EasyChair Preprint

№ 7411

Rahhal: a Tourist Arabic Chatbot

Sarah Alhumoud, Dana Aldukhail, Amjad Alshalhoub,
Reema Alabdullatif, Amna Diab, Dalal Alqahtany,
Maram Alalyani and Fai Bin-Aqeel

February 1, 2022

Rahhal: A Tourist Arabic Chatbot

Sarah AlHumou[1], Dana AlDukhai, Amjad AlShalhoub, Reema AlAbdullatif, Amna Diab, Dalal AlQahtany, Maram AlAlyani, Fai Bin-Aqeel
College of Computer and Information Science
Imam Mohammad Ibn Saud Islamic University (IMSIU)
Riyadh, Saudi Arabia

{ dsaldkail65, ayalshalhoub, raualabdullatif, aqadiab, dftalqahtany, mmbalalyani, fiaaqeel }@sm.imamu.edu.sa
[1] Corresponding Auther: sohumoud@imamu.edu.sa

***Abstract*—A chatbot is an artificial intelligence-based program that is able to interact with humans in a messaging style. It answers queries and provides useful information that could otherwise be offered with costly human labor. Chatbots are important to enhance user and customer experience in multiple domains like education, medicine, entertainment, and commerce. Although the research and commercialization of English chatbots is evolving relatively rapidly, Arabic chatbots are comparably scarce. Moreover, the tourism sector in Saudi Arabia, the second largest in the Middle East, is growing rapidly, especially after 2019. Hence, the contribution of this paper as it presents an Arabic chatbot that aids leisure tourists in Saudi Arabia with two implementations, the fitst is implemented using Java and the second was developed using IBM Watson. Both chatbots are deployed in Telegram. The tesing of the chatbots proved its high usability and utility in aiding turists.**

Keywords—Artificial intelligence; Arabic chatbot; conversational agent; ArabChat; human-machine interaction; utterance.

I. INTRODUCTION

Chatbots are special agents that communicate with human users via natural language processing (NLP), which is the branch of artificial intelligence that is concerned with enabling computers to understand human languages [1]. A chatbot or a chat-agent interacts with humans via natural language and emulates human conversation. This area has attracted increased interest from both research and industry fields in the past few years [2]. The first chatbot was developed at the Massachusetts Institute of Technology (MIT), where Weizenbaum implemented the ELIZA chatbot to emulate a psychotherapist in 1966 [3]. Chatbots range from unpretentious systems that extract answers from datasets when they match specific keywords to more advanced ones that utilize NLP techniques. A chatbot could be programmed to serve almost any language. Although research on English chatbots is diffused widely, research on Arabic chatbots is scarce due to language and resource challenges. One of these challenges is that Arabic has three forms: classical Arabic, modern standard Arabic (MSA), and dialectical Arabic. Classical Arabic is the form of language used in The Qur'an, the holy book of Islam. MSA is similar to classical Arabic with less sophistication and more generic words; it is used in formal written and spoken media such as the news, education, and literature. Dialectical or colloquial Arabic is used mostly in daily life and has regional variations. Although dialectical Arabic is used mainly as a spoken language, it is currently used in written social communication on social media and short messages and includes more than 30 dialects [4].

This paper contribution comes into two folds. First, it presents a comprehensive overview of Arabic chatbots by completing the survey conducted by AlHumoud et al. [5], that covered studies until 2017. Hence, this paper covers papers starting from 2018. Second, it presents a contribution in the field of Arabic chatbots by providing "Rahhal"—a tourism chatbot that helps leisure tourists to find activities and interesting locations in 81 main cities in Saudi Arabia. "Rahhal" in Arabic means traveler. For performance comparison, the chatbot was implemented in two ways: using Java programming language and IBM Watson.

The rest of the paper is structured as follows: Section 2 discusses Arabic chatbot research. Section 3 outlines the chatbot methodology and architecture. Section 4 describes the evaluation criteria. Section 5 presents the results and discussion, and Section 6 provides the conclusion.

