architecture are:

1. **Featurizer**: Featurization converts user messages into numerical vectors. The DIETClassifier relies on two types of featurizers: dense and sparse. Dense features includes a range of pre-trained embeddings such as BERT [107], GloVe [108], or ConveRT [109]. In contrast, sparse features consist of token level one-hot encodings and multi-hot encodings of n-grams ($n \leq 5$) [103].

2. **Multi-Layer Transformer**: The core component of DIET, a multi-layer transformer model that takes the preprocessed input features and performs the joint intent classification and entity recognition tasks. DIET employs a multi-layer transformer architecture to effectively encode the contextual information across the entire input sentence. Specifically, it utilizes a two-layer transformer model [110] that incorporates the relative position attention mechanism [111].

3. **Named Entity Recognition**: DIET employs a Conditional Random Field (CRF) [112] tagging layer over the transformer output sequence to predict a sequence of entity labels y_{entity} corresponding to the input tokens. The loss function $L_E = L_{\text{CRF}}(a, y_{\text{entity}})$ measures the negative log-likelihood of the predicted entity labels given the transformer output a [113]. This CRF layer, combined with the transformer's contextual encoding, enables accurate entity recognition while jointly training for intent classification through multi-task learning.

4. **Intent Classification Head**: This component is responsible for predicting the intent label based on the output of the transformer. The transformer output for the CLS token (a_{CLS}) and intent labels (y_{intent}) are embedded into a shared semantic vector space $h_{\text{CLS}} = \text{E}(a_{\text{CLS}})$, h_{intent} = E((y_{intent}) with a dimensionality of 20. DIET employs the dot-product loss function [114] to maximize the similarity ($h_{\text{CLS}}^T * h_{\text{intent}}^+$) between the CLS token representation and the target intent label, while minimizing similarities with incorrect intent labels. This aligns the CLS token encoding with the correct intent, leveraging the transformer's contextual understanding for effective intent classification.

3.4.3 Training Data

The team created a comprehensive dataset tailored to a mental health chatbot for the pilot version. The dataset has grown since the inception of MIRA, currently containing 91 intents with 2,292 examples. The intents were defined based on the specific use cases of the MIRA chatbot. For instance,intents such as "feeling symptoms","need information","need crisis support" were defined to capture relevant user inputs. The training data contains at least 10 examples for each intent.

For instance, for the "feeling symptoms" intent, the examples include:

- "I feel depressed"
- "I don't want to get out of bed"
- "I have no energy"

During this process, we annotated key keywords that the chatbot should detect. For example, if a user types "I want to know about the symptoms of anxiety," the word "anxiety" is marked as an entity.

To improve the accuracy of intent detection and entity extraction, we employed several data augmentation techniques. These included EDA (Easy Data Augmentation) [115], which uses methods like synonym replacement, random insertion, random swap, and random deletion to diversify the dataset. Synonym replacement involved swapping specific terms with synonyms from WordNet [116] to maintain the original meaning [117]. Embedding replacement identified contextually similar words in a latent representation space [118]. Additionally, we used language models to predict and replace words based on context, allowing for more localized adjustments [119]. These augmentation techniques collectively enhanced the robustness and performance of MIRA.

3.4.4 Dialogue Management

The Rasa agent in Fig 3.3 is responsible for processing user inputs and selecting subsequent actions in the conversation. The agent retrieves the user's conversational inputs from the MIRA Chat Interface and interprets the input to determine the next steps in the dialogue.

The agent is also responsible for intent detection and entity extraction, a function performed by Rasa's NLU pipeline. This functionality is key for a positive user experience. Accurately understanding the user's requests promotes the selection of relevant actions, increasing engagement with the chatbot. The NLU Pipeline stores the DIET model trained on curated training data in the Filesystem. The DIET model, subsequently identifies the user's intent and extracts relevant entities from their messages.

The conversation flow or the sequence of actions the Rasa agent can take was carefully curated by domain experts and organized into a structured tree framework. The dialogue manager is a core component in the backend, responsible for guiding the conversation flow. Based on the context of the ongoing dialogue, the component determines the appropriate following action to maintain coherence and engagement within the conversation. The tracker object stores the dialogue state. A tracker object within a conversation session acts as a stateful component. tracker stores slots, as well as a log of all the events that led to that state and have occurred within a conversation.

The user message flows through Rasa's NLU pipeline to the interpreter, which detects intents and extracts entities using the pipeline's components. The tracker object tracks the state of the conversation. The policy uses the tracker state to detect which module it is in and chooses the corresponding action. The chosen action is logged by the tracker. Finally, a response is sent to the user.

When a user interacts with MIRA, they navigate through a predefined conversation path defined by the conversation flow. Occasional deviations from the path, called jumps, were permitted in a few cases based on the context and the identified intent. During interactions with MIRA, user messages are classified into existing intents, triggering corresponding actions represented by nodes within the conversation tree.

3.5 Limitations

The pilot version of MIRA was pivotal, as it laid the groundwork for the subsequent iterations. This included establishing the foundations for MIRA using the Rasa framework, creating a portal for cataloguing and annotating the database of resources, constructing conversational flows using a flowchart tool, meticulously curating extensive training data for each intent, exploring various architectures for intent detection and entity extraction and developing strategies for data augmentation. While these efforts provided a strong foundation for MIRA's functionality, they also revealed limitations that required further improvement.

One major limitation with the pilot version of MIRA was that it relied heavily on Rasa's rule-based system. The custom actions were hard-coded in Python to direct conversations and provide users with relevant resources. This approach was rigid, making updates cumbersome and requiring significant Python expertise for modifications beyond minor adjustments. As custom actions stretched the capabilities of the Rasa framework, developers were directly responsible for managing the conversation state, increasing the risk of unnoticed errors that could disrupt the conversation flow.

Another challenge with the pilot version of MIRA was that it employed a conversation flow that lacked transparency for both programmers and the

larger team. The transitions or jumps between modules relied heavily on slot information, leading to erratic jumps between modules when key information was missing. This made it difficult to understand and debug the conversation flow.

To address these challenges, the **Chatbot Builder** was developed as a solution. The conversation flow was restructured using the chatbot builder to ensure logical transitions between modules, enhancing transparency and simplifying updates. As a result, the decision-making and conversation flow became clearer, more intuitive, and easier to manage, reducing the need for extensive coding expertise and improving the overall functionality of MIRA.

Chapter 4

Implementation of the Chatbot Builder

A conversation flow is a decision tree of possible conversation paths, typically represented as a directed graph, where each node or module, defines the subsequent conversational path based on specific conditions or user inputs. The detailed conversation flow includes everything from the sequence of user messages, responses from the chatbot, and actions that occur during a conversational interaction. Intelligent conversation agents should be able to handle complex dialogues and cater to a diverse range of individuals. In a rule-based setting, this means an increased number of nodes or decision points, each contributing to a significant expansion in potential conversation paths.

A key requirement for an effective chatbot is smooth transitions between conversational modules and pathways. Abrupt transitions can disrupt the natural flow of the conversation, leading to confusion, frustration, and a negative overall experience for the user. However, the pilot version of MIRA had unreliable transitions or *jumps* between modules which caused errors and interruptions in the user experience, leading to conversation non-progress. Li et al. defines conversation non-progress as scenarios where a chatbot either mis-recognizes or fails to recognize user inputs, leading to user frustration and potentially abandoning the conversation [34].

The hard-coded conversation flow used in the pilot version of MIRA presented challenges for expansion and collaboration, as it was not easily modifiable. In practice, programmers are not tasked with designing conversation flows. Designing an effective conversation flow requires collaboration with domain experts. Adopting a modular and flexible approach to conversation flow design can improve collaboration and iterative improvement.

We propose a chatbot builder as a solution to design and modify conversation flows more efficiently. The chatbot builder provides a user-friendly interface for designing, updating, and managing the conversation flow for both developers and non-programmers. In the chatbot builder, the conversational elements are segmented into modules. Each module has its own set of rules and actions and defines the subsequent path based on specific conditions. We use a *Rasa-agent* to deterministically execute the logic defined within the conversation flow, manage the transitions between modules, evaluate conditions, and ensure the conversation progresses as per the defined rules and user interactions.

Since MIRA serves as both a guidance system for resource navigation and a conversational agent, the users can interact with open-ended text outside the structured conversation flow. We incorporate features in the chatbot builder to allow our Rasa-agent to interact with the user through responses and questions, displaying suggestions as potential answers for questions in the form of buttons. At the same time, the user can interact with MIRA by clicking on these buttons or through open-ended texts within the context of mental health.

