applied sciences

MDPI

Article

Tayseer: A Novel AI-Powered Arabic Chatbot Framework for Technical and Vocational Student Helpdesk Services and Enhancing Student Interactions

Abeer Alabbas [1,*] and Khalid Alomar [2]

[1] Faculty of Computer and Information Technology, Technical and Vocational Training Corporation, Najran 66253, Saudi Arabia

[2] Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia; khalomar@kau.edu.sa

* Correspondence: aalabbas3@tvtc.gov.sa

Abstract: The rise of conversational agents (CAs) like chatbots in education has increased the demand for advisory services. However, student–college admission interactions remain manual and burdensome for staff. Leveraging CAs could streamline the admission process, providing efficient advisory support. Moreover, limited research has explored the role of Arabic chatbots in education. This study introduces Tayseer, an Arabic AI-powered web chatbot that enables instant access to college information and communication between students and colleges. This study aims to improve the abilities of chatbots by integrating features into one model, including responding with audiovisuals, various interaction modes (menu, text, or both), and collecting survey responses. Tayseer uses deep learning models within the RASA framework, incorporating a customized Arabic natural language processing pipeline for intent classification, entity extraction, and response retrieval. Tayseer was deployed at the Technical College for Girls in Najran (TCGN). Over 200 students used Tayseer during the first semester, demonstrating its efficiency in streamlining the advisory process. It identified over 50 question types from inputs with a 90% precision in intent and entity predictions. A comprehensive evaluation illuminated Tayseer's proficiency as well as areas requiring improvement. This study developed an advanced CA to enhance student experiences and satisfaction while establishing best practices for education chatbot interfaces by outlining steps to build an AI-powered chatbot from scratch using techniques adaptable to any language.

Keywords: Arabic chatbot; RASA; transformer model; DIET; vocational education

check for updates

Citation: Alabbas, A.; Alomar, K. Tayseer: A Novel AI-Powered Arabic Chatbot Framework for Technical and Vocational Student Helpdesk Services and Enhancing Student Interactions. *Appl. Sci.* **2024**, *14*, 2547. https://doi.org/10.3390/app14062547

Academic Editors: Yu Liang, Wenjun Wu and Ying Li

Received: 4 March 2024
Revised: 13 March 2024
Accepted: 14 March 2024
Published: 18 March 2024

1. Introduction

Computer programs utilizing NLP have emerged as pivotal tools for simulating human-like conversations in the evolving landscape of technology, chatbots, and virtual assistants [1]. These AI-driven entities have found widespread application in automating customer support and services [1]. The integration of artificial intelligence (AI) across various computing and technological domains [2] notably includes the field of NLP [3]. This branch of AI facilitates human–computer interaction and communication through natural language, enabling chatbots to converse with humans or other bots via text or speech [4].

The utility of chatbots extends across diverse sectors, including e-commerce [5] and education [6–10], with notable applications in customer service [5,11,12] and hospital patient counseling [13–15]. Institutions increasingly rely on corporate websites and social media for student services in the educational domain by utilizing web-based response systems [4]. However, these methods are resource-intensive and time-consuming, leading to delays in response times [5]. This issue has been further compounded by the shift towards online education during the pandemic, intensifying the demand for efficient online assistance and round-the-clock service availability [16]. The growing reliance on intelligent agents, such as chatbots, in daily life makes their integration into social media and instant messaging ecosystems increasingly commonplace and rational [17].

Conversational agents have garnered considerable interest in the education sector because of their capacity to fulfill several educational functions, such as acting as instructors, coaches, and learning companions [18]. These agents possess the ability to analyze and react to human language. These have been incorporated into educational computer systems to facilitate dynamic learning interactions with students [19,20]. Implementing AI-powered chatbots in educational settings can significantly reduce administrative burden and enhance the student experience, potentially boosting student retention rates and satisfaction [6].

This study proposes the Tayseer architecture to explore this field by utilizing advanced technological innovations such as NLP, dialogue systems, and sentiment analysis, all finely tuned for the Arabic language. This study introduces a dual-mode conversational architecture that synergizes NLP capabilities with different interaction methods (write-based, click-based, or both), specifically catering to technical students' needs. Employing the RASA framework, the system adeptly handles open-ended dialogues, whereas the menu option paradigm offers structured guidance for various tasks. This dual approach aims to provide flexible self-service channels that accommodate user preferences, regardless of whether they favor structured or unstructured interactions.

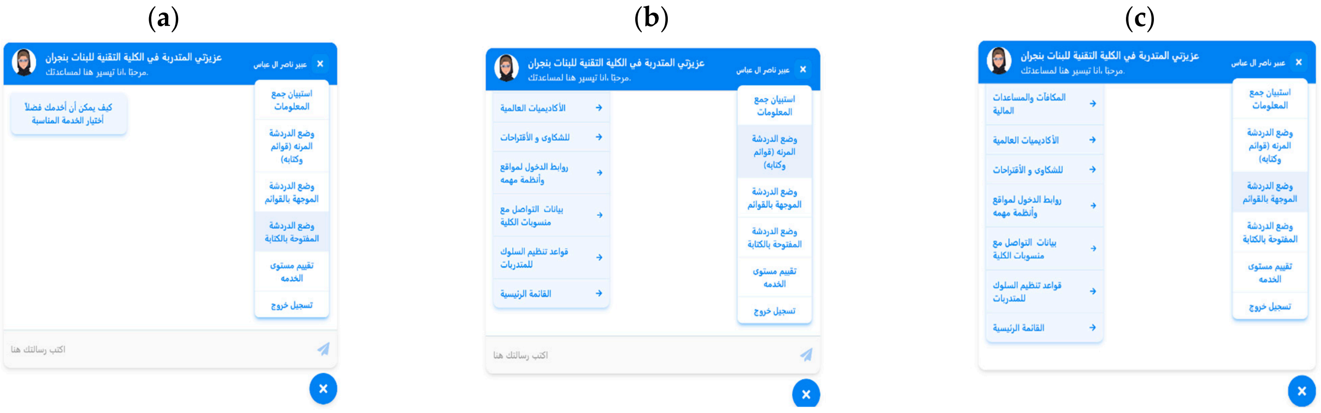
Developing effective chatbots requires a solid understanding of deep learning (DL), a subset of ML within AI [17]. DL, which is often associated with deep artificial neural networks, involves sophisticated algorithms that model data through nonlinear function transformations across multiple layers [21]. Deep learning chatbots are primarily categorized as retrieval-based and generative bots. Retrieval-based bots leverage classification models to discern user intent and retrieve suitable responses from a database, offering greater flexibility than rule-based systems. In contrast, generative bots generate responses based on both current and previous user interactions, displaying more human-like characteristics but often struggling with grammatical accuracy and consistency [22–24].

Focusing on the RASA framework, a key component in chatbot development comprises natural language understanding (NLU) and dialogue management elements. RASA NLU enables customizable NLP through various prebuilt components, including the Dual Intent and Entity Transformer (DIET) [7]. DIET, a transformer-based model, efficiently categorizes intents and extracts entities, which is compatible with pre-trained embedding models such as BERT, GloVe, and ConveRT [8]. However, there is a notable research gap in the application of the RASA framework to the Arabic language, which is characterized by complex syntax and rich morphology, presenting unique challenges for NLP [25–27].

Tayseer was developed using the DIET architecture with attention-based mechanisms to address these challenges, as shown in Figure 1. The integration of pre-trained models within DIET enables the chatbot to effectively handle intent and entity recognition tasks [28–30].

Moreover, the system was designed to securely store individual academic backgrounds and dialogue histories in a structured SQL database, facilitating personalized and context-aware responses. An integrated feedback loop leveraging sentiment analysis of student conversational experiences was employed to incrementally improve the system's effectiveness over time.

This applied research seeks to contribute to the field by integrating data-driven machine-learning methods with custom knowledge representation within a modular Arabic chatbot framework. The framework was specifically designed to address admission inquiries and frequently asked questions (FAQs) in the context of higher education for technical students. The findings of this study are intended to inform the ongoing development of effective and user-friendly conversational interfaces for the Arabic-speaking student population in line with the identified key features and design principles.


(**a**)
(**b**)
(**c**)
عبير ناصر ال عباس
استبيان جمع المعلومات
وضع الدردشة المرنه (قوائم وكتابه)
وضع الدردشة الموجهة بالقوائم
وضع الدردشة المفتوحة بالكتابة
تقييم مستوى الخدمه
تسجيل خروج
المكافآت والمساعدات المالية
الأكاديميات العالمية
للشكاوى و الأقتراحات
روابط الدخول لمواقع وأنظمة مهمه
بيانات التواصل مع منسوبات الكلية
قواعد تنظيم السلوك للمتدربات
القائمة الرئيسية
اكتب رسالتك هنا

عزيزتي المتدربة في الكلية التقنية للبنات بنجران
مرحبًا ،انا تيسير هنا لمساعدتك.

كيف يمكن أن أخدمك فضلاً أختيار الخدمة المناسبة

write-based menu-based and write-based menu-based

Figure 1. Interactions of different interface methods: (**a**) write-based; (**b**) menu-based and write-based; (**c**) menu-based.

The rest of the paper is structured as follows: Section 2 offers a comprehensive overview of pertinent prior research, providing a background for our study. Section 3 presents the methodology and the proposed model. Section 4 presents and discusses the results and highlights the key findings. Finally, Section 5 concludes the paper and offers recommendations for future research.

2. Related Work

2.1. Chatbot Overview

ELIZA [31] and PARRY [32] were among the first chatbots. These systems do not employ data-driven learning but rely on a combination of rules and patterns; they work exceptionally well when the scope of the discussion is defined, i.e., when the conversation is focused on a specific topic or task [4]. However, learning-based alternatives have been developed to circumvent some of the drawbacks of rule-based (handwritten rules) systems. These technologies enable chatbots to learn from massive volumes of available human-to-human conversations, for example, chat platforms, Twitter, or movie dialogues.

Learning-based approaches can be broadly characterized as retrieval-based or generative-based approaches. Retrieval-based models collect candidate answers from a prebuilt index, rank the candidates, and select the answer from the top-ranked options [22,33]. In contrast, generation-based approaches typically generate answers via deep learning (DL) techniques [17]. The basic premise of generative-based approaches is to create a new sentence word by word in response to a user's query [34].

Ritter et al. [35] employed phrase-based statistical machine translation (SMT) to convert a user query into a system response. They showed that the SMT method was superior to retrieval-based models in generating responses to a Twitter dataset [35,36]. Subsequently, it became apparent that the problem of generating responses differed from that of machine translation [37]. While words or phrases in the source and target sentences usually match well, a user's speech in a coherent response may not have any words or phrases in common with a coherent response [38].

It was later shown that another machine translation technique, sequence-to-sequence (seq2seq) learning, performed better at generating responses [38]. The seq2seq architecture includes an encoder model that compresses user input (query) and represents it as a vector [24]. A decoder model decodes the vector (representing the encoded input) and generates sentences word by word [24]. Typically, the encoder and decoder models are based on RNNs. Attention-based methods have been developed to allow the encoder to concentrate on certain portions of the encoded input, which is crucial for forecasting specific segments of the output during the decoding phase [39]. Bidirectional RNN (BRNN) models have also been used to improve the encoding of the context of the output sentence.

A new, simple network design based on the attention mechanism was later introduced, resulting in significantly improved response quality while requiring significantly less training time [24]. This type of design is called the transformer architecture [40]. The transformer model establishes global relationships between the input and output using only an attention mechanism [24]. This prevents recursion and allows for greater parallelization; however, the transformer model retains the encoder–decoder architecture. The development of transformer-based language models was a turning point in NLP [41]. These models, such as OpenAI GPT3, GPT4 [42], Bert [43], and T5 [44], have achieved state-of-the-art performance in a variety of NLP tasks [41].

2.2. RASA Framework for AI Chatbots

RASA has gained significant popularity as a platform for building AI chatbots across various disciplines [45]. RASA, an open-source framework based on machine learning, has proven its ability to reliably detect user intentions, extract entities, and effectively handle contextual nuances [45,46]. Researchers have examined the implementation of RASA in diverse domains, including academic research, tourism, mental health, and customer service [45,46].

The RASA framework has become a prominent tool in Arabic language processing. In addition, incorporating both morphological and syntactic disambiguation in a unified framework has shown highly favorable outcomes for languages such as Arabic [47]. An integrated strategy is essential for efficiently handling Arabic text and comprehending user input within the context of a chatbot. In addition, incorporating both morphological and syntactic disambiguation in a unified framework has shown highly favorable outcomes for languages such as Arabic [47]. An integrated method is essential for efficiently processing Arabic text and comprehending user inputs within the context of a chatbot.

To address these issues, it is essential to utilize the available tools and technology specifically designed for Arabic language processing. The MADA tool, also known as morphological analysis and disambiguation for Arabic, is utilized to perform part-of-speech tagging, stemming, and lemmatization of Arabic text [48]. Moreover, advancements in Arabic conversational AI, such as chatbots, have broadened the range of text-based conversational AI systems [49].