II. ARABIC CHATBOT RELATED WORK

This section presents a survey of Arabic chatbots covering 14 different Arabic chatbot studies written after 2018 [5].

On reviewing the literature, we found that Arabic chatbots are built using two different modeling techniques: pattern matching (PM), and artificial intelligence markup learning (AIML). PM compares two patterns to determine whether they match (i.e., that they are the same) or do not match (i.e., that they differ) [6]. Where AIML is a well-known XML derived language to build chatter bot knowledge bases, in the context of case-based reasoning and textual PM algorithms [7].

To harness the power and benefits of AI in building and creating chatbots machine learning (ML) and deep learning (DL) models were recently utilized for several tasks in chatbots design. For example, [8] used ML to identify dialogue act. Where [9] and [10] used ML to find question similarity and maximize similarity between inputs and intents. Other studies, such as [11] used DL for generating responses. Also, [12] used DL to enable the chatbot to memorize messages in the training data file and to predict the next action. Lastly, [13] utilized DL for Intent classification and entity extraction. in summary, 20% of the studies under consideration used ML [8], [9], [10] and 20% used DL [11], [12], [13].

In the following, we present a discussion of the literature based on the different implementation techniques, dataset models, domains, and conversation interactions as shown in Table 2.

Where domains in chatbots are of two types: open and closed. Open-domain chatbots refer to chatbots that deal with multiple kinds of data from different fields. Where closed-domain refers to chatbots that deal with a specific domain or accept only a restricted kinds of data [14].

A. *Open-domain*

This section presents chatbots studies whose domain is open. The number of studies on open-domain chatbots are five.

The chatbot implemented by Alshareef and Siddiqui [11] was built to process the Arabic Gulf dialect utilizing a DL architecture known as neural LSTM-based Sequence-to-Sequence (Seq2Seq). The dataset was collected from Twitter API in a total of 5.2k pairs of post-replies. The chatbot domain was open. The modeling of the chatbot in Arabic was done using PM and information retrieval. The LSTM model was designed to estimate the probability of generating responses. Because of the presence of extensive dependencies in Arabic dialect phrases, the authors decided to use Transformer-based modeling. The Transformer outperformed Seq2Seq in terms of running time and parameter improvement.

In evaluating Gulf Arabic chatbot they used Bilingual Evaluation Understudy (BLEU) score. They tested the chatbot with 21 evaluators who spoke and knew the Arabic Gulf dialect. The authors applied descriptive statistics to analyze the results from all evaluators. Moreover, the authors calculated the overall mean score for all responses.

"SeerahBot" developed by Yassin and Khan [9] is an Arabic textual chatbot on the prophet Mohammed's biography. The domain is closed, and the sources of the chatbot dataset are retrieval-based. The dataset contains 200 questions and answers on the Prophet's biography from one of his biography books [15]. They built a chatbot from scratch using Python and the NLTK library utilizing ML techniques. the ML performance results are not included in the study. SeerahBot was deployed on Telegram. It was tested, and the authors claimed that the results are acceptable, although the sample size was 14, and this could be questionable.

Al-Ghadhban and Al-Twairesh [16] proposed "Nabiha", a chatbot that can support conversation with Information Technology (IT) students at King Saud University using the Saudi Arabic dialect. The dataset was collected from King Saud University IT students' accounts in Askme.com, and they collected a total of 248 inputs/outputs using Twitter and the Pandorabots platform. Nabiha is the first chatbot that uses the Saudi dialect. It is a closed domain, and the dataset model is retrieval based. Nabiha was tested by the students of the IT department, with a sample of 13 student. Authors stated that the results were satisfactory, considering the difficulty of the Arabic language in general and the Saudi dialect in particular.