4.1 Workflow and Integration

This visual interface of the chatbot builder, as illustrated in Fig. 4.1 simplifies the process of creating, modifying, and organizing conversational modules and their transitions. The conversation flow is represented and stored in JSON

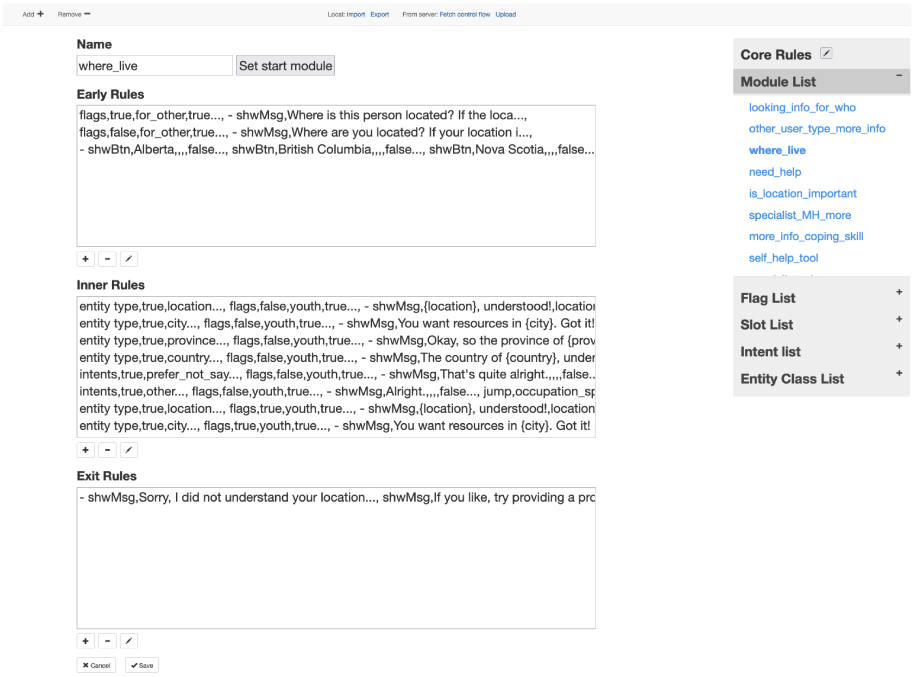


Figure 4.1: The user interface of the chatbot builder platform displays the `where_live` module, which includes early, inner, and exit rules. On the right, a list of components are displayed to assist with defining conditions or actions. At the top, users have the option to add or remove modules, as well as import or export conversation flows between their local machine and the server.

Figure 4.2: Figure shows the final output of the `where_live` module from Fig. 4.1 on the chat interface, constructed using the chatbot builder.

format, allowing for easy manipulation and integration with other components of the chatbot system. When users save or export their conversation flow, the file is validated on the server using a simplified chatbot version. This chatbot verification process ensures the integrity and functionality of the conversation flow before deployment. The chatbot builder and the chatbot share the same server, this makes it possible for the chatbot builder to directly update the active conversation file by serving a JSON file to the chatbot. This integration ensures immediate changes, eliminating the need for manual file transfers or synchronization. Additionally, the chatbot builder can also load the chatbot's current conversation flow, enabling developers to visualize and modify the live conversation logic as needed.

The chatbot builder primarily utilizes Python and JavaScript. The chatbot builder is built using Flask, a lightweight Python web framework, which serves both the frontend and backend operations. The web page is hosted directly through Gunicorn, a Python Web Server Gateway Interface (WSGI) Hypertext Transfer Protocol (HTTP) server. The setup runs on a local server within a development environment that mimics production settings. The bulk of the chatbot builder's functionality is handled by JavaScript, which processes JavaScript Object Notation (JSON) within the browser and reduces the load on the host server by running entirely in-browser.

The modifications made to the Rasa framework to integrate custom actions have significantly changed how conversation flows are managed and deployed. This shift eliminates the reliance on Rasa's rule-based flows and the need for hard-coded custom actions. Decoupling the conversation flow management from Rasa's core components gave us higher flexibility and control over the conversational logic. Through custom Python actions, we now use a modular and dynamic approach. This approach makes it possible to add new conversation flows or modules without disrupting the existing logic, allowing us to

scale and adapt MIRA to changing requirements. This shift towards a more modular approach made the development of the chatbot builder possible. The conversation flows in the chatbot builder are represented and stored as simple JSON files, which can be easily tracked and managed using version control systems like Git. The version control simplifies collaboration and makes it easy to revert by maintaining backup logs of conversation flows. One of the most significant advantages of this approach is the ability to update or revert conversation flows without any downtime, ensuring continuous and uninterrupted chatbot operation.

4.2 Components

The platform consists of module-based building blocks that users can create and customize to implement *rules* and *actions*. We refer to modules with the conversation flow that have their own sets of rules and actions as *dynamic modules*. Each module is responsible for handling a single conversational turn, which involves the exchange of a message with the user, before jumping to another module. We refer to hard-coded modules, as *external modules*, they have been almost entirely phased out in favor of dynamic modules.

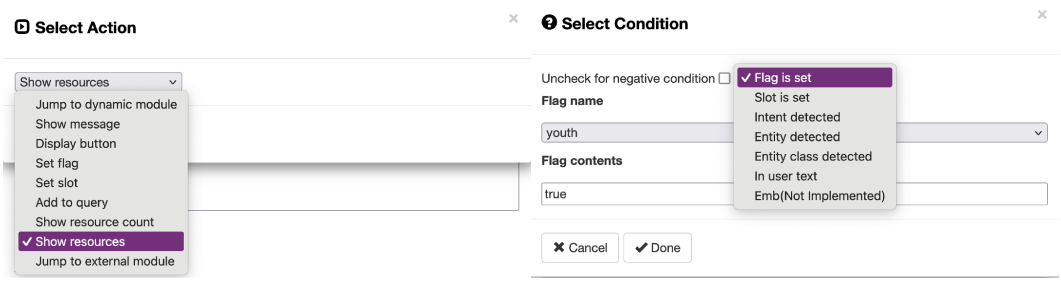
Rule sets contain multiple logical rules that can conditionally trigger a set of actions based on the content of the user message. The rules can be left empty to ensure the automatic execution of a specific predefined action. The rule sets fall into three categories, early, inner and exit rules.

Early rules: These rules are activated upon entry into a conversational module. Their primary function is to display any relevant text, buttons, or questions that introduce the module's context or prompt the user for input. For example, in Fig. 4.1, the `where_live` module is composed of early rules that

prompt the user for their location. The phrasing of the message varies based on a flag value: in one case, it asks for the user's location, while in another, it inquires about the location of someone else. It then presents a list of locations as buttons representing various Canadian provinces, along with options for "Other" or "Prefer not to say." Figure 2 depicts the chat interface where the final output of this module, constructed using the chatbot builder, is displayed to the user.

Inner rules: Active during the user interaction phase, inner rules are triggered after the user submits a message or input. These rules are responsible for processing and responding to the user's input, storing relevant data, and determining whether to transition to a new module based on the received information. Inner rules form the core logic of the conversational flow, enabling the chatbot to understand and respond appropriately to the user's inputs. For example, in Fig. 4.1, the inner rules are defined to capture the location through entities such as city, province, or country. Depending on the youth flag, these rules decide the direction of the conversation flow. If the user is an adult and the youth flag is false, the flow transitions to the occupation-specific module. On the other hand, if the youth flag is true, the flow transitions to the need help module.

Exit rules: They are triggered upon exit from a conversational module. Exit rules run when all inner rules are implemented, but no jumps are executed. They serve as a fallback mechanism. Exit rules are used to inform the user that their message was not understood, provide additional context or clarification, or initiate a jump to a specific conversational module regardless of the user's input as a default behaviour. For example, in Fig. 4.1, if the inner rules do not result in any transitions and remain within the same module, the exit rule



(a) Choice of actions (b) Choice of conditions

Figure 4.3: Figure displaying the set of possible actions and conditions (rules) available in the chatbot builder for constructing and customizing conversation flows.

informs the user that it did not understand their location.

The rules can be used to prompt a range of actions. The actions are executed only when all the specific conditions are met. To ensure a particular action is always executed, the conditions field can be left empty, prompting a default behaviour. The chatbot builder currently supports nine actions. We will continue to improve our platform to support additional actions based on the evolving requirements of our team.

The actions the chatbot builder supports include:

1. Responding with a message
2. Displaying responses in the form of buttons
3. Transitioning or *jumping* to another dynamic module
4. Jumping to an external module
5. Setting flags
6. Setting the value of slots (entities)
7. Adding strings to a running query
8. Sending that running query to the MIRA Library and displaying relevant resources

9. Displaying the number of resources

Slots and *flags* are ways to store and manage information for controlling the conversation flow or for querying external resources. The contextual information stored in slots is sent as a query to the resource library to retrieve relevant resources. Slots can also be used to control the conversation flow; Decisions regarding the next conversational module or pathway can be made based on the contents of the slots.

Flags are used exclusively to control the conversation flow. Unlike slots, flags are not included in queries sent to the resource library. Flags can be used to track various conversational states, such as whether a particular topic has been discussed, whether specific conditions have been met, or whether certain actions need to be taken.

Finally, *core rules* are always active and can be used to interrupt the flow, regardless of the module. For instance, in MIRA, the core rules are always active and continuously detect if a user wants to end the conversation, indicates self-harm or suicidal intent, or is in an emergency.

4.3 Interaction Flow in MIRA

We customized the Rasa framework to exclusively support custom Python actions, which drive the functionalities of the chatbot builder. Custom actions are used in the following ways:

- Load and switch between conversation flows.
- Manage the processing of dynamic modules.
- Interface the chatbot with the resource library for query management via an API.
- Track the conversation state through Rasa slots.