Multiple studies have focused on the research and implementation of RASA chatbots in Arabic and other languages. Alruily [26] evaluated the performance of the ArRASA framework, which is a deep learning-based Arabic NLU chatbot framework developed using the RASA framework. This research demonstrates the potential of the RASA framework to produce accurate and effective chatbots in Arabic language settings. A survey by Al-Hagbani and Khan [50] emphasized the difficulties associated with creating Arabic chatbots, mainly due to the intricacy of the language. Notwithstanding these difficulties, RASA has been effectively implemented in several fields, including university [51,52] and college inquiry chatbots [53]. The combination of RASA NLU and neural network techniques has also been investigated to create an advanced chatbot system [25,54].

The creation of an Arabic RASA chatbot necessitates profound comprehension of the distinctive linguistic attributes of the Arabic language and the employment of specialized tools and technology specifically designed for processing the Arabic language. Utilizing open-source frameworks such as RASA and including morphological and syntactic disambiguation greatly enhances the creation of a proficient Arabic RASA chatbot [26,53].

Despite these studies, there is still a lack of research that specifically addresses the gaps in using the RASA framework for AI chatbots in Arabic. The existing literature highlights the scarcity of Arabic chatbots and the need for more research on Arabic chatbot development, linguistic complexities, and content availability [55,56].

2.3. Chatbots for Arabic Settings

Arabic is a complex language; therefore, creating Arabic chatbots has proven to be difficult for researchers [57]. To date, only a few researchers have attempted to develop

Arabic chatbots. According to a survey [55], Arabic chatbots are still in their infancy; all of the current work is rule-based or retrieval-based, and the lack of accessible datasets impedes their growth [25].

However, some studies have been able to overcome Arabic language limitations by using translation tools. For example, Mozannar et al. [56] developed a question-answering method, and Biltawi et al. [58] developed an Arabic language model. The authors of [57] developed MidoBot, a deep learning-based generative chatbot in Arabic that uses a sequence-to-sequence model to generate new responses from a dataset. The performance of these developments and that of [59] demonstrates the potential of neural models for understanding the Arabic language.

In addition, Eljundi et al. [60] highlighted the potential of neural models for Arabic language comprehension. This prompted an investigation of neural solutions for open tasks in Arabic, such as empathic response production. Recent advances in transformer-based models have shown that language-specific BERT-based models are incredibly efficient in understanding language, provided they are pre-trained on an extensive corpus [59]. These models have established new benchmarks and have achieved state-of-the-art results for most NLP tasks [41].

Recent research has focused entirely on data-driven end-to-end systems that use neural generative models to generate the appropriate responses [41]. The most popular choices are seq2seq and transformer, and pre-trained models [33,37] and GPT3 [42] are two examples of encoder–decoder models. To explore these models for Arabic, large corpora of conversations with verified and cleaned annotations are needed [44] by creating contextualized word embeddings, transformers, and pre-trained models that can compensate for data deficiencies [33].

Their low computational cost makes them suitable for small datasets that Arabic can handle. Recent contextualized embeddings can be taught on unannotated materials, thereby improving various Arabic NLP tasks [41]. Multilingual BERT [61], AraT5 [62], AraGPT2 [63], ARBERT & MARBERT [64], AraELECTRA [41], and AraBERT [59] are contextualized embedding models that support Arabic.

Controllability is another aspect that can be used to construct sound Arabic conversational AI systems [33]. Hybrid models are an excellent alternative to this, as they integrate the capabilities of retrieval and generative models, where returned answers can be matched with generated answers to help retrieval models find a better answer [37].

2.4. Arabic Chatbots in University Settings

Chatbots are becoming increasingly prevalent across various industries, including product/service domains, healthcare, medicine, and education. Two key factors driving chatbot popularity are advancements in artificial intelligence (AI) and the growth of mobile messaging apps [65]. Recent research reviews indicate a burgeoning focus on exploring chatbot applications for educational purposes [33].

Based on an analysis of 80 studies on educational chatbots, the primary uses identified were administrative services and teaching assistants for delivering information and enhancing student learning [66]. The highlighted advantages of educational chatbots include 24/7 availability, adaptability to learner needs, and individualized support.

Jooka [67] is a bilingual chatbot for education that aims to improve the university admission process in both Arabic and English. The authors of [68] developed a social chatbot called Nabiha. Nabiha acts as an academic advisor who can support conversations with students of the Information Technology Department, engage with them, and respond to their queries about course options or other academic matters at King Saud University using the Saudi Arabic dialect. Nabiha was created on the Pandorabots platform by combining a pattern-matching approach with Artificial Intelligence Markup Language (AIML).

LANA -I [69] is an Arabic chatbot designed to help young people with autism spectrum disorder (ASD) in their educational endeavors. The system is rule-based and performs its functions of pattern matching and text similarity. Jordanian universities have proposed [2] the development of an intelligent Arabic chatbot system to overcome the outbreak of the pandemic COVID-19 and its global consequences. This technology mainly promotes spoken Arabic through the Jordanian dialect among students, especially at Jordan's Al-Zaytoonah Private University. Regrettably, it was not stated whether the students had tried this method.

Labeeb [70] is another chatbot that helps students answer questions about education or academic rules. This chatbot uses the Wikipedia API to retrieve the first paragraph of each requested query in the form of XML documents. However, the authors did not elaborate on the NLP models they used. SeerahBot is an Arabic chatbot specializing in the biography of Prophet Muhammad [8]. It utilizes a retrieval technique incorporating 200 questions and answers, employing machine learning to identify semantic question similarity, thereby optimizing the match between inputs and intentions. However, the study did not address the performance outcomes of this approach. Alazzam's [71] work will primarily concentrate on developing a chatbot tailored to comply with educational regulations in the UAE.

In the context of university admission helpdesks and frequently asked questions (FAQs), chatbots can be used to answer frequently asked questions and provide information about various university-related topics, such as courses, admissions, facilities, and impending events [72]. Powered by artificial intelligence, these avatars can comprehend and communicate using natural language processing techniques [72]. Using machine learning algorithms and accessing vast amounts of data, they can perpetually learn and evolve [72].

Al-Madi et al. [73] proposed an intelligent Arabic chatbot system to alleviate the burden of admission departments in a separate study. The study demonstrated that the chatbot system could provide efficient and gratifying responses to the majority of user queries, indicating its potential to streamline the admission process and enhance the user experience [27].

While there is a growing body of research on AI chatbots for university admission helpdesks and FAQ inquiries, there are still gaps in our comprehension of their efficacy, particularly in Arabic-speaking environments. Future research should concentrate on designing and evaluating chatbot systems that can effectively address the requirements of diverse student populations and enhance the admission process. Guidelines and ethical considerations regarding the use of chatbots in academic settings should be investigated further.

It is important to note that not all chatbots are powered by AI. Rule-based chatbots, for instance, adhere to pre-programmed logic and have limited intelligence [72]. Frequently, they are used to respond to inquiries regarding delivery and transportation [72]. To date, Arabic chatbots in education have been rule-based, which means that there is a need to develop intelligent conversational agents (CAs) capable of meeting academic and practitioner expectations. Moreover, further research is needed to enhance conversational quality using advanced natural language processing techniques [33]. Conversely, chatbots powered by AI can comprehend and communicate in human language using natural language processing techniques [74]. They provide sophisticated features and respond to various inquiries.

However, there are still gaps in the literature regarding the use of AI chatbots in Arabic-speaking settings for admission helpdesks and FAQ inquiries [71,73]. Fuad and Al-Yahya [30] conducted an assessment of recent developments in Arabic conversational AI and identified the need for further research on building human-like Arabic conversational AI systems. Similarly, [52] examined the advancements and challenges of intelligent Arabic chatbots and emphasized the need to surmount the linguistic complexities of Arabic.

Although chatbots have been studied for their potential educational benefits, there is a lack of solutions that cater to the specific requirements of technical and vocational students speaking Arabic. Our research aims to help fill some of these gaps by leveraging state-of-the-art NLP to enrich conversational abilities. Using previous works as a springboard, our framework introduces novel features that emphasize personalization and emotional intelligence.

3. Method

Our study presents the methodology behind the development of “Tayseer”, an Arabic AI chatbot designed for educational support, including addressing inquiries on entrance admissions and frequently asked questions. This chatbot features both menu-based and write-based (RASA-based) interfaces to cater to diverse user preferences; it provides a structured navigation system through predefined options and a dynamic conversational interface for natural language interactions, as shown in Figure 1.

Menu-based interaction simplifies user navigation with predefined options, making it ideal for those preferring direct methods or finding natural language processing complex. This facilitates switching between RASA-based and menu-based interactions to suit user preferences, enhancing comfort and usability.

The write-based framework employs RASA, split into RASA NLU for understanding user inputs and RASA Core for dialogue flow management, chosen for its flexibility, natural language handling, and Arabic language support. Tayseer, designed to address queries about entrance and admission, among other frequent topics, incorporates multimodal interactions, feedback mechanisms, and sentiment analysis for comprehensive user engagement. It also supports diverse content types, such as images, links, and text, ensuring a versatile communication platform.

3.1. Menu-Based Interaction

The menu-based chatbot uses JavaScript and follows a hierarchical menu structure for user interaction.

A. Data Collection:

A comprehensive Arabic dataset encompassing intents, entities, and examples is currently being sought for menu-based and write-based modes. This initiative aims to equip the chatbot with the capability to accurately interpret inputs in Arabic. The desired dataset should cover a broad spectrum of college-related subjects, including but not limited to admissions, classes, scheduling, professors, amenities, and events. This dataset is derived from multiple authoritative sources, ensuring a rich and comprehensive compilation.

- Utilization of regulatory frameworks and procedural manuals to create a list of questions and answers for students by closely following the rules and steps found in manuals for technical and vocational training.
- Consultation with domain experts and faculty members to gather suggestions and recommendations.
- A pre-survey aimed at identifying the information priorities of prospective students was conducted among the current student population. The survey sought to understand the specific pieces of information that students consider imperative to know before and after college admission. A notable question within the survey was, “What are the crucial pieces of information you endeavor to obtain before and after being admitted to the college?” The survey garnered responses from 248 students, and a chart accompanying the results delineated the most common inquiries along with their respective percentages, as shown in Figure 2.

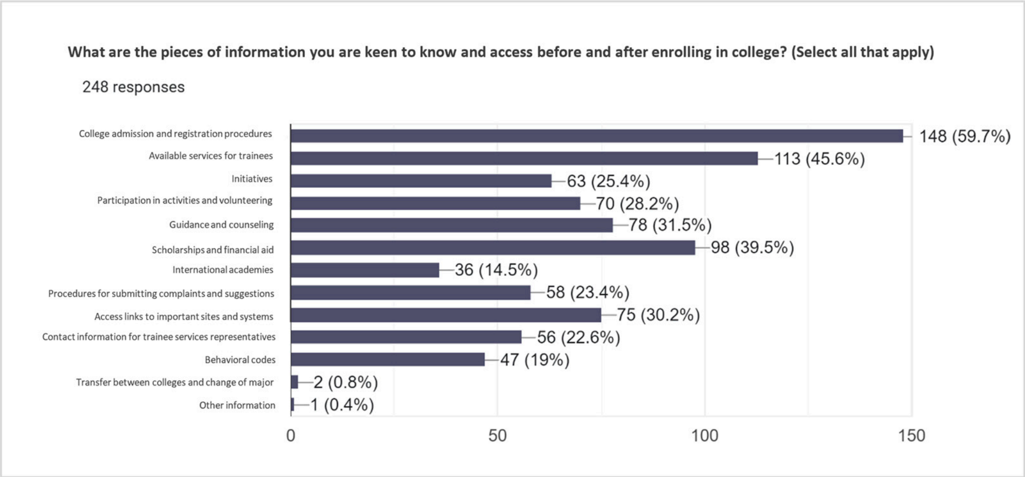


Figure 2. The most common inquiries.

B. Menu Structure:

The menu structure of the Taysser is represented as an object called "**menu**" in the code. Each key in the "**menu**" object represents a "**menu**" item, and the corresponding value is an object containing the "**text**" to display and the menu array representing the submenu items. For example, the top-level menu items include "القبول والتسجيل بالكلية" (College Admission and Registration), "خدمات المتدربات" (Trainee Services), "المبادرات" (Initiatives), and so on. Each menu item has its own submenu, allowing for hierarchical navigation.

C. Menu Rendering:

The "**menu**" options are rendered dynamically based on the "**menu**" object. The code iterates over the keys of the menu object and generates the corresponding UI elements to display the menu items. For instance, the code may generate buttons or list items for each menu item, with the appropriate text and event handlers to handle user interactions. The specific UI components and rendering process would depend on the chosen frontend framework or library.