Mabrouk et al. [10] implemented a multilingual African chatbot. It is a chatbot web interface for crisis communication that answers questions using English, French, Arabic, Tunisian, Igbo, Yoruba, and Hausa without any predefined scenarios. The domain of the chatbot is open. The chatbot takes as input a textual question to look for corresponding answers within a large textual dataset. They implemented a system that works on modified StarSpace embedding [17] tailored for African dialects for the question-answering task. Data were collected from official sources. For instance, Tunisian information was provided by the Ministry of Health (MoH) and Nigerian information from Nigerian local non-governmental organizations (NGOs). Data is divided into two main categories: Frequently Asked Questions and chitchat. To measure the chatbot's quality, they used the Sensibleness and Specificity Average (SSA), which is a human evaluation metric proposed by Google Brain Team. The sensibleness was 68%, and specificity was 60%.

Bashir et al. [13] proposed a neural network technique to build a natural language understanding model that consists of two components: the intent classifier and the entity extractor. For this Arabic task-oriented chatbot for home automation, they used DL model such as Ara-Vec, LSTMs, CNNs, character-based word embeddings and implemented a generative-based open-domain system. They also utilized DL algorithms for text classification and named entity recognition. The data was gathered using an online survey that received 768 responses via Twitter.

B. *Closed-domain*

This section presents chatbots studies whose domain is closed.

"Labeeb" chatbot (Wiseman in Arabic) by Almurtadha [18] was implemented to response to student inquiries. It is a retrieval-based chatbot. The domain is closed for students' inquiries. This is to solve the problem of the overwhelming number of student inquiries that need immediate reactions, as opposed to the limited number of lecturers with limited time. For that reason, they implemented an intelligent text-based and voice-based Arabic chatbot. The author used Microsoft Speech Synthesizer, which uses Microsoft's conversational speech recognition system. Although the authors proposed the chatbot, they did not provide any details on the NLP models or any performance measures.

The "dialogue system" was proposed by Bendjamaa and Taleb [19] based on an existing Qur'anic ontology. The ontology represents a conceptualization of the Qur'an, which facilitates access to the information contained in this sacred book. Its domain is closed on the Qur'an book: Qur'anic chapters and verses. It does not cover a word morphology search, but they added links to the Qur'anic vocabulary to cover the pronouns. This system allows easy access to Qur'anic information interactively by SPARQL Protocol and RDF Query Language (SPARQL) queries to test whether it can find the verses and concepts from pre-existing Qur'anic ontology. This project was coded in Java and used the CoreNLP plugin. The ontology target is to build a computational model capable of representing all the concepts mentioned in the Qur'an and the relations between them using Protégé Web Ontology Language (OWL). The ontology could be questioned with SPARQL requests. Protégé allows editing, visualization, control, and fusion of ontologies [20]. Although the authors proposed the chatbot they provided and cases that the system could handle and respond to, they did not provide any measurement of the chatbot's performance.

Joukhadar et al. [8] implemented a retrieval-based chatbot. Its domain is closed with eight types of dialogue. The system is used to recognize the dialogue acts of the users in a textual chatbot using the Levantine Arabic dialect. Dialogue acts have eight types: greeting, goodbye, thanks, confirm, negate, ask to repeat, ask for alternative, and apology. The dataset was

collected manually with a total of 873 sentences from restaurant orders and airline ticketing platforms. They used five ML models and built an ensemble model to identify dialogue act. The best results were achieved using Support Vector Machine (SVM) model with 86% accuracy.

Fadhil and AbuRa'ed [21] implemented an Arabic chatbot named OlloBot to assist physicians and support patients with the care process. It provides health-tracking, support, and assists physicians with care delivery through a conversation medium deployed on the Telegram platform. The dialogue flow is handled by IBM Watson Conversation. The chatbot domain is closed for patients with care process. OlloBot architecture is a generative-based application that uses PM. To test the system the ample size was 43 native speakers using a framework of 30 questions to measure the usability of the chatbot. The framework tests four items of usability: usefulness, ease of use, ease of learning, and satisfaction with the application. The dialogue and various questions the bot asked are based on a World Health Organ[22]ization (WHO) questionnaire [22] .