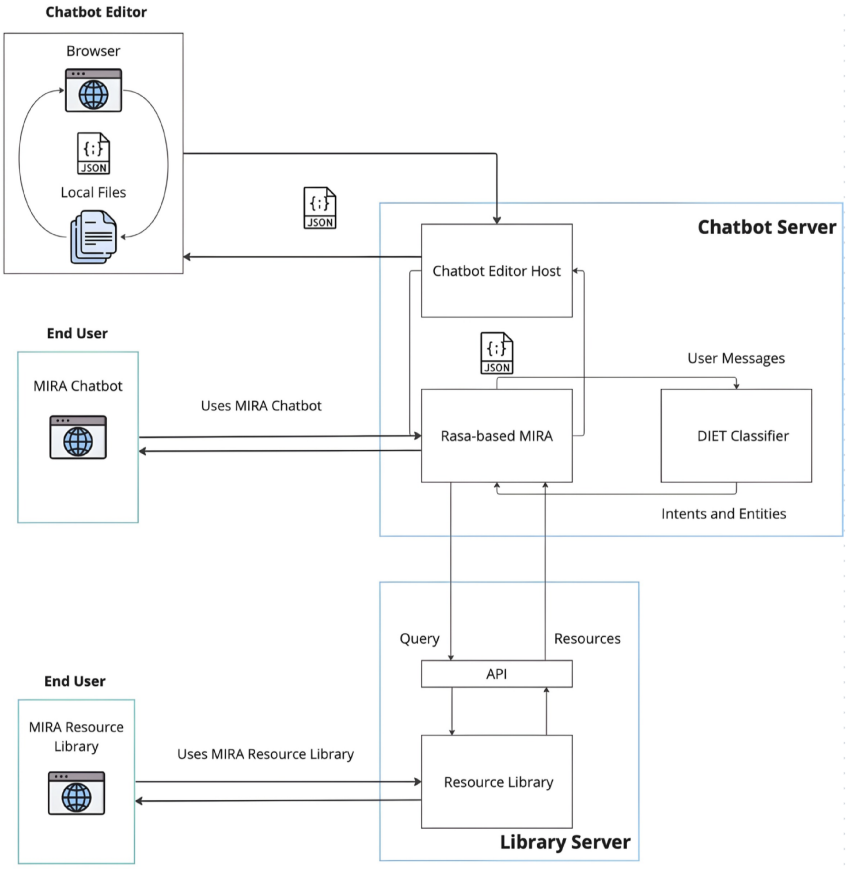


Figure 4.4: This figure illustrates the interaction between the Chatbot Builder or Editor, Chatbot Server, and the Resource Library. It outlines the design and execution of conversation flows. End users interact with the system via the MIRA Chatbot and the MIRA Resource Library.

- Handle the processing and interpretation of intents and entities detected by Rasa NLU.

The interactions among the various components in the MIRA chatbot architecture are depicted in Fig. 4.4. The internal team uses the Chatbot Builder to design and manage conversation flows. These flows can be saved to or loaded from the Chatbot Builder Host as JSON files. The Chatbot Editor Host communicates with the Chatbot Server to send updated JSON conversation flows or retrieve existing conversation flows. When end users interact with the MIRA Chatbot, their messages are processed by the Rasa-based MIRA agent using the DIET classifier, which identifies user intents and extracts relevant entities.

This extracted information is used by the Rasa agent to guide the conversation according to the conversation logic in the JSON file.

The Rasa agent formulates a structured query based on the extracted entities. This query is sent to the Resource Library through an API to fetch and retrieve relevant resources. The resources are scored, filtered and ranked based on their relevance to the user's query. The ranked resources are subsequently returned to the chatbot for user interaction.

End users interact with the MIRA Chatbot through their browsers, sending messages that are processed to generate appropriate responses. They can also directly access the MIRA Resource Library for specific queries, bypassing the chatbot if necessary or in case of interruptions in the conversation flow.

4.3.1 Analogy of a Banking Assistant

Since the effectiveness of the Chatbot builder is not limited to MIRA and applies to any Rasa-based general chatbot that requires ongoing iterations, we explain the components of the chatbot builder using the analogy of a banking assistant. Consider a task-oriented banking chatbot built to assist customers with banking transactions and inquiries. Here, the modules could correspond to various services the bank offers, presenting high-level choices for users such as transferring funds, applying for a loan or checking their balance. Depending on the user's preference, the conversation flow would direct them to relevant modules designed for that specific ask. For instance, if the user indicated wanting to apply for loans, the flow could lead the user to the `apply_loans` module, followed by `specific_loan_type` modules, then a `loan_eligibility_check` module, and so forth. The rules with each module may include verifying user credentials, detecting user intent, and identifying transaction details as particular entities. The chatbot executes actions such as retrieving account balances or redirecting users to a secure payment portal. If a user indicates a preference

to speak with a human agent, the system detects this core rule and transfers control to a customer service agent.

4.4 Flow Visualizer

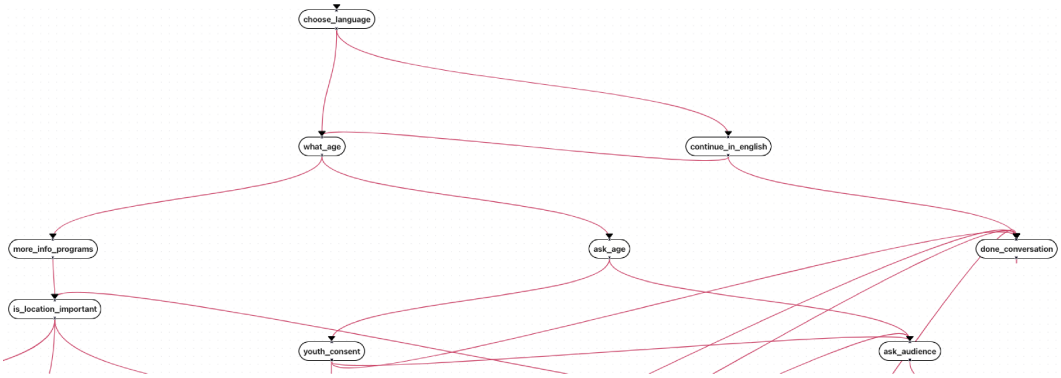


Figure 4.5: A snippet of the conversation flow for MIRA, constructed using the flow visualizer.

We also built a dynamic flow visualizer to visualize the conversation flow. As the conversation structure grew in size and complexity, it became evident that having a visual representation of the conversation flow would make it easier to understand and verify the overall structure. The team previously hand-crafted the visualization of the conversation flow structure and drew detailed diagrams using LucidChart [12] to map out each conversational pathway. The dynamic visualizer provides a clear and intuitive representation of the existing flow, and simplifies the processes of making changes, adding new paths, and restructuring the logic. The flow visualizer is particularly useful in extending MIRA to multiple audiences. A snippet of the MIRA conversation flow, designed with the chatbot builder, is illustrated using the flow visualizer in Fig. 4.5. Additionally, an illustration of the overall conversation structure of MIRA is presented in Fig. 5.2 in the next chapter. With the visualizer, it is easier to debug the flow and identify gaps, missing paths or inconsistencies in the conversation logic. The visualizer essentially streamlines the iterative

design process and makes it easier for the larger team to collaborate in designing various conversation pathways. Once a conversation flow is designed, it is tested rigorously and refined iteratively by our team members and volunteers from MDSC.

To build the dynamic flow visualizer, we implement a recursive function that traverses the JSON file containing the conversation flow logic. This function identifies all the linked modules starting from the *start module* connected through inner and exit rules. For each unique module encountered, new nodes are created in the visualization. The resulting nodes represent individual modules, and the links represent transitions or connections between modules based on defined rules. If a module has already been visited during the current traversal, the function avoids creating duplicate nodes or links, preventing infinite recursion.

Nodes are arranged hierarchically to reflect the logical flow of the conversation. Parent nodes are placed above their child nodes, creating a tree-like structure. Nodes are spaced out to avoid crowding, with the layout organized into rows and columns. The spacing between nodes is dynamically adjusted to accommodate the complexity and number of elements. An algorithm ensures that nodes are arranged in a hierarchical order, from root to leaf, maintaining their relationships. This approach helps keep the visual flow clear and easy to follow, even as the number of nodes increases. The layout adjusts to different zoom levels and screen sizes.

The layout follows a deterministic approach. This means, generating a flow visualization from a given conversation flow will consistently produce the same visual representation each time. Adding or removing a node typically results in incremental changes to the visualization. The layout adjusts to these modifications while maintaining the overall structure and hierarchy. However, significant changes can occur in specific scenarios. For example, if a node

is added or removed in a crucial part of the hierarchy, such as the root or major branching nodes, the entire structure may need to be reconfigured, causing noticeable shifts in the layout. Additionally, in cases where nodes are highly interconnected, removing or adding a node can substantially alter the arrangement of other nodes, especially if a central node is involved.

In the future, we plan to explore alternative approaches, such as tree layouts with collapsible nodes. These layouts can simplify the view by allowing users to focus on specific branches of the conversation flow, thereby enhancing clarity for large and complex structures. We also plan to explore methods to tightly integrate the chatbot builder with the flow visualizer by making the visualizer more interactive, so that clicking on a node within the visualization would directly navigate the user to the corresponding module in the chatbot builder interface.

Chapter 5

Results and Discussion

In this chapter, we discuss the impact of the chatbot builder, starting with the user interface that we use to create and manage conversation flows. In the next Section 5.2, we show the visuals from the flow visualizer of the conversation flow we built. We highlight the extensions we built with the chatbot builder, such as youth, French, and Indigenous pathways, with snapshots of these flows deployed on MIRA's chat interface. To evaluate improvements over previous methods of creating conversation flows, we compare time estimates required to build a conversation flow using the chatbot builder versus the hard-coded approach. Finally, to verify whether flows created using the chatbot builder have fewer disruptions compared to the previous method, we analyze the trend in the frequency of fallback or default responses.

5.1 Chatbot Builder

The key advantage of the chatbot builder is its ability to serve as a conversation flow manager, allowing the transfer of contextual information from intents detected from current modules to intents detected in subsequent modules in terms of flags, slots, or entities. Additionally, the chatbot builder allows our team to address and create follow-up actions for modules within the conversation flow. This includes designing fallbacks and exit rules to handle scenarios

where the chatbot may encounter uncertainties, such as reaching the end of a conversation pathway or disrupting the flow because particular user intent is either undefined or not routed to any modules.

These features enhance our chatbot's overall robustness and adaptability, enabling smoother and more engaging user experiences while navigating through different conversation flows.

The chatbot builder proved beneficial in several ways:

1. Enables fast prototyping, resolution of bugs, and updates to the flow of the conversation.

2. Supports collaboration within an interdisciplinary team. The larger team does not need coding experience to make changes to the flow.