D. User Interaction:

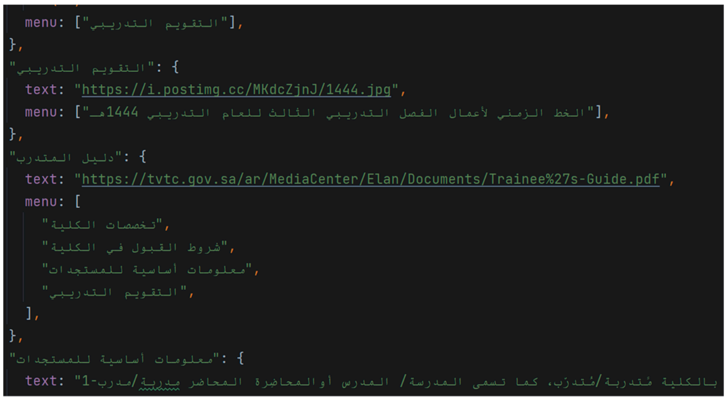
Tayseer handles user interactions by capturing user clicks or selections on the menu items. When a user selects a menu item, the chatbot navigates to the corresponding submenu or performs the associated action. For example, if the user selects "بالكلية القبول والتسجيل" (College Admission and Registration), the chatbot would display the submenu items related to college admission and registration, such as "تخصصات الكلية" (College Specializations), "شروط القبول في الكلية" (College Admission Requirements), and so on.

E. Menu Text and Links:

The text property of each menu item can contain plain text, URLs, or a combination of both. The chatbot renders the appropriate content based on the value of the text property. For example, let us consider the menu item "التقويم التدريبي" (Training Calendar). The text property of this menu item is set to "https://i.postimg.cc/MKdcZjnJ/1444.jpg" (accessed on 15 March 2023), representing an image URL. When the user selects this menu item, the chatbot will display the image associated with the training calendar.

On the other hand, the menu item "دليل المتدرب" (Trainee's Guide) has a text value of "https://tvtc.gov.sa/ar/MediaCenter/Elan/Documents/Trainee's-Guide.pdf" (accessed on 15 March 2023). This URL points to a document or a web page containing the trainee's guide. When the user selects this menu item, the chatbot provides a link or redirects the user to the specified URL, allowing them to access the trainee's guide.

In some cases, the text property contains plain text without any links. For instance, the menu item "معلومات أساسية للمستجدات" (Basic Information for New Students) has a text value that consists of several paragraphs of plain text. When this menu item is selected, the chatbot will display the text content directly to the user. Figure 3 illustrates an example of how the chatbot may render different types of menu text and links.



```
  menu: ["التقويم التدريبي"],
},
"التقويم التدريبي": {
  text: "https://i.postimg.cc/MKdcZjnJ/1444.jpg",
  menu: ["الخط الزمني لأعمال الفصل التدريبي الثالث للعام التدريبي 1444هـ"],
},
"دليل المتدرب": {
  text: "https://tvtc.gov.sa/ar/MediaCenter/Elan/Documents/Trainee%27s-Guide.pdf",
  menu: [
    "تخصصات الكلية",
    "شروط القبول في الكلية",
    "معلومات أساسية للمستجدات",
    "التقويم التدريبي",
  ],
},
"معلومات أساسية للمستجدات": {
  text: "1-بالكلية مُتدربة/مُتدرّب، كما تسمى المدرسة / المدرس أو المحاضرة المحاضر مدربة/مدرب
```

Figure 3. An example of how the chatbot renders different types of menu text and links.

In the example shown in Figure 3, the chatbot would handle each menu item differently based on the text property. For the "التقويم التدريبي" (training calendar) item, it would display the associated image. For the "دليل المتدرب" (trainee's guide) item, it would provide a link or redirect the user to the trainee's guide. And for the "معلومات أساسية للمستجدات" (basic information for new developments) item, it would display the plain text content directly. By leveraging the flexibility of the text property, Tayseer can accommodate various types of content, such as images, links, and plain text, providing a rich and informative user experience.

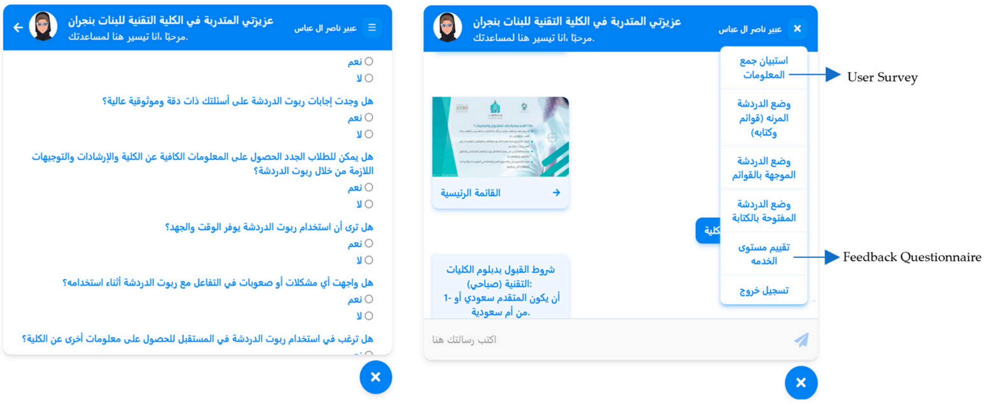
F. Error Handling and Fallback Mechanisms:

When the user selects an invalid or non-existent menu option, the chatbot provides appropriate error messages or fallback responses to guide the user back to valid menu options.

The chatbot's menu structure is represented as an object, and the menu options are dynamically rendered based on this structure. User interactions are handled by capturing user selections, navigating to the corresponding submenus, or performing associated actions. The menu-based chatbot offers a user-friendly and intuitive interface for users to interact with the system.

G. Evaluation and User Feedback:

To assess the effectiveness and usability of the Arabic AI chatbot, a comprehensive evaluation and user feedback process was integrated into Tayseer's front end using JavaScript, as shown in Figure 4. The evaluation consisted of a feedback questionnaire.



عزيزتي المتدربة في الكلية التقنية للبنات بنجران
مرحبًا، انا تيسير هنا لمساعدتك.
عبير ناصر ال عباس

○ نعم
○ لا

هل وجدت إجابات ربوت الدردشة على أسئلتك ذات دقة وموثوقية عالية؟
○ نعم
○ لا

هل يمكن للطلاب الجدد الحصول على المعلومات الكافية عن الكلية والإرشادات والتوجيهات اللازمة من خلال ربوت الدردشة؟
○ نعم
○ لا

هل ترى أن استخدام ربوت الدردشة يوفر الوقت والجهد؟
○ نعم
○ لا

هل واجهت أي مشكلات أو صعوبات في التفاعل مع ربوت الدردشة أثناء استخدامه؟
○ نعم
○ لا

هل ترغب في استخدام ربوت الدردشة في المستقبل للحصول على معلومات أخرى عن الكلية؟

عزيزتي المتدربة في الكلية التقنية للبنات بنجران
مرحبًا، انا تيسير هنا لمساعدتك.
عبير ناصر ال عباس
استبيان جمع المعلومات
User Survey
وضع الدردشة المرنه (قوائم وكتابه)
وضع الدردشة الموجهة بالقوائم
وضع الدردشة المفتوحة بالكتابة
تقييم مستوى الخدمه
Feedback Questionnaire
تسجيل خروج
القائمة الرئيسية
اكتب رسالتك هنا

شروط القبول بدبلوم الكليات التقنية (صباحي):
1- أن يكون المتقدم سعودي أو من أم سعودية.

Figure 4. User feedback process.

3.2. Write-Based (RASA-Based)

The choice of RASA for the conversational framework was motivated by its open-source nature, flexibility, and comprehensive support for Arabic. The architecture of RASA, comprising RASA NLU for natural language understanding and RASA Core for managing conversation flows, is particularly adept at handling the complexities of Arabic, including dialectal variations and script nuances. This framework facilitated the development of a chatbot capable of engaging in meaningful Arabic conversations and addressing the specific needs of technical and vocational college inquiries. The key components of the RASA framework are as follows:

- RASA NLU: processes and interprets user inputs in Arabic, accommodating linguistic complexities.
- RASA Core: guides conversation flow using machine learning to provide contextually relevant responses.
- Domain: defines the chatbot's scope, including intents, entities, and actions tailored to college-related queries.
- Rules and Stories: establish conversational guidelines and example paths to train chatbots for diverse interactions.
- Actions: both predefined text messages and custom actions allow for dynamic responses and functionalities such as API calls or database queries.

A. Data Collection

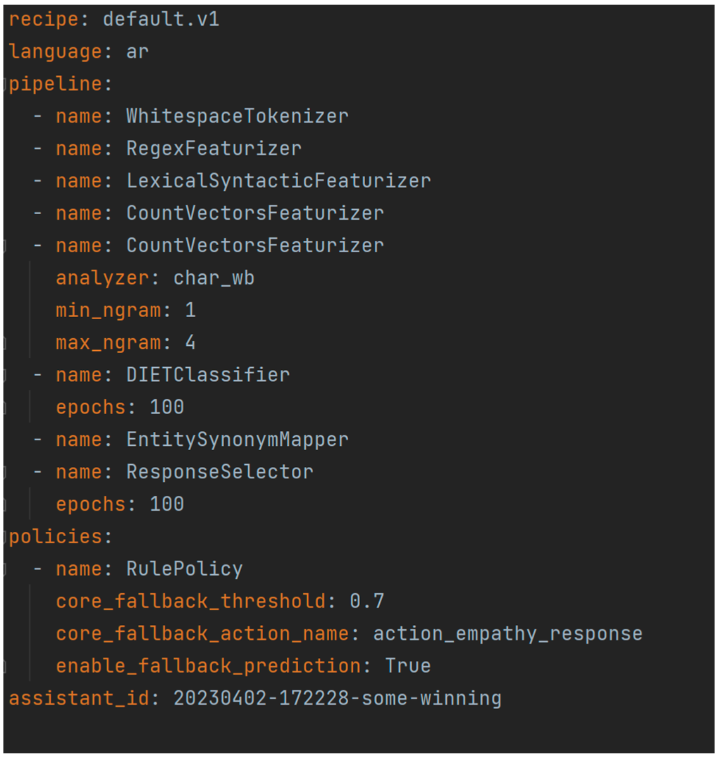
The data collection process for the write-based approach followed the same methodology described in the "Data Collection" sub-section for the menu-based approach. The same sources, including existing college documentation, frequently asked questions (FAQs), and expert knowledge from college staff and administrators, were utilized to gather a diverse and comprehensive dataset. The collected data were preprocessed and prepared for training the write-based chatbot model, ensuring consistency and compatibility with the write-based approach.

B. Data Preprocessing:

- The collected data underwent preprocessing to prepare it for training the chatbot model.
- Text normalization techniques were applied to standardize the text's format, including converting characters to lowercase and removing irrelevant characters or information.
- Tokenization was performed to split the text into individual words or tokens, enabling accurate processing and analysis.
- Arabic-specific challenges, such as discretization and morphological variations, were handled using specialized libraries and techniques to ensure proper processing of the Arabic language.

C. Choosing the RASA NLU Pipeline

The fundamental role of RASA NLU is to interpret the data provided by the user to the chatbot, which includes identifying necessary intents and entities. Within the RASA Open Source, a series of components sequentially process incoming messages. This ordered execution of components, known as the processing pipeline, is established in the config.yml file [75]. The supervised embedding pipeline allows for training in any global language, as it is designed to learn from the ground up. This type of pipeline is similar to that of the Tayseer pipeline, as shown in Figure 5.



```
recipe: default.v1
language: ar
pipeline:
  - name: WhitespaceTokenizer
  - name: RegexFeaturizer
  - name: LexicalSyntacticFeaturizer
  - name: CountVectorsFeaturizer
  - name: CountVectorsFeaturizer
    analyzer: char_wb
    min_ngram: 1
    max_ngram: 4
  - name: DIETClassifier
    epochs: 100
  - name: EntitySynonymMapper
  - name: ResponseSelector
    epochs: 100
policies:
  - name: RulePolicy
    core_fallback_threshold: 0.7
    core_fallback_action_name: action_empathy_response
    enable_fallback_prediction: True
assistant_id: 20230402-172228-some-winning
```

Figure 5. Tayseer supervised embedding pipeline.

The open-source RASA platform provides users with a powerful toolkit for building AI-driven agents that can chat, fine-tuned with a natural language processing pipeline. Figure 6 illustrates the agent training process executed in RASA.

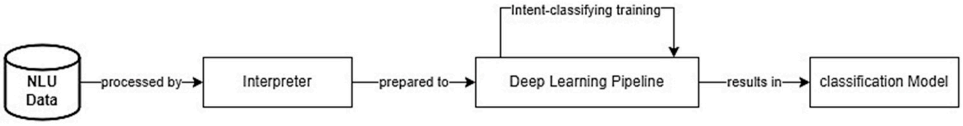


Figure 6. Illustration of the agent training process executed in RASA.

Within the scope of the Arabic language, RASA supports comprehensibility and conversational participation through a set of customizable components specified as follows. First, the preprocessing phase uses the **WhitespaceTokenizer** for simple text segmentation. This is complemented by the **RegexFeaturizer,** which extracts features based on the patterns. The **LexicalSyntacticFeaturizer** performs deep syntactic parsing to ease navigation through the complexities of Arabic syntax. It uses two **CountVectorsFeaturizer** components: the first to produce token-level analysis using the bag of words and the second to analyze the Arabic language's linguistic subtleties using character models ranging from 1- to 4-g. Intent and entity recognition are handled quite well by the **DIETClassifier,** which leverages transformer technology to produce iteratively updated context-sensitive predictions, as shown in Figure 7. Subsequent levels include the **EntitySynonymMapper**, which aligns terms of entities for overall consistency, and the **ResponseSelector**, which is adept at sourcing suitable replies to common queries.