"LANA-I" by Aljameel et al. [23] is an Arabic conversational intelligent tutoring system (CITS) for children with autism spectrum disorder (ASD) that adapts to the visual, auditory, and kinesthetic learning styles model to enhance learning. Children with high-functioning autism have difficulties in social and communication skills, with a shortage of specialized teachers to deal with their specific needs. One solution is to use a virtual tutor to supplement the education of children with ASD in mainstream schools. This study took place in the UK with neurotypical children from the target age group (10-12 years) whose first language is Arabic. The engine of LANA-I is based on the two main chatbot development strategies: the PM engine and a short text similarity (STS) algorithm that calculates the matching strength of a pattern to the user utterance. The results show that LANA-I is effective as an Arabic chatbot, with the majority of conversations leading to the goal of completing the tutorial (95.8%) and providing correct responses (89%). The results illustrated that 57.18% of all the utterances input by the users were different from the scripted patterns, and in this case, the system used the STS algorithm with cosine algorithm. The log file shows that 34 unique utterances relating to 15 different rules were correctly recognized and dealt with by the LANA-I algorithm by firing the appropriate rule. Although LANA-I is able to lead the user toward the goal of the tutorial, the results have identified some weaknesses in the LANA-I architecture, mainly concerning the number of unrecognized utterances. LANA-I failed to recognize some utterances from the users (38%). The platform uses MySQL with Java programming language to create the CITS applications and uses the science curriculum for 10-12-year-olds as the source.

The Arabic flight booking chatbot was developed by Al-Ajmi and Al-Twairesh [24] to serve airline ticket booking. The chatbot domain is closed, and the sources of the chatbot dataset are retrieval-based. They proposed a hybrid rule-based and data-driven approach in a text-based flight booking. The chatbot is capable of handling customers' utterances and providing the right response. The proposed chatbot was built utilizing the Wit.ai natural language interface, and the Telegram Messenger framework was used as the chatbot interface for the system's dialogue manager. The conversation flow was configured using the Wizard of Oz technique, which is an experimental evaluation mechanism. It allows the observation of a user operating an apparently fully functioning system whose missing services are supplemented by a hidden wizard [25]. The chatbot intents and entities were developed using crowdsourcing training examples because the collected initial knowledge was not sufficient to enhance the system's ability to detect the desired values for the created rule-based entities and the trained data-driven entities. In total, 1,651 training examples were crowdsourced. Of these, 1,518 were positive examples and 133 were negative examples (i.e., nothing could be detected from these examples). A total of 472 locations were collected manually from the Saudia airline website and different sources in the format ''city_name airport_code''. The countries and airport names were extracted manually from Wikipedia, and the data were pre-processed by removing duplicated airport codes and train station records. Also, approximately 200 preliminary knowledge booking scenarios were gathered from volunteers. The evaluation results show that the system developed was able to understand user utterances and to self-feed efficiently. To evaluate the performance of the chatbpt, they asked the participants if the identity of the interlocutor in the text-based booking channel would make a difference in the resulting ticket booking, and 63.64% of the first-stage participants answered ''Yes'' because it would understand them better if it were a human agent. For the second-stage participants, 50% of them answered ''Yes'' because it would be more accurate and understand them better if it were a human agent. Also, 52% of participants had difficulty in understanding Telegram commands. This indicates that the ability of the machine and bot to understand human utterances is still questioned. Also, the sample size that tested the chatbot was 21.