3. Helps integrate new pathways to the conversational flow. This, in turn, makes it easier to extend the functionalities of MIRA to serve more communities.

The interface of the chatbot builder platform is depicted in Fig. 5.1. Since the development of the chatbot builder, we have expanded the scope of MIRA beyond healthcare workers to include the general audience, youth (14 years and older), French-speaking Canadians and Indigenous communities, broadening its accessibility and impact. The versions involving French and Indigenous communities are undergoing beta testing and will soon be publicly deployed.

It should be noted that the chatbot builder keeps all functionalities intact that were previously achievable through code, ensuring no loss of capabilities and maintaining the system's operational scope. While a programmer's expertise is still needed to incorporate newer features, the chatbot builder complements this process, making routine tasks more efficient and allowing for quick iterations.

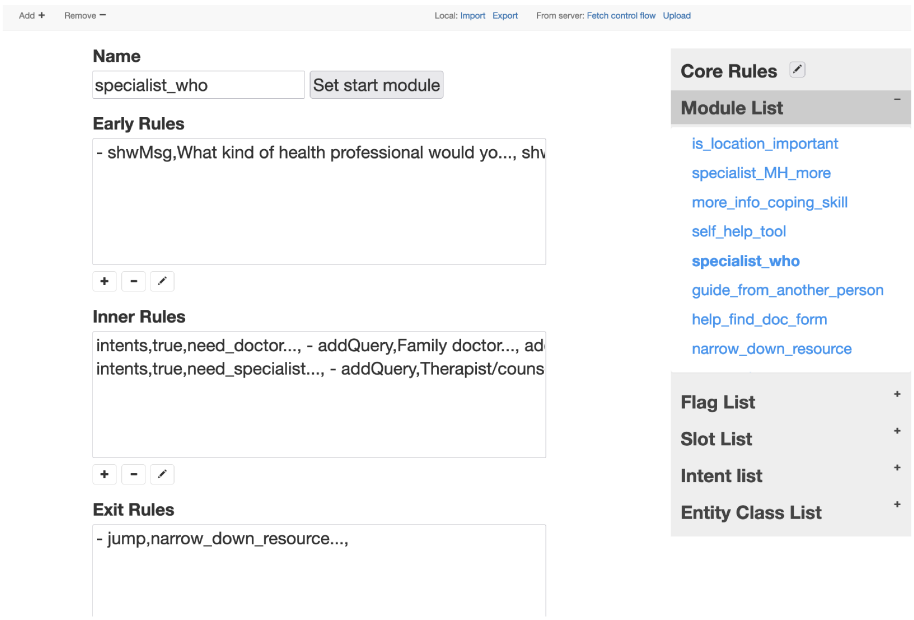


Figure 5.1: The UI of the chatbot builder platform shows the *specialist who* module. The early rules prompts users to choose a health professional, while the inner rules detect the *need doctor* or *need specialist* intents. On exit, as a fallback rule, it transitions to the *narrow down resource* module.

By extending MIRA to support broader populations, we address the third research question by ensuring MIRA's functionalities are preserved for health-care workers while adapting it for other groups. We also partially address the fourth research question on bilingual capabilities by creating a French version of MIRA. The French version, for simplicity, is a separate chatbot operating primarily through a rule-based flow, with users selecting options rather than engaging in open-text conversations in French. Additionally, when users switch from English to French, the chatbot restarts the conversation. In the future, we aim to make this transition more seamless, allowing the conversation to continue in French from where it left off in English. We will conduct extensive user trials with French-speaking users to gather insights and make iterative improvements, ensuring the open-text conversational experience in French is as seamless and effective as it is in English.

5.2 Flow Visualizer

The dynamic flow visualizer was useful for visualizing and iterating on the conversation flows through graphical representations of conversational pathways. Previously, resolving issues within these flows solely through code proved to be challenging and time-consuming. The conversation flow designed exclusively with the chatbot builder for MIRA is depicted using the flow visualizer in Fig. 5.2. We integrated the youth pathway within the English version of MIRA using the chatbot builder.

The flow visualizer begins with the `choose_language` module, where users select their preferred language. Following this, the `what_age` module determines if the user is seeking resources for an adult or a youth, directing them to the respective pathway. For youth, there is an additional requirement to complete a youth consent form. Subsequently, users are guided through modules that collect demographic information. For adults, this includes occupation-specific information, while for youth, MIRA gathers details about the audience, such as whether they are a parent/caregiver, child, or teenager. If location is relevant to the information users seek and they choose to disclose it, both youth and adults are prompted to select their location from a list of Canadian provinces. This step is followed by questions about their specific mental health concerns, the type of mental health issues they are seeking resources for, and the kind of resources they need.

The core of MIRA is the `need_help` module, which branches into three modules: informational resources, programs and services, or homepage resources. Each of these modules is tailored to guide users through a pathway that provides resources relevant to their needs. This approach helps users navigate the MIRA resource library in a structured way, accessing general information, specific programs and services, or direct links to homepage re-

sources.

5.3 Extensions

Figures 5.3a and 5.3b showcase snapshots from the MIRA chat interface featuring the French and Indigenous extensions, respectively. Figures 5.4a and 5.4b show snapshots of the English version of MIRA. Figure 5.4a shows the youth pathways, and Figure 5.4b displays the adult pathways. The below extensions were constructed entirely using the chatbot builder.

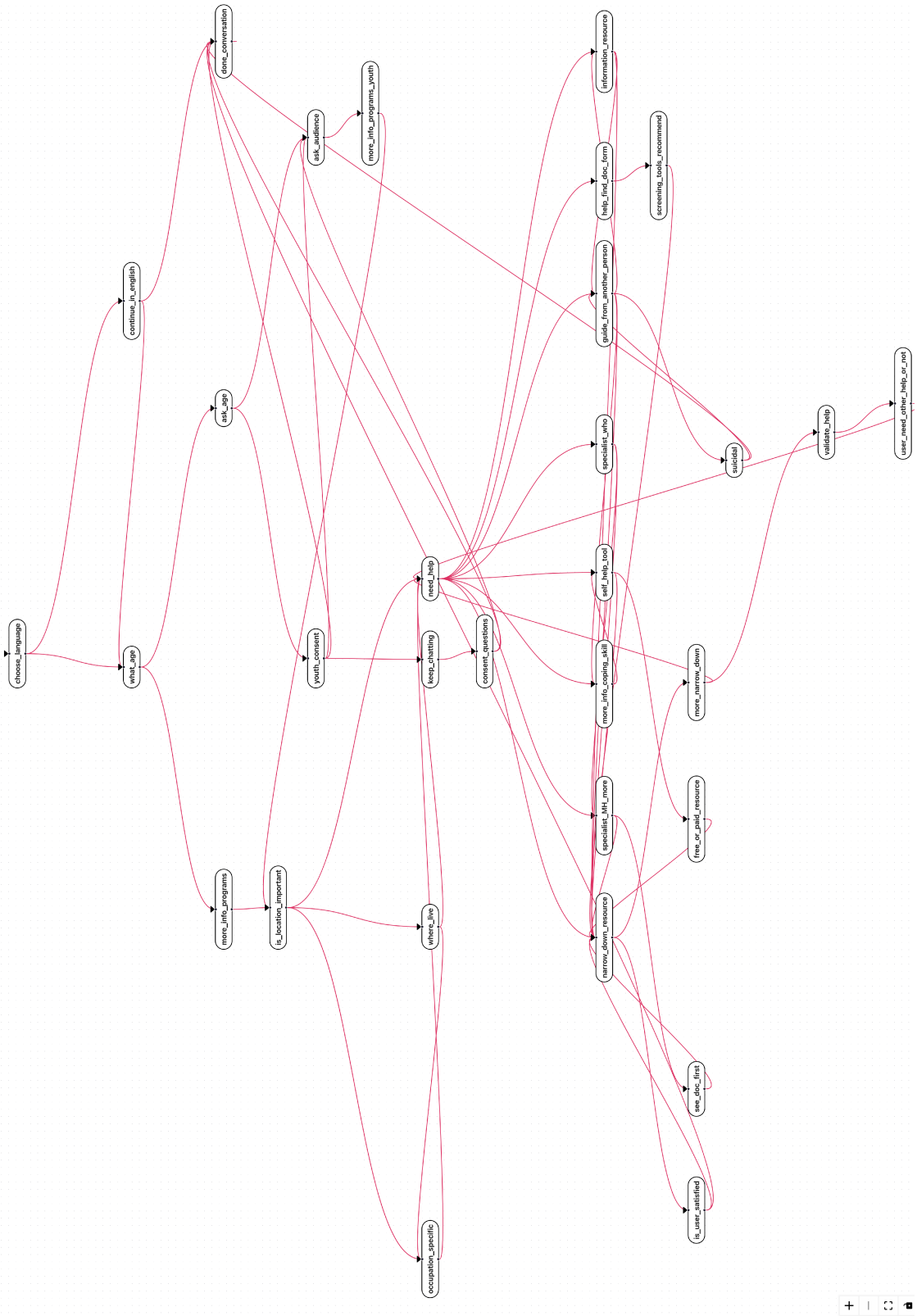
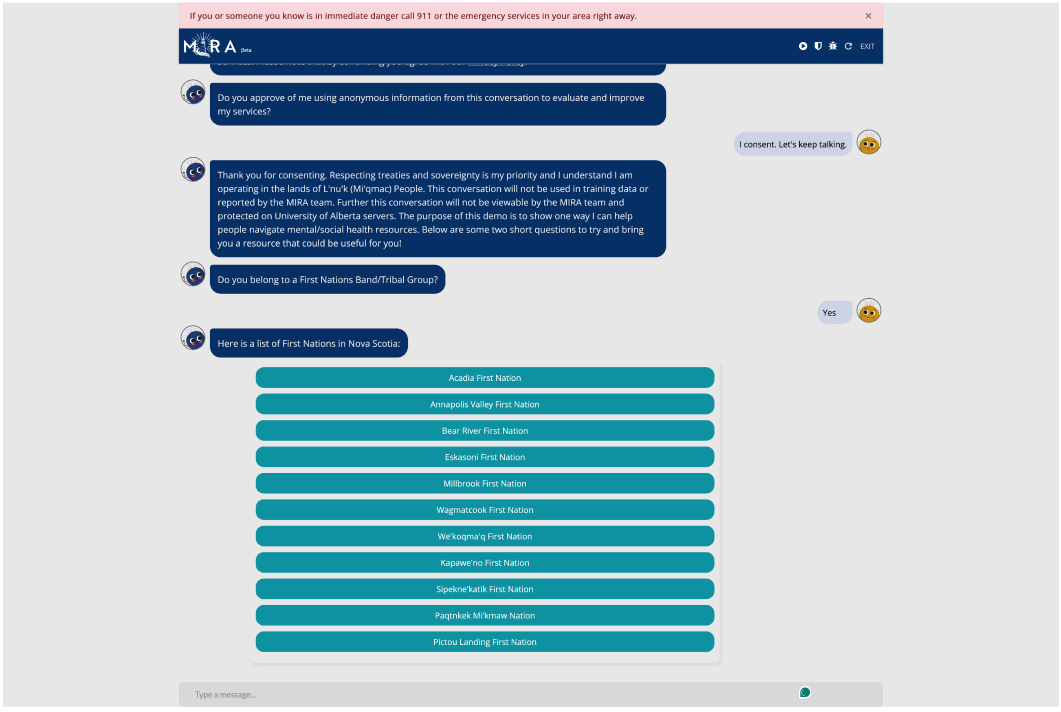