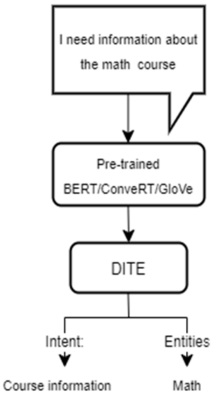

I need information about the math course
Pre-trained BERT/ConveRT/GloVe
DITE
Intent:
Entities
Course information
Math

Figure 7. Overview of the DIET classifier architecture.

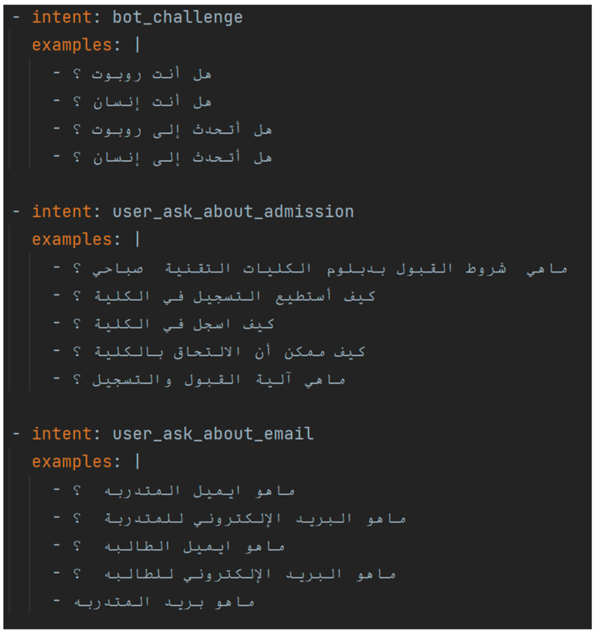
In instances where predictions may falter with low confidence, a **RulePolicy** steps in to maintain conversational coherence, and there is a unique assistant ID for every part of the pipeline to make it possible to trace any part. As indicated in Table 1, the RASA configuration is designed conscientiously, especially for Arabic conversational AI, with each unit serving a separate function. Designed with educational dialogues to be precisely tuned, the system is engineered to offer an unbeatable, knowledgeable virtual aid for technical and vocational students. In general, RASA's flexible pipeline combines state-of-the-art NLP methodologies that are sensitive to producing next-generation conversational AI abilities in Arabic.

D. Intent and Entity Recognition:

- The intents and entities relevant to the chatbot's domain were carefully defined based on the collected data and the chatbot's desired functionality.
- Intents represent the underlying purpose or goal of a user′s query, while entities refer to specific pieces of information within the query.
- Examples of intents defined for the Arabic AI chatbot include "user_ask_about _admission", "user_ask_about_email", and "user_ask_about_major_change".
- Each intent was associated with a set of example user expressions that capture different ways users might express their queries so the chatbot could respond with greater specificity, as shown in Figure 8.

Table 1. Summary of the responsibility of each component in the chosen configuration.

Component	Functionality	
Tokenizer	WhitespaceTokenizer	Splits text into tokens using spaces for clear whitespace languages.
Featurizer	RegexFeaturizer	Uses regex to identify patterns and shapes in text for comprehension.
	LexicalSyntacticFeaturizer	Captures lexical and syntactic text features, like POS tags.
	CountVectorsFeaturizer	Converts text to numerical data using a bag-of-words model.
Classifier	DIETClassifier	Predicts intents and extracts entities simultaneously.
Mapper	EntitySynonymMapper	Maps synonyms to entities to recognize varied expressions as the same.
Selector	ResponseSelector	Picks appropriate responses from predefined options and trains for 100 epochs.



```
- intent: bot_challenge
  examples: |
    - هل أنت روبوت ؟
    - هل أنت إنسان ؟
    - هل أتحدث إلى روبوت ؟
    - هل أتحدث إلى إنسان ؟

- intent: user_ask_about_admission
  examples: |
    - ماهي  شروط القبول بدبلوم الكليات التقنية  صباحي ؟
    - كيف أستطيع التسجيل في الكلية ؟
    - كيف اسجل في الكلية ؟
    - كيف ممكن أن الالتحاق بالكلية ؟
    - ماهي آلية القبول والتسجيل ؟

- intent: user_ask_about_email
  examples: |
    - ماهو ايميل المتدربه  ؟
    - ماهو البريد الإلكتروني للمتدربة  ؟
    - ماهو ايميل الطالبه  ؟
    - ماهو البريد الإلكتروني للطالبه  ؟
    - ماهو بريد المتدربه
```

Figure 8. A set of example user expressions.

- The intent is specified in the "nlu.yml" and "domain.yml" files. The "nlu.yml" file will encompass the intent data that will be utilized for subsequent training purposes. Table 2 displays the various contexts and quantities of intent within each. For this research, we created 75 Arabic intents. Figure 9 illustrates the distribution of Tayseer's intents by context.

Table 2. Description of intents.

Context	Description
Greeting	Respond to various forms of greetings from users.
College admission and registration	Handle queries related to the college admissions and registration process.
Trainee services	Address queries or requests related to services provided to trainees.
Initiatives	Responding to user inquiries about initiatives for trainee career guidance and college goal achievement.
Activity participation and volunteering	Responding to user inquiries or interests in extracurricular activities and volunteering opportunities.
Guidance and counseling	Address user queries related to advice, support, or information on personal, educational, or career matters.
Rewards and financial aid	Respond to queries about financial support.
Global academies	Inquiries on Adobe, Cisco, and Microsoft academies' programs and courses.
Complaints and suggestions	For handling user feedback and suggestions.
Access key site/system links	Assist users in finding and navigating to essential website or system links.
Faculty staff contact information	Assist users in obtaining faculty members' contact details.
Trainee conduct regulations	Information on trainee behavior rules.

- To manage diverse inquiries, a strategy is established in the "domain.yml" file under "utter", specifying conversational responses and their intents. The templates in this file address intent-specific queries. Tayseer encompasses more than 40 functions, with Table 3 providing samples from its diverse "utter" (user expressions or inputs).
- The training data was annotated with the corresponding intents and entities using the RASA NLU framework, which provides a structured format for training the chatbot's natural language understanding component.

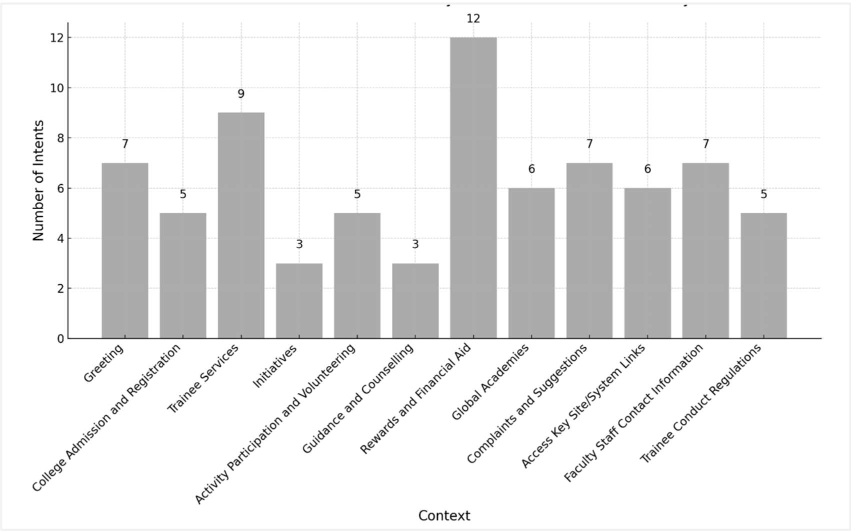


Figure 9. The distribution of chatbot intents by context.

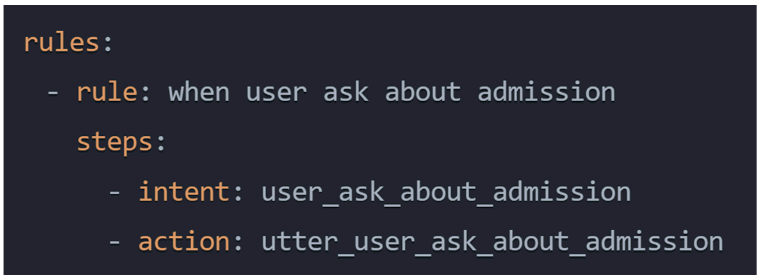
Table 3. List of examples of utter.

#	Ultter Name
1	utter_greet
2	utter_goodbye
3	utter_bot_challenge
4	utter_user_ask_about_admission
5	utter_user_ask_about_major_change
6	utter_user_ask_about_transfer_college
7	utter_user_ask_about_handicapped
8	utter_user_ask_about_admission_comparison
9	utter_user_wants_to_check_marks
10	utter_user_ask_about_re-enrollment
11	utter_user_ask_about_conditions_changing_specialization
12	utter_user_ask_about_studying_in_TVTC
13	utter_user_ask_about_rewarding_for_Student_withdrawn_deferred
14	utter_user_ask_about_instances_of_nonpayment_rewarding
15	utter_user_ask_about_training_record
16	utter_user_college_disciplines
17	utter_user_Training_plans_for_specializations
18	utter_user_Training_Guide
19	utter_user_Trainee's_Guide
20	utter_user_Training_calendar
21	utter_user_Complaints_Suggestions
22	utter_user_activities_volunteering
23	utter_user_uniform
24	utter_user_functional_coordination

E. Dialogue Management:

- The dialogue management component of the chatbot was implemented using the RASA framework.
- The conversation flow was defined by mapping intents to appropriate responses and actions using rules and stories.
- Rules specify the triggering intents and the corresponding actions or responses to be executed.
- Stories represent more extended conversation flows and allow for more complex interactions, enabling the chatbot to handle multi-turn dialogues.

- The dialogue management component enables the chatbot to maintain context, handle user input, and generate appropriate responses based on the identified intents and entities, as shown in Figure 10.



```
rules:
  - rule: when user ask about admission
    steps:
      - intent: user_ask_about_admission
      - action: utter_user_ask_about_admission
```

Figure 10. Intent-based response generation.

F. Response Generation:

- Generating accurate and coherent responses in Arabic was a crucial aspect of the chatbot's development.
- For each intent, predefined response templates were created to provide relevant information to users.
- These templates were designed to include specific details based on the identified intent, such as college admission requirements, email formats, and transfer procedures.
- The response templates were written in Arabic, considering proper grammar, gender agreement, and idiomatic expressions to ensure natural and fluent responses.
- The response generation process involved retrieving the appropriate template based on the user's intent and filling in any necessary dynamic information.

G. Integration and Deployment:

- Chatbot development: In this study, we explored the implementation of a web-based chatbot, "Tayseer", designed to enhance educational support through 24/7 accessibility, self-service convenience, interactive learning, and privacy. The chatbot was developed using the RASA open-source platform, enabling connections with popular messaging platforms and allowing for website integration.
- Front-end development: The front-end user interface of "Tayseer" was built using Next.js, a robust framework that ensures a highly responsive and user-friendly experience. Next.js was chosen for its ability to create dynamic and interactive user interfaces, which is crucial for engaging users and providing a seamless chatbot experience.
- Back-end development: The back end of "Tayseer" was developed using the Nest.js framework. Nest.js was selected for its robustness and ease of integration with the front end. It facilitates the management of critical functions such as user authentication and feedback processing, ensuring smooth communication between the chatbot system's front-end and back-end components.
- Deployment environment setup: To deploy "Tayseer", we utilized Microsoft Azure as the cloud platform. Azure provides a scalable and reliable production environment for hosting the chatbot. The deployment process involved setting up the necessary virtual machine resources; configuring Nginx as a high-performance proxy server; and deploying the RASA NLU, RASA Core, and custom action components.
- Front-end deployment: The web-based front-end interface, built with Next.js, was deployed separately on Vercel. Vercel is a cloud platform that specializes in hosting and deploying front-end applications. By deploying the front end on Vercel, we ensure fast and efficient delivery of the user interface to end users.
- Back-end deployment: The Nest.js back-end code was uploaded to a DigitalOcean Droplet. DigitalOcean is a cloud infrastructure provider that offers scalable and

secure hosting solutions. Hosting the back-end on a DigitalOcean Droplet ensures reliable and efficient processing of user requests and interactions with the chatbot.

- System integration: The final deployment step involved tying the front-end and back-end systems together. This integration ensures the user interface (UI) seamlessly communicates with the back-end components, enabling a unified and effective chatbot platform. Figure 11 depicts the Tayseer system architecture, illustrating the integration of various components.
- User authentication: To enhance the user experience, a user authentication process utilizing an SQL database was developed. User-specific information was stored and retrieved to enable tailored responses based on individual user profiles. Custom actions were developed for efficient data retrieval from the database, facilitating personalized interactions. Figure 12 illustrates the user registration process and the flow of user authentication.

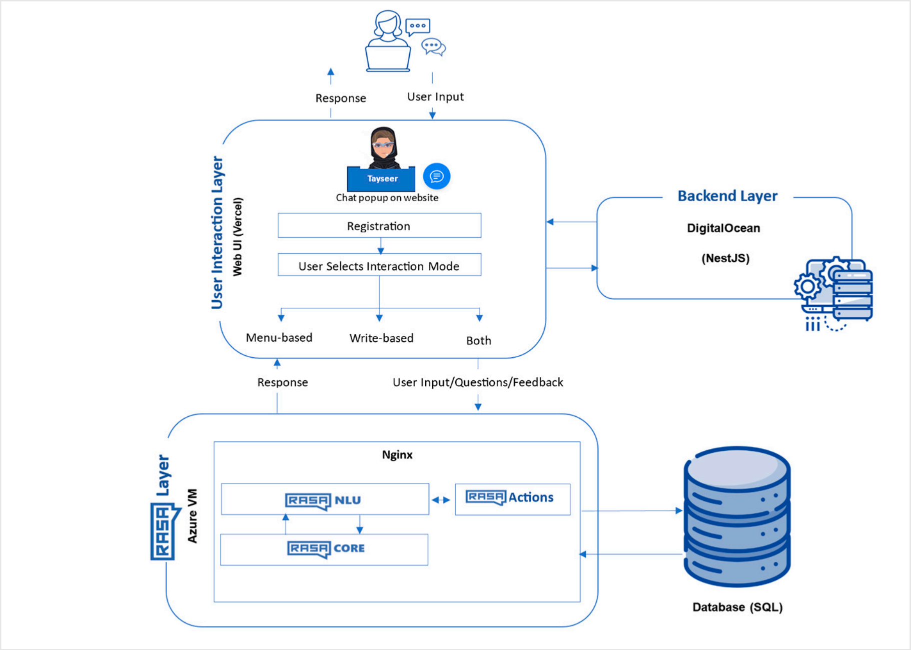
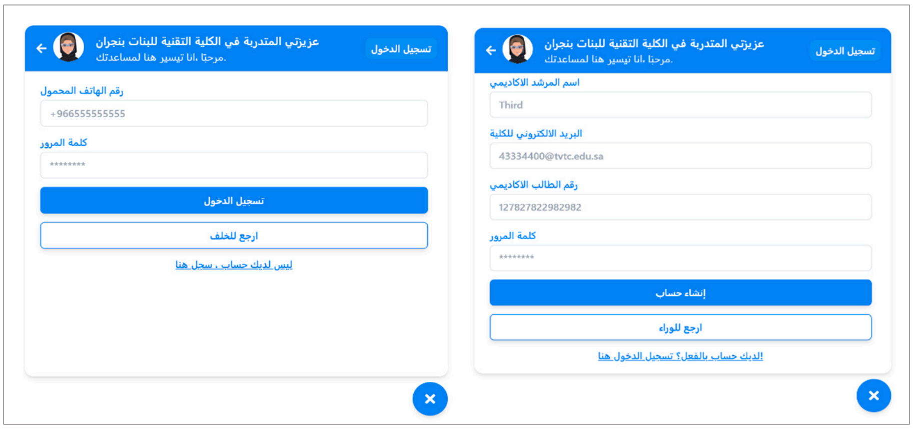


Figure 11. Tayseer system architecture.



عزيزتي المتدربة في الكلية التقنية للبنات بنجران
مرحبًا ،انا تيسير هنا لمساعدتك.

رقم الهاتف المحمول: +966555555555
كلمة المرور: ********

عزيزتي المتدربة في الكلية التقنية للبنات بنجران
مرحبًا ،انا تيسير هنا لمساعدتك.

اسم المرشد الاكاديمي: Third
البريد الالكتروني للكلية: 43334400@tvtc.edu.sa
رقم الطالب الاكاديمي: 127827822982982
كلمة المرور: ********

Figure 12. Registering a new user profile.

4. Experiment

4.1. Training and Validation

In this work, we experimented with a dataset with 75 intents, inquiring about main issues such as college admission and registration, trainee services, initiatives, guidance and counseling, rewards and financial aid, faculty staff contact information, and trainee conduct regulations. The dataset consisted of approximately 265 examples.

If you want to ask about the conditions for changing specialization at the Technical College for Girls in Najran, you can send the message "ماهي شروط تغيير التخصص" ("What are the conditions for changing specialization?"). RASA NLU will then extract all the necessary information and identify the user's intent as "user_ask_about_conditions_changing_specialization". This enables Tayseer to provide an accurate and relevant response to the user's query.

We created a global 80% train and 20% test split from the nlu.yml file to ensure robustness and reliability. This split was performed to evaluate the model's performance on unseen data and assess its generalization ability. Additionally, we applied k-fold cross-validation to the training data to obtain a more comprehensive assessment of the model's performance.

The training data were divided into five equal-sized groups for k-fold cross-validation. In each iteration, one-fold was used as the testing set, while the remaining folds were used for training. This process was repeated until each of the five folds had been used for testing. Although computationally expensive, this cross-validation strategy provides an adequate evaluation of the model's performance, especially when the dataset is not excessively large.

To handle the scenario where the conversation can start with any intent, we added each story to the stories.yml file with a single intent mentioned in the domain.yml file. This ensured that Tayseer could respond appropriately, regardless of the user's initial intent. The pipeline and policies for the chatbot were defined in the config.yml file.

4.2. User Testing

A user testing phase assessed the chatbot's usability and effectiveness in a real-world scenario. College students and staff members were invited to interact with the chatbot and provide feedback. The objectives of user testing were to evaluate the chatbot's ability to understand user queries, provide relevant and accurate information, and enhance user satisfaction.

Users were given specific tasks to perform, such as asking about admission requirements, requesting information about email formats, or inquiring about major change procedures. Feedback was collected through surveys and interviews, capturing users' experiences, opinions, and suggestions for improvement.

4.3. Sentiment Analysis

After conducting user testing, sentiment analysis was performed on the collected user feedback to gain insights into users' sentiments and opinions regarding their interaction with the chatbot. The sentiment analysis process involved the following steps:

A. Data collection: The user feedback data, including survey responses and interview transcripts, were collected and preprocessed for analysis. The relevant feedback questions and responses were extracted and organized in a structured format.

B. Data preprocessing: The feedback data underwent preprocessing to prepare for sentiment analysis. This involved cleaning the text, removing any irrelevant characters or information, and handling Arabic-specific challenges such as discretization and morphological variations.

C. Sentiment lexicon: A sentiment lexicon specific to Arabic was utilized to assign sentiment scores to words and phrases. The lexicon contained a list of words along with their associated sentiment scores, ranging from positive to negative.

D. Sentiment scoring: Each feedback response was tokenized, and the sentiment score for each word was calculated using the sentiment lexicon. The overall sentiment score for each response was determined by aggregating the sentiment scores of individual words.

E. Sentiment classification: Based on the sentiment scores, each feedback response was classified as positive, negative, or neutral. Responses with scores above a certain threshold were considered positive, those below a threshold were considered harmful, and those within a specific range were considered neutral.

F. Analysis and visualization: The sentiment analysis results were analyzed to identify patterns, trends, and insights regarding users' sentiments towards the chatbot. The distribution of positive, negative, and neutral sentiments was visualized using charts and graphs to provide a clear overview of user opinions.

By conducting sentiment analysis on the user feedback, valuable insights were gained into the users' perceptions and experiences with the chatbot. The sentiment analysis results complemented the user testing findings, providing a comprehensive understanding of user satisfaction and areas for improvement.

5. Evaluation and Discussion

We employed a comprehensive evaluation methodology to ensure Tayseer's efficacy and efficiency. This methodology was designed to assess the chatbot's performance across multiple dimensions, ensuring that it understands user inputs accurately, responds promptly, and can scale to handle many queries. The conversational agent was evaluated on several key performance metrics to assess its natural language understanding capabilities and scalability.

5.1. Accuracy

To calculate the accuracy of our chatbot, we evaluated the two main components of every chatbot using multiple evaluation metrics:

- NLU model evaluation: The chatbot's natural language understanding (NLU) component was rigorously tested using a held-out test set. This test set comprised user inputs that the model had not seen previously during training, simulating real-world interactions. The model's predictions on this test set were then compared with the actual intents and entities to determine their accuracy.
- Dialogue model evaluation: The dialogue management component was evaluated using test stories. These stories represent potential user–bot interactions, and the chatbot's responses in these stories were compared to the expected actions to determine their accuracy in managing dialogues.

To calculate the accuracy of the NLU and dialogue models, we used multiple evaluation metrics: confusion matrix, precision, recall, F1 score, and support. By leveraging these metrics, we were able to gain a holistic understanding of our chatbot's performance, pinpointing areas of strength and potential improvement. Figure 13 shows the results of the intent confusion matrix. This table shows the performance of the classification model. It compares actual and predicted classifications. Our chatbot shows how often each action or intent was correctly predicted and how often an incorrect action or intent was predicted.

The histogram in Figure 14 shows the distribution of confidence scores for correct and incorrect intent predictions made using the conversational AI model. The vertical axis displays the confidence probability, ranging from 0 to 1. The horizontal axis represents the number of predictions within each confidence interval. Conversely, a notable number of incorrect predictions (red bar) align with a confidence level just below the midpoint of 0.5, indicating a higher likelihood of error as the model's confidence diminishes. This distribution illustrates the model's ability to align prediction confidence with accuracy, a desirable characteristic indicating effective model calibration. Figure 15 shows the results of the intent error.

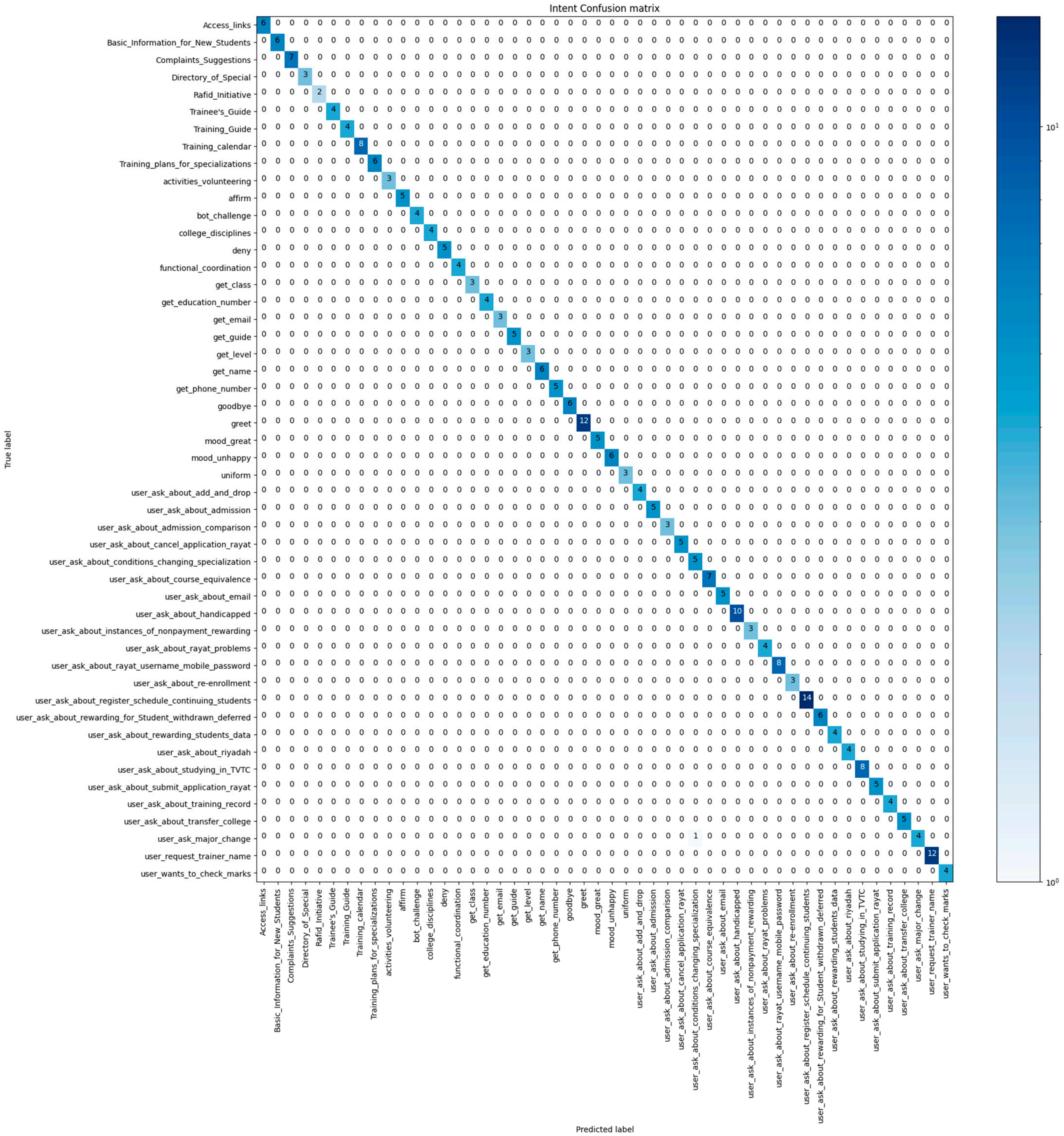


Figure 13. Intent confusion matrix.