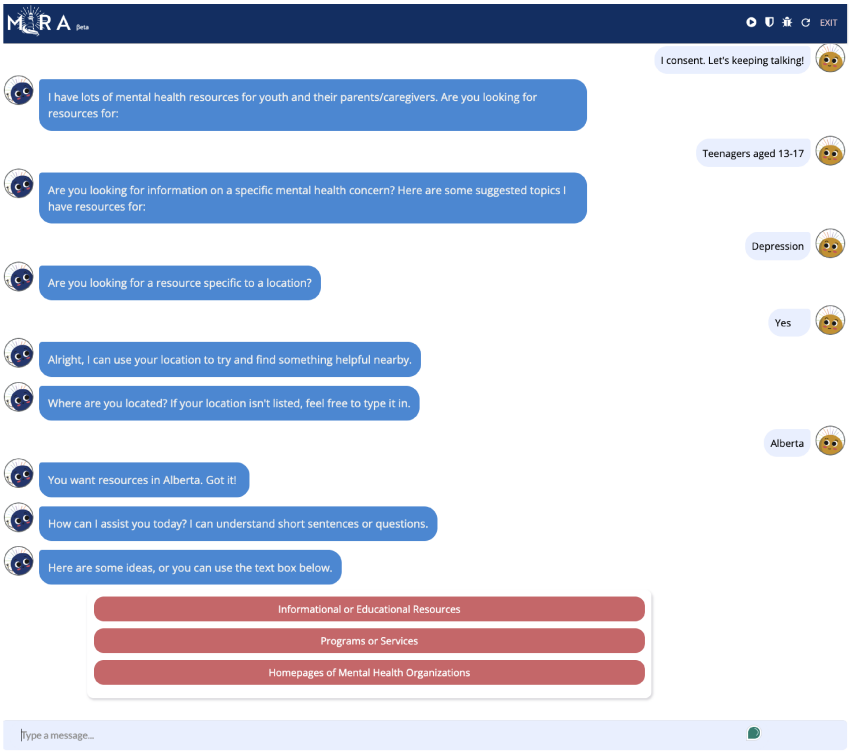
Figure 5.2: The conversation flow for MIRA, constructed using the flow visualizer.

(a) French Extension

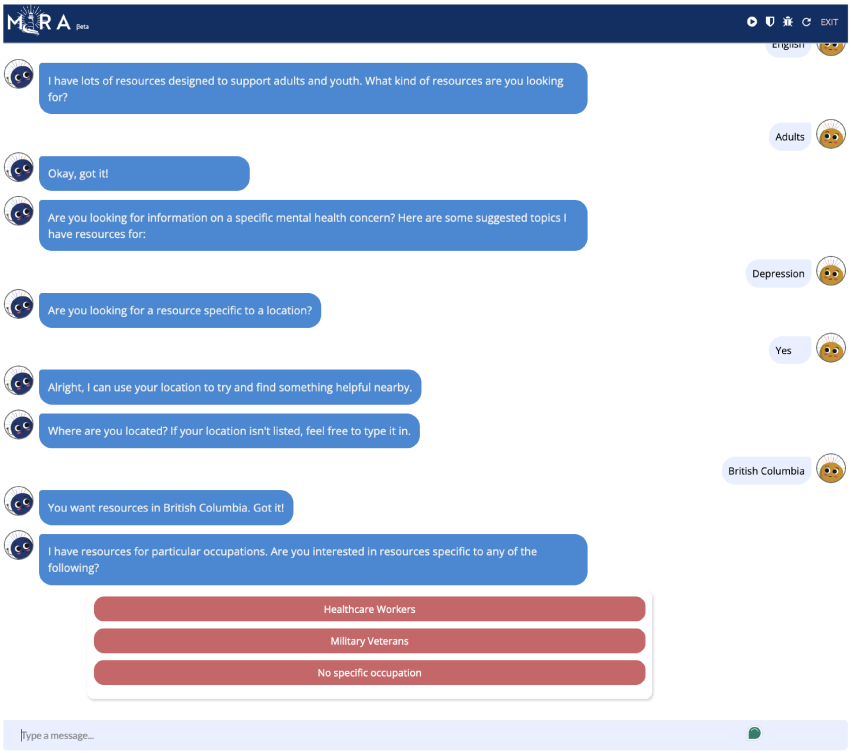


(b) Indigenous Extension

Figure 5.3: The conversation flow for the French and Indigenous extensions.

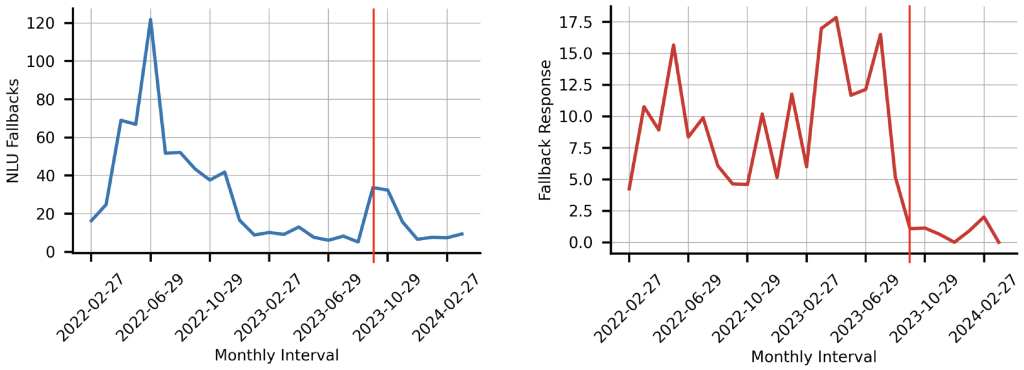


(a) Youth



(b) Adult

Figure 5.4: The conversation flow is structured differently for the adults and the youth.



(a) Frequency of NLU fallback intent per 10,000 messages over time

(b) Number of fallback responses per 10,000 messages over time

Figure 5.5: Trends in NLU fallbacks and response rates over a monthly interval. The red bar indicates the shift to the conversation flow created with the chatbot builder.

We evaluate the effectiveness of the chatbot builder by assessing the improvements in the chatbot brought about by the chatbot builder and the flow visualizer. This includes measuring if there are fewer disruptions in the conversation flow and misclassified intents. A decrease in misclassified intents implies that the intent handling and routing to relevant modules have improved. Consequently, the chatbot builder gave us more control over managing intents and follow-ups, allowing us to extract intents better and establish clear conversation paths. To empirically estimate the chatbot builder's effectiveness, we recorded metrics to determine if there were reduced interruptions in the conversational flow and fewer instances of misclassified intents. This evaluation aimed to quantify the improvements in chatbot performance enabled by the chatbot builder platform.

The NLU Fallback intent is triggered when the confidence level of a detected intent is below 60%, prompting the chatbot to send a fallback response requesting the user to rephrase their sentence. MIRA can also respond with an error response when the dialogue flow encounters an error or fails to progress. With the release of a version of MIRA that used the conversation flow designed

with the chatbot builder, we observed a significant drop in the number of error responses generated by MIRA, as shown in Fig. 5.5a, and a decrease in the number of times the NLU Fallback intent was triggered, as shown in Fig. 5.5b. This indicates that the system is more stable and easier to troubleshoot and iterate since the entity and intent detection remain the same.

Integrating a conversational script for approximately five interconnected modules into the existing conversation flow can be implemented in under an hour using the chatbot builder. On the other hand, integrating the same modules through code is more time-intensive and depends on several factors. The complexity of the nodes plays a role. Simple, linear conversations might take around 2 hours to code, whereas complex, branching dialogues with multiple conditions could require 4 to 5 hours. However, this assumes the developer is familiar with the existing system and knows the codebase well. A new developer might take significantly longer to understand the architecture and integrate pathways directly via code. While these measures provide an estimate, we plan to conduct a more rigorous study in the future to measure and evaluate the actual time required for these tasks. This study will compare the time it takes for programmers to implement features through traditional coding methods versus the time it takes for non-programmers to modify or create a flow entirely using the chatbot builder.