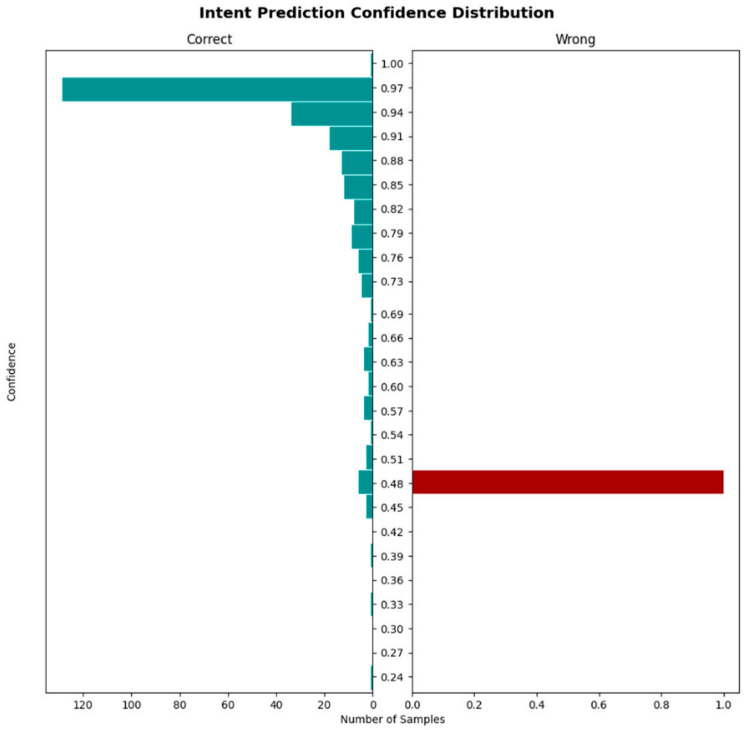
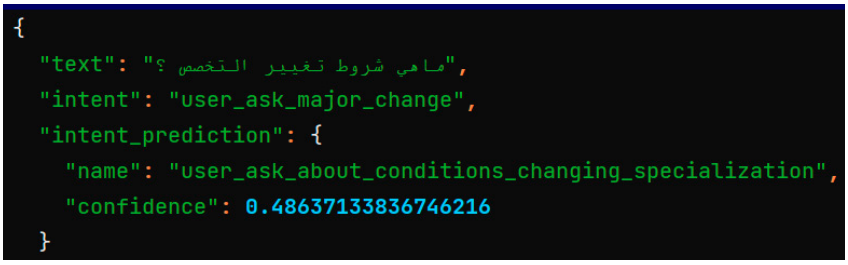


Figure 14. Intent prediction confidence distribution.



```
{
  "text": "ماهي شروط تغيير التخصص ؟",
  "intent": "user_ask_major_change",
  "intent_prediction": {
    "name": "user_ask_about_conditions_changing_specialization",
    "confidence": 0.48637133836746216
  }
```

Figure 15. Intent_Error.json.

The sentence "ماهي شروط تغيير التخصص" is meant to go on the user_ask_major_change intent but is predicted to go on the user_ask_about_conditions_changing_specialization intent. This will affect the confidence level of the intent prediction, resulting in responses that do not align with the inquirer's intended question. Table 4 shows the test result value of F1, recall, support, and precision score of the Tayseer intent.

The F1 score measures the test accuracy by combining precision and recall metrics, with maximum and minimum values of 1.0 and 0, respectively. Testing of the developed chatbot model revealed near-perfect F1 scores of 1.0 for most intent classes. However, two intents, user_ask_major_change and user_ask_about_conditions_changing_specialization, exhibited suboptimal F1 scores below 1.0. As illustrated in Table 4, the reduced F1 scores for these minority classes stem from sporadic errors in the model predictions. The confusion between these intents with similar semantic meanings contributed to diminished precision and recall. While most intents were classified with high accuracy, optimization of the model could potentially improve identification of the more easily confused intents, thereby increasing their F1 scores towards the optimal value of 1.0. Targeted augmentation of the training data for those specific intents may help enhance the precision and recall of the model for those classes. Overall, the F1 metric highlights strengths in broad classification accuracy while revealing specific areas for improvement in minority intents that could benefit from focused refinement.

Table 4. Test results.

Intent	Result			
	Precision	**Recall**	**F1 Score**	**Support**
Access_links	1.0	1.0	1.0	6
greet	1.0	1.0	1.0	12
user_ask_about_rewarding_for_Student_withdrawn_deferred	1.0	1.0	1.0	6
user_ask_about_re-enrollment	1.0	1.0	1.0	3
user_ask_about_admission	1.0	1.0	1.0	5
user_ask_about_rewarding_students_data	1.0	1.0	1.0	4
user_ask_major_change	1.0	0.8	0.888888888888889	5
user_ask_about_rayat_problems	1.0	1.0	1.0	4
mood_great	1.0	1.0	1.0	5
user_ask_about_admission_comparison	1.0	1.0	1.0	3
uniform	1.0	1.0	1.0	3
activities_volunteering	1.0	1.0	1.0	3
Complaints_Suggestions	1.0	1.0	1.0	7
bot_challenge	1.0	1.0	1.0	4
get_guide	1.0	1.0	1.0	5
user_ask_about_riyadah	1.0	1.0	1.0	4
mood_unhappy	1.0	1.0	1.0	6
user_ask_about_submit_application_rayat	1.0	1.0	1.0	5
get_phone_number	1.0	1.0	1.0	5
user_ask_about_rayat_username_mobile_password	1.0	1.0	1.0	8
get_name	1.0	1.0	1.0	6
deny	1.0	1.0	1.0	5
user_ask_about_email	1.0	1.0	1.0	5
user_ask_about_transfer_college	1.0	1.0	1.0	5
get_level”	1.0	1.0	1.0	3
functional_coordination	1.0	1.0	1.0	4
Training_plans_for_specializations	1.0	1.0	1.0	6
goodbye	1.0	1.0	1.0	6
college_disciplines	1.0	1.0	1.0	4
Rafid_Initiative	1.0	1.0	1.0	2
Training_Guide	1.0	1.0	1.0	4
user_ask_about_instances_of_nonpayment_rewarding	1.0	1.0	1.0	3
user_ask_about_cancel_application_rayat	1.0	1.0	1.0	5
get_education_number	1.0	1.0	1.0	4
user_ask_about_training_record	1.0	1.0	1.0	4
get_email	1.0	1.0	1.0	3
Trainee’s_Guide	1.0	1.0	1.0	4
get_class	1.0	1.0	1.0	3
user_wants_to_check_marks	1.0	1.0	1.0	4
Directory_of_Special	1.0	1.0	1.0	3
user_ask_about_conditions_changing_specialization	0.8333333333333334	1.0	0.9090909090909091	5
Training_calendar	1.0	1.0	1.0	8
affirm	1.0	1.0	1.0	5
user_ask_about_handicapped	1.0	1.0	1.0	10
user_request_trainer_name	1.0	1.0	1.0	12
Basic_Information_for_New_Students	1.0	1.0	1.0	6
user_ask_about_register_schedule_continuing_students	1.0	1.0	1.0	14
user_ask_about_course_equivalence	1.0	1.0	1.0	7
user_ask_about_add_and_drop	1.0	1.0	1.0	4
user_ask_about_studying_in_TVTC	1.0	1.0	1.0	8

5.2. Response Time

Chatbot responsiveness is crucial for user satisfaction. To evaluate this, we logged the timestamp when a user sent a message and the bot responded. The difference between these timestamps provided the response time, allowing us to assess the chatbot’s speed in generating and delivering responses. The response times for intents averaged under 1 s (967 ms), providing reasonably fast interactions, as shown in Figure 16. Any responses above 1 s, such as for user_ask_about_course_equivalence, present optimization opportunities.

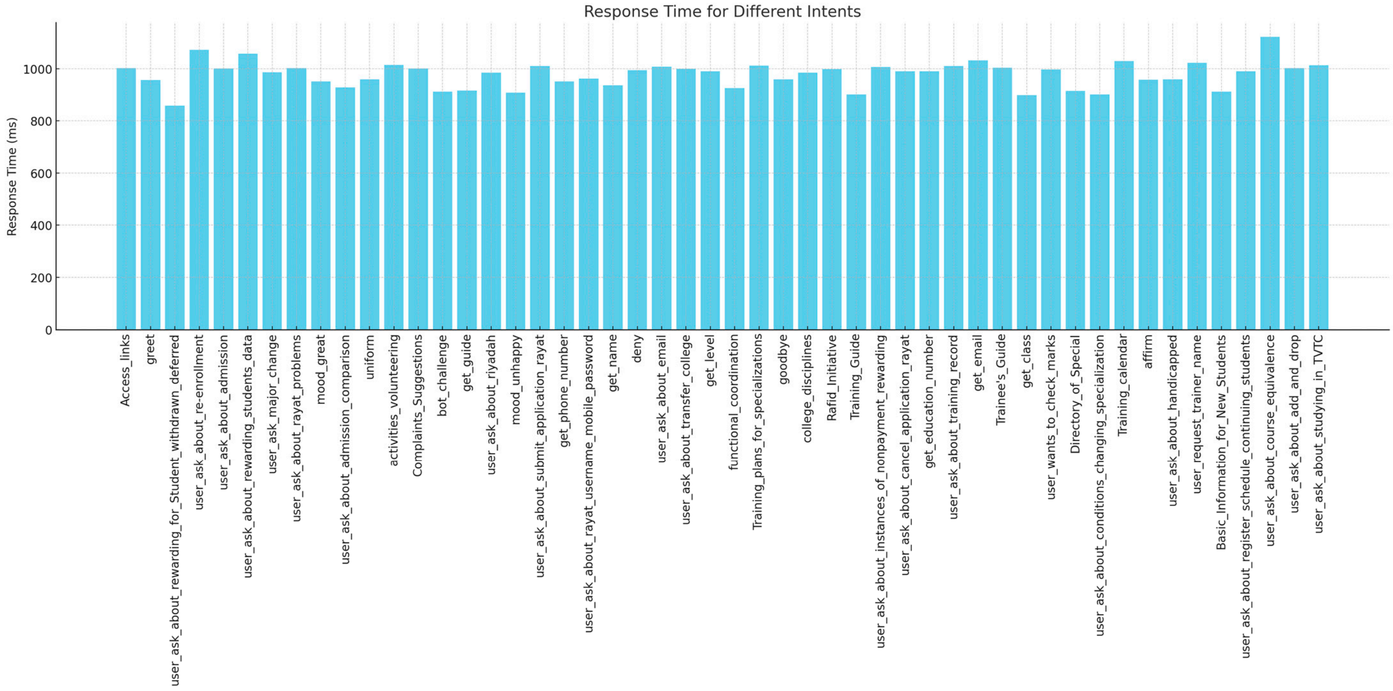


Figure 16. Response time for different intents.

5.3. Scalability

Scalability is a vital metric in chatbot solutions. It determines the chatbot's ability to handle many simultaneous user queries without performance degradation or increased response time. Scalability tests involved subjecting the chatbot to a high volume of user queries in a short timeframe and observing its performance. Testing up to six concurrent users showed minor increases in the average response times from 981 to 1023 ms, as shown in Figure 17. This linear scalability indicates the potential for maintaining performance with higher request volumes.

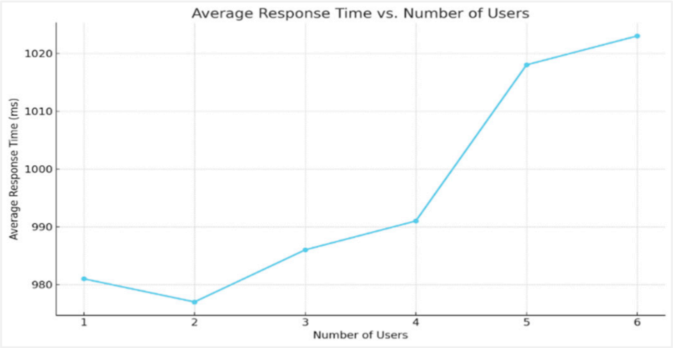


Figure 17. Average response time vs. number of users.

5.4. User Feedback and Sentiment Analysis

A. Collecting User Feedback

Our system integrates a feedback mechanism post-interaction to ascertain the chatbot's effectiveness and enhance user experience. At the conclusion of their chat sessions, users were invited to rate their experiences or offer comments and suggestions. This feedback was then cataloged within an SQL database for subsequent analysis and enhancement.

B. Sentiment Analysis Procedure

We employed sentiment analysis to extract insights into user satisfaction and discern general sentiments from the feedback. This involved utilizing a sentiment analysis model to classify feedback sentiment as positive, negative, or neutral, aiding in identifying areas for improvement and directing future chatbot enhancements.

C. Data Preprocessing Steps

Data preprocessing involved several critical steps, initiated by importing feedback data into a panda DataFrame for the initial analysis. Subsequently, entries marked as "None" in specific columns were excluded to ensure the integrity of the data. The process then focused on refining the dataset for sentiment analysis by filtering out columns, specifically isolating those related to the "rate", and textual responses ranging from "answer1" to "answer19".