In a complex conversation flow with interconnected modules, a single error can cascade across modules in ways that are not obvious or intuitive to resolve. These errors can cause conversation disruptions or non-progress, significantly impacting user experience. Debugging errors directly on the codebase is challenging, time-consuming and prone to mistakes. Identifying the modules responsible for errors through the chatbot builder is easier. This visual approach to modifying the conversation flow reduced the time needed to resolve issues, allowing for real-time testing and iteration.

The chatbot builder allows changes to the conversation flow such as rephrasing questions, modifying button text, rerouting intents to specific modules, creating new modules, implementing conditional routing based on flags or slot values, and removing unnecessary modules. However, there are challenges the chatbot builders cannot help with. Debugging can become more time-intensive for issues related to the backend infrastructure or algorithmic errors. For instance, consider issues like the resource ranking algorithm failing to retrieve the right resource or the system not capturing a user intent because it is not defined. Resolving such requires the expertise of a developer to examine the underlying code or database structures since they cannot be resolved through the chatbot builder interface.

Chapter 6

Conclusion

This work extends the capabilities of the pilot version of MIRA to new audiences to maximize its impact and reach. We built a chatbot builder platform to accelerate the development of a chatbot with rapidly evolving requirements in a multidisciplinary environment. It is to be noted that the primary focus of this work is not towards the development of a chatbot but to identify and develop flexible and scalable methods to extend the functionalities of chatbots. To achieve this, we evaluate flexibility and scalability in terms of the overall impact of successfully extending MIRA to other audiences

6.1 Structured Conversations and Safety

MIRA employs a rule-based conversation flow, resembling a tree structure of possible conversational pathways. Mental health chatbots often leverage rule-based dialogues to ensure safety and enforce guardrails on topics they can discuss. Notable examples of mental health applications that use structured conversations are Woebot [120] and Wysa [121]. The decision to use a rule-based approach makes it easier to guide users navigate the MIRA resource library and encourage focused conversations within the mental health domain. Additionally, studies emphasize the importance of providing suggestions to users through prompts, guiding the conversation in a structured manner [34].

In MIRA, we use clickable buttons as prompts to guide user navigation and interaction.

6.2 Chatbot Builder Platform

Subject matter experts are actively involved in designing the conversation flows for MIRA. As we expand MIRA to newer audiences, experts develop conversational pathways specific to the unique requirements of a particular audience. The chatbot builder platform supports multidisciplinary collaboration by making it easier and faster to update and improve the conversation flow iteratively. To effectively extend MIRA's functionalities, we address the limitations identified in the pilot version. Our primary challenge was adding new features and capabilities without jeopardizing the existing conversational flow and functionality for healthcare workers. To achieve this, we transitioned away from a rigid and hard-coded flow towards a more modular approach. We also improved the reliability of transitions between the various conversational models. Addressing these limitations was crucial to set the stage for the development of the chatbot builder.

The chatbot builder resulted in greater flexibility in conversation design and easier integration of new pathways and modifications to existing ones. The chatbot builder removed the dependence on Rasa's rule-based and hard-coded custom actions, making it useful for any Rasa-based chatbot. We also developed a dynamic flow visualizer to assist the subject matter experts in visualizing the conversation flow as a directed graph. In conclusion, the chatbot builder helped facilitate the expansion of the scope of MIRA beyond healthcare workers, broadening its accessibility and impact. This expansion has streamlined the development cycle, enabling quicker deployment of changes to MIRA. Our platform is constantly evolving to meet the requirements and needs of our team. In the future, we plan to collect further feedback from users about their

satisfaction levels in modifying and designing chatbots from scratch within the framework. We aim to launch an open-source chatbot builder suitable for both Rasa-based and NLP-driven chatbots. This will help enable the creation of more advanced, less predefined, natural language-driven chatbots.

6.3 Exploring LLMs for MIRA: Benefits and Risks

While rule-based conversational structures provide a safe and controlled environment for conversations about mental health, they are inherently limited in their ability to capture the broad spectrum of user intents. Rule-based conversation flows cannot capture nuances in user expressions, handle complex or multi-faceted queries, or adapt to evolving user needs and language patterns.

Continuing to add intents to address edge cases to improve the chatbot's responsiveness can result in an increasingly complex rule-based tree. This is time-consuming for both development and maintenance. Another area where MIRA falls short is in its ability to provide empathetic interactions, integrating emotional dialogue can support users through more compassionate interactions. To address these limitations, more sophisticated conversational models are necessary. Future directions for MIRA could adopt a hybrid approach that combines the structured rule-based methodology with generative LLMs. While generative models can vastly improve the conversational experience, they come with high-risks, including hallucinations or offering advice that could be problematic or harmful to the user. Ensuring user safety requires rigorous study and implementation of strict guardrails to mitigate these risks and maintain ethical standards. MIRA can help raise awareness through information about mental health programs and services. Through MIRA, we hope to improve accessibility by providing quality controlled information on mental health services and programs tailored to individual queries.

References

[1] R. Alonzo, J. Hussain, S. Stranges, and K. K. Anderson, "Interplay between social media use, sleep quality, and mental health in youth: A systematic review," *Sleep medicine reviews*, vol. 56, p. 101 414, 2021.

[2] S. Y. Sohn, P. Rees, B. Wildridge, N. J. Kalk, and B. Carter, "Prevalence of problematic smartphone usage and associated mental health outcomes amongst children and young people: A systematic review, meta-analysis and grade of the evidence," *BMC psychiatry*, vol. 19, pp. 1–10, 2019.

[3] P. D. McGorry, C. Mei, A. Chanen, C. Hodges, M. Alvarez-Jimenez, and E. Killackey, "Designing and scaling up integrated youth mental health care," *World Psychiatry*, vol. 21, no. 1, pp. 61–76, 2022.

[4] N. Vindegaard and M. E. Benros, "Covid-19 pandemic and mental health consequences: Systematic review of the current evidence," *Brain, behavior, and immunity*, vol. 89, pp. 531–542, 2020.

[5] E. K. Jenkins, C. McAuliffe, S. Hirani, *et al.*, "A portrait of the early and differential mental health impacts of the covid-19 pandemic in canada: Findings from the first wave of a nationally representative cross-sectional survey," *Preventive medicine*, vol. 145, p. 106 333, 2021.

[6] A. C. Gadermann, K. C. Thomson, C. G. Richardson, *et al.*, "Examining the impacts of the covid-19 pandemic on family mental health in canada: Findings from a national cross-sectional study," *BMJ open*, vol. 11, no. 1, e042871, 2021.

[7] W. Cullen, G. Gulati, and B. D. Kelly, "Mental health in the covid-19 pandemic," *QJM: An International Journal of Medicine*, vol. 113, no. 5, pp. 311–312, 2020.

[8] M. L. Wainberg, P. Scorza, J. M. Shultz, *et al.*, "Challenges and opportunities in global mental health: A research-to-practice perspective," *Current psychiatry reports*, vol. 19, pp. 1–10, 2017.

[9] Statistics Canada, *Mental health care needs, 2018*, Available at: `https://www150.statcan.gc.ca/n1/pub/82-625-x/2019001/article/00011-eng.htm`, 2019.

[10] P. Corrigan, "How stigma interferes with mental health care.," *American psychologist*, vol. 59, no. 7, p. 614, 2004.

[11] *Introduction to Rasa Open Source & Rasa Pro — rasa.com*, https://rasa.com/docs/rasa/, [Accessed 20-05-2024].

[12] *Intelligent Diagramming — Lucidchart — lucidchart.com*, https://www.lucidchart.com/, [Accessed 02-07-2024].

[13] P. Smetanin, C. Briante, D. Stiff, S. Ahmad, and M. Khan, *The life and economic impact of major mental illnesses in Canada*. Mental Health Commission of Canada, 2015.

[14] E. Stephenson, "Mental disorders and access to mental health care," 2023.

[15] R. O'sullivan, A. Burns, G. Leavey, *et al.*, "Impact of the covid-19 pandemic on loneliness and social isolation: A multi-country study," *International journal of environmental research and public health*, vol. 18, no. 19, p. 9982, 2021.

[16] M. Bower, S. Smout, A. Donohoe-Bales, *et al.*, "A hidden pandemic? an umbrella review of global evidence on mental health in the time of covid-19," *Frontiers in Psychiatry*, vol. 14, p. 1 107 560, 2023.

[17] *Media Centre: Media Releases — Mental Health Research Canada — mhrc.ca*, https://www.mhrc.ca/media-centre/media-releases, [Accessed 02-06-2024].

[18] J. M. Haddad, C. Macenski, A. Mosier-Mills, *et al.*, "The impact of social media on college mental health during the covid-19 pandemic: A multinational review of the existing literature," *Current psychiatry reports*, vol. 23, pp. 1–12, 2021.

[19] J. Gao, P. Zheng, Y. Jia, *et al.*, "Mental health problems and social media exposure during covid-19 outbreak," *Plos one*, vol. 15, no. 4, e0231924, 2020.

[20] C. Heritage, *Statistics on official languages in Canada - Canada.ca — canada.ca*, https://www.canada.ca/en/canadian-heritage/services/official-languages-bilingualism/publications/statistics.html, [Accessed 04-06-2024].

[21] N. Schmitz, P. Holley, X. Meng, L. Fish, and J. Jedwab, "Covid-19 and depressive symptoms: A community-based study in quebec, canada," *The Canadian Journal of Psychiatry*, vol. 65, no. 10, pp. 733–735, 2020.

[22] Statistics Canada, *Table 13-10-0763-01 health characteristics of children and youth aged 1 to 17 years, canadian health survey on children and youth 2019*, Statistics Canada, 2019.

[23] A. J. Browne, C. M. Varcoe, S. T. Wong, *et al.*, "Closing the health equity gap: Evidence-based strategies for primary health care organizations," *International journal for equity in health*, vol. 11, pp. 1–15, 2012.

[24] S. De Leeuw, *Determinants of Indigenous Peoples' Health*. Canadian Scholars' Press, 2015.

[25] S. E. Nelson and K. Wilson, "The mental health of indigenous peoples in canada: A critical review of research," *Social Science & Medicine*, vol. 176, pp. 93–112, 2017.

[26] J. Borghouts, E. Eikey, G. Mark, *et al.*, "Barriers to and facilitators of user engagement with digital mental health interventions: Systematic review," *Journal of medical Internet research*, vol. 23, no. 3, e24387, 2021.

[27] D. C. Mohr, M. N. Burns, S. M. Schueller, G. Clarke, and M. Klinkman, "Behavioral intervention technologies: Evidence review and recommendations for future research in mental health," *General hospital psychiatry*, vol. 35, no. 4, pp. 332–338, 2013.

[28] C. A. Webb, I. M. Rosso, and S. L. Rauch, "Internet-based cognitive-behavioral therapy for depression: Current progress and future directions," *Harvard review of psychiatry*, vol. 25, no. 3, pp. 114–122, 2017.

[29] S. De Hert, "Burnout in healthcare workers: Prevalence, impact and preventative strategies," *Local and regional anesthesia*, pp. 171–183, 2020.

[30] J. A. Himle, A. Weaver, A. Zhang, and X. Xiang, "Digital mental health interventions for depression," *Cognitive and Behavioral Practice*, vol. 29, no. 1, pp. 50–59, 2022.

[31] *Siri - apple*, `https://www.apple.com/ca/siri/`, [Accessed 20-05-2024].

[32] *You've received an Alexa Link — alexa.amazon.com*, `https://alexa.amazon.com`, [Accessed 20-05-2024].

[33] *Chatgpt*, `https://openai.com/chatgpt/`, [Accessed 20-05-2024].

[34] C.-H. Li, S.-F. Yeh, T.-J. Chang, M.-H. Tsai, K. Chen, and Y.-J. Chang, "A conversation analysis of non-progress and coping strategies with a banking task-oriented chatbot," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–12.

[35] W.-Y. Chang and Y.-N. Chen, "Salesbot 2.0: A human-like intent-guided chit-chat dataset," *arXiv preprint arXiv:2308.14266*, 2023.

[36] S. A. Thorat and V. Jadhav, "A review on implementation issues of rule-based chatbot systems," in *Proceedings of the international conference on innovative computing & communications (ICICC)*, 2020.

[37] *Microsoft Copilot: Your everyday AI companion — copilot.microsoft.com*, `https://copilot.microsoft.com`, [Accessed 20-05-2024].

[38] *Perplexity ai*, `https://www.perplexity.ai`, [Accessed 20-05-2024].

[39] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," *Advances in neural information processing systems*, vol. 30, 2017.

[40] S. Wu, H. Fei, L. Qu, W. Ji, and T.-S. Chua, "Next-gpt: Any-to-any multimodal llm," *arXiv preprint arXiv:2309.05519*, 2023.

[41] Y. Talebirad and A. Nadiri, "Multi-agent collaboration: Harnessing the power of intelligent llm agents," *arXiv preprint arXiv:2306.03314*, 2023.

[42] A. J. Thirunavukarasu, D. S. J. Ting, K. Elangovan, L. Gutierrez, T. F. Tan, and D. S. W. Ting, "Large language models in medicine," *Nature medicine*, vol. 29, no. 8, pp. 1930–1940, 2023.

[43] S. Wu, O. Irsoy, S. Lu, *et al.*, "Bloomberggpt: A large language model for finance," *arXiv preprint arXiv:2303.17564*, 2023.

[44] M. Chen, J. Tworek, H. Jun, *et al.*, "Evaluating large language models trained on code," *arXiv preprint arXiv:2107.03374*, 2021.

[45] M. U. Hadi, R. Qureshi, A. Shah, *et al.*, "A survey on large language models: Applications, challenges, limitations, and practical usage," *Authorea Preprints*, 2023.

[46] V. Rawte, A. Sheth, and A. Das, "A survey of hallucination in large foundation models," *arXiv preprint arXiv:2309.05922*, 2023.

[47] A. Shawar, E. Atwell, and A. Roberts, "Faqchat as in information retrieval system," in *Human Language Technologies as a Challenge for Computer Science and Linguistics: Proceedings of the 2nd Language and Technology Conference*, Poznań: Wydawnictwo Poznańskie: with co-operation of Fundacja Uniwersytetu ..., 2005, pp. 274–278.

[48] C. Zhai and S. Massung, "Text data management and analysis: A practical introduction to ir and text mining," *ACM and Morgan*, 2016.

[49] M. Karan and J. Šnajder, "Paraphrase-focused learning to rank for domain-specific frequently asked questions retrieval," *Expert Systems with Applications*, vol. 91, pp. 418–433, 2018.

[50] R. Shrivastava, A. Pujahari, S. P. Singh, and T. Bhowmik, "Efficient question answering in chatbot using tf-idf and cosine similarity," in *Innovations in Information and Communication Technologies: Proceedings of ICIICT 2022*, Springer, 2022, pp. 25–31.

[51] Z. Yan, N. Duan, J. Bao, *et al.*, "Docchat: An information retrieval approach for chatbot engines using unstructured documents," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 516–525.

[52] M. Hardalov, I. Koychev, and P. Nakov, "Machine reading comprehension for answer re-ranking in customer support chatbots," *Information*, vol. 10, no. 3, p. 82, 2019.

[53] A. Lommatzsch and J. Katins, "An information retrieval-based approach for building intuitive chatbots for large knowledge bases.," in *LWDA*, 2019, pp. 343–352.

[54] J. Gao, C. Xiong, and P. Bennett, "Recent advances in conversational information retrieval," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 2421–2424.

[55] Y. Mass, B. Carmeli, H. Roitman, and D. Konopnicki, "Unsupervised faq retrieval with question generation and bert," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 807–812.

[56] W. Sakata, T. Shibata, R. Tanaka, and S. Kurohashi, "Faq retrieval using query-question similarity and bert-based query-answer relevance," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019, pp. 1113–1116.

[57] P. Lewis, E. Perez, A. Piktus, *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.

[58] A. Følstad, M. Skjuve, and P. B. Brandtzaeg, "Different chatbots for different purposes: Towards a typology of chatbots to understand interaction design," in *Internet Science: INSCI 2018 International Workshops, St. Petersburg, Russia, October 24–26, 2018, Revised Selected Papers 5*, Springer, 2019, pp. 145–156.

[59] R. Agarwal and M. Wadhwa, "Review of state-of-the-art design techniques for chatbots," *SN Computer Science*, vol. 1, no. 5, p. 246, 2020.

[60] S.-F. Yeh, M.-H. Wu, T.-Y. Chen, *et al.*, "How to guide task-oriented chatbot users, and when: A mixed-methods study of combinations of chatbot guidance types and timings," in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 2022, pp. 1–16.

[61] S. Stumpf, V. Rajaram, L. Li, *et al.*, "Interacting meaningfully with machine learning systems: Three experiments," *International journal of human-computer studies*, vol. 67, no. 8, pp. 639–662, 2009.

[62] J. van der Waa, E. Nieuwburg, A. Cremers, and M. Neerincx, "Evaluating xai: A comparison of rule-based and example-based explanations," *Artificial intelligence*, vol. 291, p. 103 404, 2021.

[63] *An Architectural Overview of Task-Oriented Dialog Systems (TODS) - Building Effective Virtual Agent Chatbots — linkedin.com*, `https://www.linkedin.com/pulse/architectural-overview-task-oriented-dialog-systems-tods-mendiratta/`, [Accessed 08-06-2024].

[64] W. Maeng and J. Lee, "Designing a chatbot for survivors of sexual violence: Exploratory study for hybrid approach combining rule-based chatbot and ml-based chatbot," in *Proceedings of the Asian CHI Symposium 2021*, 2021, pp. 160–166.

[65] S. Verma, J. Fu, M. Yang, and S. Levine, "Chai: A chatbot ai for task-oriented dialogue with offline reinforcement learning," *arXiv preprint arXiv:2204.08426*, 2022.

[66] F. Radlinski and N. Craswell, "A theoretical framework for conversational search," in *Proceedings of the 2017 conference on conference human information interaction and retrieval*, 2017, pp. 117–126.

[67] J. Meredith, "Analysing technological affordances of online interactions using conversation analysis," *Journal of pragmatics*, vol. 115, pp. 42–55, 2017.

[68] Z. Ashktorab, M. Jain, Q. V. Liao, and J. D. Weisz, "Resilient chatbots: Repair strategy preferences for conversational breakdowns," in *Proceedings of the 2019 CHI conference on human factors in computing systems*, 2019, pp. 1–12.

[69] S. Chiu, M. Li, Y.-T. Lin, and Y.-N. Chen, "Salesbot: Transitioning from chit-chat to task-oriented dialogues," *arXiv preprint arXiv:2204.10591*, 2022.

[70] J. Pereira and Ó. Díaz, "Chatbot dimensions that matter: Lessons from the trenches," in *Web Engineering: 18th International Conference, ICWE 2018, Cáceres, Spain, June 5-8, 2018, Proceedings 18*, Springer, 2018, pp. 129–135.