D. Sentiment Analysis Model Development

Developing the sentiment analysis model involved creating a lexicon-based approach that utilized a compilation of positive and negative terms in both Arabic and English, with each term assigned a specific sentiment score. This model operated through a series of steps. Initially, feedback texts underwent tokenization, breaking down the content into individual terms for detailed analysis. Subsequently, each term received a sentiment score based on the lexicon, with the collective scores of the terms used to ascertain the overall sentiment of the feedback. Finally, aggregate scores were categorized into three sentiment classes.

- Positive (score > 0)
- Neutral (score = 0)
- Negative (score < 0)

E. Sentiment Analysis of User Feedback Results

The analysis of the sentiment results was structured to provide a comprehensive overview of the feedback sentiments, as shown in Figure 13. We began by calculating the frequency of positive, negative, and neutral sentiments for each item under review. Following this, an overall sentiment distribution analysis was carried out to understand the cumulative sentiment trends across all the received feedback. Specific to the nature of the questions, the analysis included distribution analyses for responses in the "rate" column, categorizing them into classifications such as "Good" and "Normal". The percentages of "Yes/No" responses were computed for the binary questions. In addition, for the open-ended questions, prevalent responses were identified to gauge common themes or opinions. The findings are summarized in Figures 18 and 19, showing the sentiment analysis results and overall sentiment distribution across the feedback dataset.

The evaluation of 210 users' feedback following their interaction with the chatbot prototype revealed favorable outcomes. Among the 1460 sentiment expressions identified in all the questions, 1206 were positive, 184 were negative, and 70 were neutral. These findings suggest that most participants had a favorable encounter when engaging with the chatbot.

The specific characteristic emphasized was the chatbot's ability to provide clear and easily comprehensible responses, which received a favorable sentiment score of 68.

- Prompt and efficient response times to inquiries were recorded (69 good attitudes);
- Seamless and effortless engagement in interactive chats with the chatbot resulted in a favorable sentiment score of 70.

Areas requiring enhancement encompassed the following: encountering specific challenges in engaging with the chatbot resulted in 48 instances of negative sentiment responses. Moreover, the sentiment on the extent of comprehensiveness in the FAQ subjects and alternatives varied, with only 36 instances of positive sentiment compared to 32 instances of negative sentiment. Overall, the sentiment and reaction towards the customized academic support chatbot concept and existing prototype were pretty favorable, with particular areas for enhancing the user experience.

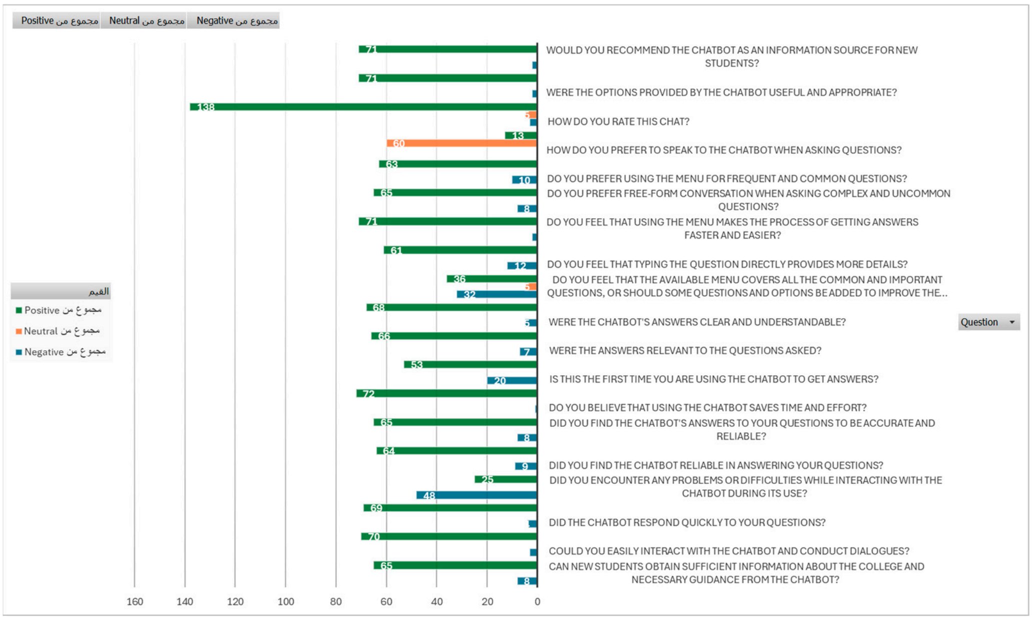
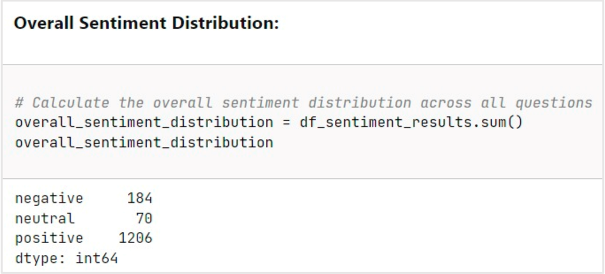


Figure 18. Sentiment analysis results.



Overall Sentiment Distribution:

```
# Calculate the overall sentiment distribution across all questions
overall_sentiment_distribution = df_sentiment_results.sum()
overall_sentiment_distribution
```

```
negative     184
neutral       70
positive    1206
dtype: int64
```

Figure 19. Overall sentiment distribution.

To summarize the evaluation metrics and results, conversational AI underwent an evaluation to assess its proficiency in comprehending natural language by employing a sense test.

- The confusion matrix for intent classification and the distribution of prediction confidence demonstrated robust overall performance, albeit with scope for enhancement in differentiating between particular comparable intentions.
- The F1 score for the majority of intentions was high, approaching 1.0, indicating a robust level of precision and recall. However, two semantically similar intentions exhibited lower F1 scores, suggesting occasional confusion errors. Additional training data augmentation can enhance the optimization of these less common intentions.
- The average response time for user interactions was 967 ms, which is considered reasonably fast. However, certain specific intents, such as course equivalency, had longer response times, indicating areas that could be optimized.
- Scalability testing involved up to six concurrent users and demonstrated satisfactory performance, with only slight increments in response times from 981 to 1023 ms. This suggests an ability to manage large quantities without a decrease in performance.
- Positive user sentiment expressions were predominant, accounting for 83% of interactions. The main advantages were the respondents' clear and comprehensible answers as well as seamless interaction.

- Areas for improvement included difficulties faced with the chatbot and limitations in the range of FAQ topics addressed.
- The positive user response underscores the need for more chatbot improvements to assist students with typical academic inquiries and procedures.

Through the application of these metrics, a comprehensive evaluation of the chatbot's performance was achieved, illuminating its proficiency and areas that necessitate enhancement. This rigorous analysis not only underscores the chatbot's current capabilities but also guides future refinements to augment its effectiveness in user interaction.

6. Conclusions

To summarize, this study has highlighted the vast potential of AI-driven conversational agents in efficiently managing academic student inquiries through natural language processing of the Arabic language. With DL techniques made by the RASA framework and implemented on the web, over 75 data intents in the dataset were handled with care by Tayseer and used in multiple ways to be processed into an effective pipeline and a strategy for data pre-processing.

A thorough assessment suggests that the prototype chatbot created offers precise predictions of user intentions for most scenarios, achieving F1 scores close to 1.0 and delivering rapid responses in less than a second. Scalability tests demonstrate the capacity to sustain performance even when faced with increased levels of requests. Nevertheless, there are still opportunities for improvement, specifically for minority intentions that occasionally have misunderstandings caused by semantic resemblance. We can increase precision and recall by focusing on improving the training data for those particular intentions. There is also the possibility of broadening the chatbot's knowledge base to encompass additional academic question categories that interest pupils and enhancing the dialog system to be more context-aware.

Significantly, user sentiment analysis indicated primarily favorable perceptions of utilizing the chatbot for acquiring university admission and registration aid. Students expressed positive feedback regarding the chatbot's clear and user-friendly responses. However, certain difficulties were faced while dealing with increasingly complex questions, which emphasizes the necessity for continuous improvements in a particular area.

Chatbots with AI are meaningful tools for highly individualized and accessible academic support, leaving no doubt about discovering new ways to achieve a more elevated level of experience through effective management of internal functions. Future work will focus on refining chatbots using user feedback to better adapt to changing educational needs. Open-ended input from students could provide specific suggestions that could be used to determine new question subjects. Moreover, the knowledge base should be expanded, and advanced language models should be integrated. By implementing these enhancements, chatbots have significant potential to function as an exceptionally valuable self-service tool, effectively reducing the time consumption for both students and college administrators. This study highlights the importance of user-centric approaches in creating engaging chatbot experiences that align with human preferences.

Author Contributions: The presented idea was planned by A.A., who also wrote the manuscript with the help of K.A., who supervised the final revisions and contributed to preparing the original draft. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: This research project has received approval from the Institutional Review Board (IRB) at the Technical and Vocational Training Corporation of Saudi Arabia, thereby guaranteeing that the study follows ethical guidelines and conforms to the necessary requirements for conducting research with participants.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shawar, B.A.; Atwell, E. Chatbots: Are they really useful? *LDV-Forum Z. Comput. Sprachtechnol.* **2007**, *22*, 29–49.
2. Al-Madi, N.A.; Maria, K.A.; Al-Madi, M.A.; Alia, M.A.; Maria, E.A. An Intelligent Arabic Chatbot System Proposed Framework. In Proceedings of the 2021 International Conference on Information Technology, ICIT 2021, Amman, Jordan, 14–15 July 2021; pp. 592–597. [CrossRef]
3. Russell, S.; Nilsson, P.; Nilsson, N.J. Artificial Intelligence: A Modern Approach. *Artif. Intell.* **1996**, *82*, 369–380.
4. Adamopoulou, E.; Moussiades, L. Chatbots: History, technology, and applications. *Mach. Learn. Appl.* **2020**, *2*, 100006. [CrossRef]
5. Skjuve, M.; Brandzaeg, P.B. Measuring user experience in chatbots: An approach to interpersonal communication competence. In *Internet Science*; Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer: Cham, Switzerland, 2019; pp. 113–120. [CrossRef]
6. Crisp, G.; Baker, V.L.; Griffin, K.A.; Lunsford, L.G.; Pifer, M.J. Mentoring Undergraduate Students. *ASHE High. Educ. Rep.* **2017**, *43*, 7–103. [CrossRef]
7. Wardhana, A.K.; Ferdiana, R.; Hidayah, I. Empathetic Chatbot Enhancement and Development: A Literature Review. In Proceedings of the AIMS 2021, International Conference on Artificial Intelligence and Mechatronics Systems, Jakarta, Indonesia, 28–30 April 2021; pp. 1–6. [CrossRef]
8. Yassin, S.M.; Khan, M.Z. SeerahBot: An Arabic Chatbot About Prophet's Biography SeerahBot: An Arabic Chatbot About Prophet's Biography. *Int. J. Innov. Res. Comput. Sci. Technol.* **2021**, *9*, 89–97. [CrossRef]
9. Moshuber, L. Grätzelbot: Using a Chatbot-Based Scavenger Hunt to Support Students' Transition to University. Master's Thesis, University of Vienna, Vienna, Austria, 2020.
10. Dibitonto, M.; Leszczynska, K.; Tazzi, F.; Medaglia, C.M. Chatbot in a campus environment: Design of LiSa, a virtual assistant to help students in their university life. In *Human-Computer Interaction: Interaction Technologies*; Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer International Publishing: Berlin/Heidelberg, Germany, 2018; pp. 103–116. [CrossRef]
11. Agarwal, A.; Maiya, S.; Aggarwal, S. Evaluating Empathetic Chatbots in Customer Service Settings. *arXiv* **2021**. [CrossRef]
12. Xu, A.; Liu, Z.; Guo, Y.; Sinha, V.; Akkiraju, R. A new chatbot for customer service on social media. In Proceedings of the Conference on Human Factors in Computing Systems, Denver, CO, USA, 6–11 May 2017; pp. 3506–3510. [CrossRef]
13. Fadhil, A.; AbuRa'Ed, A. Ollobot—Towards a text-based Arabic health conversational agent: Evaluation and results. In Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP, Varna, Bulgaria, 2–8 September 2019; pp. 295–303. [CrossRef]
14. Høiland, C.G.; Følstad, A.; Karahasanovic, A. Hi, Can I Help? Exploring How to Design a Mental Health Chatbot for Youths. *Hum. Technol.* **2020**, *16*, 139–169. [CrossRef]
15. Vaidyam, A.N.; Wisniewski, H.; Halamka, J.D.; Kashavan, M.S.; Torous, J.B.; Torous, J. Chatbots and Conversational Agents in Mental Health: A Review of the Psychiatric Landscape. *Can. J. Psychiatry* **2019**, *64*, 456–464. [CrossRef]
16. Von Wolff, R.M.; Hobert, S.; Schumann, M. How May I Help You?—State of the Art and Open Research Questions for Chatbots at the Digital Workplace. In Proceedings of the 52nd Hawaii International Conference on System Sciences, Maui, HI, USA, 8–11 January 2019; Volume 6, pp. 95–104.
17. Agarwal, R.; Wadhwa, M. Review of State-of-the-Art Design Techniques for Chatbots. *SN Comput. Sci.* **2020**, *1*, 246. [CrossRef]
18. Kuhail, M.A.; Alturki, N.; Alramlawi, S.; Alhejori, K. Interacting with Educational Chatbots: A Systematic Review. *Educ. Inf. Technol.* **2022**, *28*, 973–1018. [CrossRef]
19. Pérez-Marín, D. A Review of the Practical Applications of Pedagogic Conversational Agents to Be Used in School and University Classrooms. *Digital* **2021**, *1*, 18–33. [CrossRef]
20. Briel, A. Toward an Eclectic and Malleable Multiagent Educational Assistant. *Comput. Appl. Eng. Educ.* **2021**, *30*, 163–173. [CrossRef]
21. Gupta, M.; Verma, S.K.; Jain, P. Detailed Study of Deep Learning Models for Natural Language Processing. In Proceedings of the IEEE 2020 2nd International Conference on Advances in Computing, Communication Control and Networking, ICACCCN, Greater Noida, India, 18–19 December 2020; pp. 249–253. [CrossRef]
22. Luo, B.; Lau, R.Y.K.; Li, C.; Si, Y. A critical review of state-of-the-art chatbot designs and applications. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2022**, *12*, e1434. [CrossRef]
23. Palasundram, K.; Sharef, N.M.; Nasharuddin, N.A.; Kasmiran, K.A.; Azman, A. Sequence to sequence model performance for education chatbot. *Int. J. Emerg. Technol. Learn.* **2019**, *14*, 56–68. [CrossRef]
24. Sojasingarayar, A. Seq2Seq AI Chatbot with Attention Mechanism. *arXiv* **2020**, arXiv:2006.02767.
25. Alkhurayyif, Y.; Sait, A.R.W. A comprehensive survey of techniques for developing an Arabic question answering system. *PeerJ Comput. Sci.* **2023**, *9*, e1413. [CrossRef] [PubMed]