[71] I. Dagkoulis and L. Moussiades, "A comparative evaluation of chatbot development platforms," in *Proceedings of the 26th Pan-Hellenic Conference on Informatics*, 2022, pp. 322–328.

[72] *Compare Chatbot Platforms and Pick the Best One for You — chatbot.com*, `https://www.chatbot.com/blog/chatbot-platforms/`, [Accessed 10-06-2024].

[73] *Chatbot Development Tools Choosing the Right Platform for Your Business — MoldStud — moldstud.com*, `https://moldstud.com/articles/p-chatbot-development-tools-choosing-the-right-platform-for-your-business`, [Accessed 10-06-2024].

[74] S. Srivastava and T. Prabhakar, "Desirable features of a chatbot-building platform," in *2020 IEEE International Conference on Humanized Computing and Communication with Artificial Intelligence (HCCAI)*, IEEE, 2020, pp. 61–64.

[75] S. Pérez-Soler, S. Juarez-Puerta, E. Guerra, and J. de Lara, "Choosing a chatbot development tool," *IEEE Software*, vol. 38, no. 4, pp. 94–103, 2021.

[76] *Conversational Interfaces Community Group — w3.org*, `https://www.w3.org/community/conv/`, [Accessed 10-06-2024].

[77] M. Milne-Ives, C. de Cock, E. Lim, *et al.*, "The effectiveness of artificial intelligence conversational agents in health care: Systematic review," *Journal of medical Internet research*, vol. 22, no. 10, e20346, 2020.

[78] A. Alattas, G. Teepe, K. Leidenberger, *et al.*, "To what scale are conversational agents used by top-funded companies offering digital mental health services for depression?" In *Proceedings of the 14th International Joint Conference on Biomedical Engineering Systems and Technologies*, Science and Technology Publications, Lda, vol. 5, 2021, pp. 801–808.

[79] S. M. Lim, C. W. C. Shiau, L. J. Cheng, and Y. Lau, "Chatbot-delivered psychotherapy for adults with depressive and anxiety symptoms: A systematic review and meta-regression," *Behavior Therapy*, vol. 53, no. 2, pp. 334–347, 2022.

[80] S. Mujeeb, M. H. Javed, and T. Arshad, "Aquabot: A diagnostic chatbot for achluophobia and autism," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 9, pp. 209–216, 2017.

[81] S. Jang, J.-J. Kim, S.-J. Kim, J. Hong, S. Kim, and E. Kim, "Mobile app-based chatbot to deliver cognitive behavioral therapy and psychoeducation for adults with attention deficit: A development and feasibility/usability study," *International journal of medical informatics*, vol. 150, p. 104 440, 2021.

[82] C. Wibhowo and R. Sanjaya, "Virtual assistant to suicide prevention in individuals with borderline personality disorder," in *2021 International Conference on Computer & Information Sciences (ICCOINS)*, IEEE, 2021, pp. 234–237.

[83] M. Lee, S. Ackermans, N. Van As, H. Chang, E. Lucas, and W. IJsselsteijn, "Caring for vincent: A chatbot for self-compassion," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–13.

[84] S. Gabrielli, S. Rizzi, S. Carbone, V. Donisi, *et al.*, "A chatbot-based coaching intervention for adolescents to promote life skills: Pilot study," *JMIR human factors*, vol. 7, no. 1, e16762, 2020.

[85] E. M. Boucher, N. R. Harake, H. E. Ward, *et al.*, "Artificially intelligent chatbots in digital mental health interventions: A review," *Expert Review of Medical Devices*, vol. 18, no. sup1, pp. 37–49, 2021.

[86] C. Sweeney, C. Potts, E. Ennis, *et al.*, "Can chatbots help support a person's mental health? perceptions and views from mental healthcare professionals and experts," *ACM Transactions on Computing for Healthcare*, vol. 2, no. 3, pp. 1–15, 2021.

[87] R. Fulmer, A. Joerin, B. Gentile, L. Lakerink, M. Rauws, *et al.*, "Using psychological artificial intelligence (tess) to relieve symptoms of depression and anxiety: Randomized controlled trial," *JMIR mental health*, vol. 5, no. 4, e9782, 2018.

[88] T. Kamita, T. Ito, A. Matsumoto, T. Munakata, T. Inoue, *et al.*, "A chatbot system for mental healthcare based on sat counseling method," *Mobile Information Systems*, vol. 2019, 2019.

[89] O. Perski, D. Crane, E. Beard, and J. Brown, "Does the addition of a supportive chatbot promote user engagement with a smoking cessation app? an experimental study," *Digital health*, vol. 5, p. 2 055 207 619 880 676, 2019.

[90] K. Kypri and H. M. McAnally, "Randomized controlled trial of a web-based primary care intervention for multiple health risk behaviors," *Preventive medicine*, vol. 41, no. 3-4, pp. 761–766, 2005.

[91] G. M. Lucas, A. Rizzo, J. Gratch, *et al.*, "Reporting mental health symptoms: Breaking down barriers to care with virtual human interviewers," *Frontiers in Robotics and AI*, vol. 4, p. 51, 2017.

[92] A. A. Abd-Alrazaq, M. Alajlani, A. A. Alalwan, B. M. Bewick, P. Gardner, and M. Househ, "An overview of the features of chatbots in mental health: A scoping review," *International journal of medical informatics*, vol. 132, p. 103 978, 2019.

[93] L. Laranjo, A. G. Dunn, H. L. Tong, *et al.*, "Conversational agents in healthcare: A systematic review," *Journal of the American Medical Informatics Association*, vol. 25, no. 9, pp. 1248–1258, 2018.

[94] *Dialogflow — Google Cloud — cloud.google.com*, `https://cloud.google.com/dialogflow`, [Accessed 28-05-2024].

[95] *AI Chatbot - Amazon Lex - AWS — aws.amazon.com*, `https://aws.amazon.com/lex/`, [Accessed 28-05-2024].

[96] *IBM Watson — ibm.com*, `https://www.ibm.com/watson`, [Accessed 28-05-2024].

[97] *Microsoft Bot Framework — dev.botframework.com*, `https://dev.botframework.com`, [Accessed 28-05-2024].

[98] *Botkit is an open source developer tool for building chat bots, apps and custom integrations for major messaging platforms. — github.com*, `https://github.com/howdyai/botkit`, [Accessed 28-05-2024].

[99] *DeepPavlov: An open source conversational AI framework — deeppavlov.ai*, `https://deeppavlov.ai`, [Accessed 28-05-2024].

[100] A. Zamani, "Developing a mental health virtual assistance (chatbot) for healthcare workers and their families," 2022.

[101] M. A. Gharaat, "Finding reliable resources and chatting with mira while considering emotions when the scenario is unscripted," 2024.

[102] 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.

[103] T. Bunk, D. Varshneya, V. Vlasov, and A. Nichol, "Diet: Lightweight language understanding for dialogue systems," *arXiv preprint arXiv:2004.09936*, 2020.

[104] *Conversational AI Platform — Superior Customer Experiences Start Here — rasa.com*, `https://rasa.com`, [Accessed 28-05-2024].

[105] *Rasa Architecture Overview — rasa.com*, `https://rasa.com/docs/rasa/arch-overview/`, [Accessed 02-07-2024].

[106] A. Zamani, M. Reeson, T. Marshall, *et al.*, "Intent and entity detection with data augmentation for a mental health virtual assistant chatbot," in *Proceedings of the 23rd ACM International Conference on Intelligent Virtual Agents*, 2023, pp. 1–4.

[107] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[108] 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.

[109] M. Henderson, I. Casanueva, N. Mrkšić, P.-H. Su, T.-H. Wen, and I. Vulić, "Convert: Efficient and accurate conversational representations from transformers," *arXiv preprint arXiv:1911.03688*, 2019.

[110] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[111] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," *arXiv preprint arXiv:1803.02155*, 2018.

[112] J. Lafferty, A. McCallum, F. Pereira, *et al.*, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Icml*, Williamstown, MA, vol. 1, 2001, p. 3.

[113] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," *arXiv preprint arXiv:1603.01360*, 2016.

[114] V. Vlasov, J. E. Mosig, and A. Nichol, "Dialogue transformers," *arXiv preprint arXiv:1910.00486*, 2019.

[115] J. Wei and K. Zou, "Eda: Easy data augmentation techniques for boosting performance on text classification tasks," *arXiv preprint arXiv:1901.11196*, 2019.

[116] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller, "Introduction to wordnet: An on-line lexical database," *International journal of lexicography*, vol. 3, no. 4, pp. 235–244, 1990.

[117] O. Kolomiyets, S. Bethard, and M.-F. Moens, "Model-portability experiments for textual temporal analysis," in *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, ACL; East Stroudsburg, PA, vol. 2, 2011, pp. 271–276.

[118] A. Harris and S. H. Jones, "Words," in *Writing for Performance*, Brill, 2016, pp. 19–35.

[119] M. Bayer, M.-A. Kaufhold, and C. Reuter, "A survey on data augmentation for text classification," *ACM Computing Surveys*, vol. 55, no. 7, pp. 1–39, 2022.

[120] K. K. Fitzpatrick, A. Darcy, and M. Vierhile, "Delivering cognitive behavior therapy to young adults with symptoms of depression and anxiety using a fully automated conversational agent (woebot): A randomized controlled trial," *JMIR mental health*, vol. 4, no. 2, e7785, 2017.

[121] B. Inkster, S. Sarda, V. Subramanian, *et al.*, "An empathy-driven, conversational artificial intelligence agent (wysa) for digital mental well-being: Real-world data evaluation mixed-methods study," *JMIR mHealth and uHealth*, vol. 6, no. 11, e12106, 2018.