26. Alruily, M. ArRASA: Channel Optimization for Deep Learning-Based Arabic NLU Chatbot Framework. *Electronics* **2022**, *11*, 3745. [CrossRef]
27. Ahmed, A.; Ali, N.; Alzubaidi, M.; Zaghouani, W.; Abd-alrazaq, A.; Househ, M. Arabic chatbot technologies: A scoping review. *Comput. Methods Programs Biomed. Update* **2022**, *2*, 100057. [CrossRef]
28. Vo, V.M.N.; Ngo, V.S. An Integrated DIET-BO Model for Intent Classification and Entity Extraction. *Int. J. Intell. Syst. Appl. Eng.* **2023**, *11*, 666–673.
29. Arevalillo-Herraez, M.; Arnau-Gonzalez, P.; Ramzan, N. On Adapting the DIET Architecture and the Rasa Conversational Toolkit for the Sentiment Analysis Task. *IEEE Access* **2022**, *10*, 107477–107487. [CrossRef]
30. Fuad, A.; Al-Yahya, M. Recent Developments in Arabic Conversational AI: A Literature Review. *IEEE Access* **2022**, *10*, 23842–23859. [CrossRef]
31. Weizenbaum, J. ELIZA—A Computer Program for the Study of Natural Language Communication Between Man and Machine. *Commun. ACM* **1983**, *26*, 23–28. [CrossRef]
32. Colby, K.M. Modeling a paranoid mind. *Behav. Brain Sci.* **1981**, *4*, 515–534. [CrossRef]
33. Rashkin, H.; Smith, E.M.; Li, M.; Boureau, Y.L. Towards empathetic open-domain conversation models: A new benchmark and dataset. In Proceedings of the ACL 2019—57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 5370–5381. [CrossRef]
34. Hussain, S.; Sianaki, O.A.; Ababneh, N. *A Survey on Conversational Agents/Chatbots Classification and Design Techniques*; Springer International Publishing: Berlin/Heidelberg, Germany, 2019; Volume 927. [CrossRef]
35. Ritter, A.; Cherry, C.; Dolan, W.B. Data-driven response generation in social media. In Proceedings of the EMNLP 2011—Conference on Empirical Methods in Natural Language Processing, Edinburgh, UK, 27–31 July 2011; pp. 583–593.
36. Unsupervised Modeling of Twitter Conversations—ACL Anthology. Available online: https://aclanthology.org/N10-1020/ (accessed on 21 February 2022).
37. Huang, M.; Zhu, X.; Gao, J. Challenges in Building Intelligent Open-domain Dialog Systems. *ACM Trans. Inf. Syst.* **2020**, *38*, 21. [CrossRef]
38. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. *Adv. Neural Inf. Process. Syst.* **2014**, *4*, 3104–3112.
39. Smith, J. Empathetic Dialogue Generation Using Generation-Based Models. Diploma Thesis, Dept. Signals, Control, and Robotics, School of Electrical and Computer Eng., National Technical Univ. of Athens, Athens, Greece, 2020.
40. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.; Kaiser, L.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008.
41. Farha, I.A.; Magdy, W. Benchmarking Transformer-based Language Models for Arabic Sentiment and Sarcasm Detection. In Proceedings of the Arabic Natural Language Processing Workshop, Virtual, 19 April 2021; pp. 21–31.
42. Bubeck, S.; Chandrasekaran, V.; Eldan, R.; Gehrke, K.; Horvitz, E.; Kamar, E.; Lee, P.; Lee, Y.T.; Lundberg, S.; Nori, H.; et al. Sparks of Artificial General Intelligence: Early experiments with GPT-4. *arXiv* **2023**, arXiv:2303.12712.
43. Devlin, K.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
44. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **2020**, *21*, 1–67.
45. Escall, F.; Mello, D.A.A.; Arantes, D.S.; Rothenberg, C.E. Chat-IBN-RASA: Building an Intent Translator for Packet-Optical Networks based on RASA. In Proceedings of the IEEE 9th International Conference on Network Softwarization (NetSoft), Madrid, Spain, 19–23 June 2023; pp. 534–538. [CrossRef]
46. Bocklisch, T.; Faulkner, J.; Pawlowski, N.; Nichol, A. Rasa: Open Source Language Understanding and Dialogue Management. *arxiv* **2017**. [CrossRef]
47. Mishra, D.S.; Agarwal, A.; Swathi, B.; Akshay, K.C. Natural Language Query Formalization to SPARQL for Querying Knowledge Bases Using Rasa. *Prog. Artif. Intell.* **2022**, *11*, 193–206. [CrossRef]
48. Tsarfaty, R. Integrated Morphological and Syntactic Disambiguation for Modern Hebrew. In Proceedings of the COLING/ACL 2006 Student Research Workshop, Sydney, Australia, 20 July 2006. [CrossRef]
49. Almasoud, A.M.; Al-Khalifa, H.S. SemSignWriting: A Proposed Semantic System for Arabic Text-to-Sign Writing Translation. *J. Softw. Eng. Appl.* **2012**, *5*, 604–612. [CrossRef]
50. AL-Hagbani, E.S.; Khan, M.B. Support of Existing Chatbot Development Framework for Arabic Language: A Brief Survey. In *Advances in Intelligent Systems and Computing*; Springer International Publishing: Berlin/Heidelberg, Germany, 2018; pp. 26–35.
51. Alqahtani, Q.; Alrwais, O. Building a Machine Learning Powered Chatbot for KSU Blackboard Users. *Int. J. Adv. Comput. Sci. Appl.* **2023**, *14*, 781–789. [CrossRef]
52. Abdulkader, Z.; Al-Irhayim, Y. A Review of Arabic Intelligent Chatbots: Developments and Challenges. *Al-Rafidain Eng. J.* **2022**, *27*, 178–189. [CrossRef]
53. Aloqayli, A.; Abdelhafez, H. Intelligent Chatbot for Admission in Higher Education. *Int. J. Inf. Educ. Technol.* **2023**, *13*, 1348–1357. [CrossRef]
54. Jiao, A. An Intelligent Chatbot System Based on Entity Extraction Using RASA NLU and Neural Network. *J. Phys. Conf. Ser.* **2020**, *1487*, 012014. [CrossRef]

55. Alhumoud, S.; Al Wazrah, A.; Aldamegh, W. Arabic Chatbots: A Survey. *Int. J. Adv. Comput. Sci. Appl.* **2018**, *9*. [CrossRef]
56. Mozannar, H.; El Hajal, K.; Maamary, E.; Hajj, H. Neural Arabic question answering. In Proceedings of the Fourth Arabic Natural Language Processing Workshop, Florence, Italy, 1 August 2019; pp. 108–118. [CrossRef]
57. Boussakssou, M.; Ezzikouri, H.; Erritali, M. Chatbot in Arabic language using seq to seq model. *Multimed. Tools Appl.* **2022**, *81*, 2859–2871. [CrossRef]
58. Biltawi, M.; Awajan, A.; Tedmori, S. Towards Building an Open-Domain Corpus for Arabic Reading Comprehension Towards Building an Open-Domain Corpus for Arabic Reading Comprehension. In Proceedings of the 35th IBIMA Conference, Seville, Spain, 1–2 April 2020.
59. Antoun, W.; Baly, F.; Hajj, H. AraBERT: Transformer-based Model for Arabic Language Understanding. *arXiv* **2021**, arXiv:2003.00104.
60. Eljundi, O.; Antoun, W.; El, E. Hulmona: The Universal Language Model in Arabic. In Proceedings of the Fourth Arabic Natural Language Processing Workshop, Florence, Italy, 1 August 2019; Eljundi, O., Antoun, W., El, E., Hajj, H., El-Hajj, W., Shaban, K., Eds.; Association for Computational Linguistics: Kerrville, TX, USA, 2019; pp. 68–77.
61. Kushwaha, A.K.; Kar, A.K. MarkBot—A Language Model-Driven Chatbot for Interactive Marketing in Post-Modern World. *Inf. Syst. Front.* **2021**, 1–18. [CrossRef]
62. Nagoudi, E.M.B.; Elmadany, A.; Abdul-Mageed, M. AraT5: Text-to-Text Transformers for Arabic Language Understanding and Generation. *arXiv* **2021**, arXiv:2109.12068.
63. Antoun, W.; Baly, F.; Hajj, H. AraGPT2: Pre-Trained Transformer for Arabic Language Generation. In Proceedings of the Sixth Arabic Natural Language Processing Workshop, Kiev, Ukraine, 19 April 2021; pp. 196–207.
64. Abdul-Mageed, M.; Elmadany, A.; Nagoudi, E.M.B. ARBERT&MARBERT: Deep bidirectional transformers for Arabic. *arXiv* **2020**, arXiv:2101.01785.
65. Brandtzaeg, P.; Følstad, A. Chatbots: Changing user needs and motivations. *Interactions* **2018**, *25*, 38–43. [CrossRef]
66. Pérez, J.Q.; Daradoumis, T.; Puig, J.M.M. Rediscovering the use of chatbots in education: A systematic literature review. *Comput. Appl. Eng. Educ.* **2020**, *28*, 1549–1565. [CrossRef]
67. El Hefny, W.; Mansy, Y.; Abdallah, M.; Abdennadher, S. Jooka: A Bilingual Chatbot for University Admission. In *Advances in Intelligent Systems and Computing*; Springer: Cham, Switzerland, 2021; pp. 671–681. [CrossRef]
68. Al-Ghadhban, D.; Al-Twairesh, N. Nabiha: An Arabic Dialect Chatbot. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*. [CrossRef]
69. Aljameel, S.; O'Shea, J.; Crockett, K.; Latham, A.; Kaleem, M. LANA-I: An Arabic Conversational Intelligent Tutoring System for Children with ASD. In *Advances in Intelligent Systems and Computing*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 498–516. [CrossRef]
70. Almurtadha, Y. LABEEB: Intelligent Conversational Agent Approach to Enhance Course Teaching and Allied Learning Outcomes Attainment. *J. Appl. Comput. Sci. Math.* **2019**, *13*, 9. [CrossRef]
71. Alazzam, B.A.; Alkhatib, M.; Shaalan, K. Arabic Educational Neural Network Chatbot. *Inf. Sci. Lett.* **2023**, *12*, 2579–2589.
72. Tuning Your NLU Model. Available online: https://rasa.com/docs/rasa/tuning-your-model/ (accessed on 4 February 2024).
73. Al-madi, N.A.; Maria, K.A.A.; Al-madi, M.A.; Maria, E.A.A. An Intelligent Arabic Chatbot System to Reduce Admissions Department Burdens. *Authorea* **2023**. [CrossRef]
74. Castillo, D.; Canhoto, A.I.; Said, E. The Dark Side of AI-powered Service Interactions: Exploring the Process of Co-Destruction from the Customer Perspective. *Serv. Ind. J.* **2020**, *41*, 900–925. [CrossRef]
75. Nguyen, T.T.; Le, A.D.; Hoang, H.T.; Nguyen, T. NEU-chatbot: Chatbot for admission of National Economics University. *Comput. Educ. Artif. Intell.* **2021**, *2*, 100036. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.