El-Saadawi, et al. [26], proposed a chatbot for teaching Prophet Muhammad's biography and Islamic history. The interaction in this chatbot is via speech between the human and the computer. The input is inspected for the presence of a keyword. If such a word is found, the sentence is mapped according to a rule associated with the keyword and is retrieved. The chatbot's conversational domain is closed, designated for the Prophet's biography. The dataset source is Islamic websites on the Prophet's biography and Islamic history information. The chatbot platform is Facebook. The chatbot framework is composed of three main phases: interaction phase, processing phase, and learning phase. The proposed application was programmed using the Mongo database and PHP language accompanied by an Apache Solr search engine. The data were stored on a Heroku cloud server. The paper presented a sample of random queries to the chatbot, together with the chatbot's answers. According to the chatbot structure, the same answer can be generated for different queries with the same meaning. Although the authors proposed the chatbot and a sample of its function, they did not provide any measurement of its performance

Alotaibi et al. [12] implement a tourist chatbot application. The domain of conversation is closed for tourism recommendations. The chatbot is an AI text-based chatbot that was designed as a generative-based smartphone application to help people explore the city and arrange trips more simply and effectively. TensorFlow embedding classifiers are used to embed user communications and intent labels into the same space. Then, they proceeded with feature extraction into vectors. An online survey was used to collect data, which was then distributed throughout the city of Jeddah.

The dataset includes 160 samples of user messages, such as greetings and goodbyes. The chatbot was built by utilizing the Rasa.ai to provide the anatomy of the rasa-based chatbot coding and an LSTM neural network implemented in Keras, which allows the chatbot to memorize messages in the training data file and predict the next action. The authors tested the application to check the speed and response time. For example, if the request was on the nearest places, the response was 14 seconds. If the user asked for specific information or details about a specific place, the chatbot took three seconds. If the user asked for the weather or to find places in Jeddah, it took four seconds, and if the chatbot got an unknown or out-of-scope message, the response took two seconds.

Aljojo et al. [27] proposed a puzzle game application based on eye tracking to assist with focusing attention for individuals with dyslexia, as well as a chatbot that can motivate users. The dataset was collected from social media. The parent sample size was 317, while the sample size of professionals working with dyslexics, such as teachers, was 11, based on the replies received from Twitter and WhatsApp platforms. The study's objective is to develop an application by using three different tests, which are 'ace it', 'unscramble,' and 'practice', which provide the number of letters in the word. It may be directed toward children aged between four and seven with dyslexia who have excessive movement and lack of attention. The sources of the chatbot dataset are retrieval-based, and the domain is closed for dyslexia. Ranking analysis was applied to 317 parent and 11 professional who work with dyslexics students, in order to rate the quality of the proposed Arabic Alphabetic Puzzle Game Using Eye Tracking and Chatbot for Learning Disability (Dyslexia) from their perspectives.

III. Methodology and Architecture

This paper presents three versions of Rahhal, a chatbot that aims to aid tourists in visiting different cities in Saudi Arabia. All three are deployed on the Telegram platform. The first, Saudi_Rahhal_Bot, is coded with Java to chat in Arabic. The second, SaudiRahhalBot, is programmed with IBM Watson [28] to chat in Arabic as well, and the third, EngRahhalbot, is programmed with IBM Watson to chat in English to help non-Arab tourists. The chatbots are based on a dataset that covers 13 areas and 81 cities in Saudi Arabia and it has more than 845 data entry that could be provided to the user upon request. In the following, we present the dataset description, Java-based chatbot, and IBM Watson chatbot.

A. Dataset Description

Data were collected from different sources, including the following applications Jawlah [29], Gathern [30], and Almosafer [31]. The first is an application for touring information that is presented in a catalog style. Gathern is an application for reserving vacation houses and camping areas in Saudi. Almosafer is an application for hotel and flights reservation based in Saudi. Data were aggregated containing all areas and major cities in Saudi Arabia with their tourist attractions and activities. One of the major curations we made to the dataset is that we wrote a description for each activity, added the activity location and activity account or website. The dataset header is depicted in Table 1.

TABLE 1.The Dataset Header In Both Arabic And English Languages.

Area	**المنطقة**
City	**المدينة**
Activity type	**نوع النشاط**
Activity name	**اسم النشاط**
Activity description	**وصف النشاط**
Contact	**التواصل**

Data were collected covering the 13 areas in Saudi Arabia and 81 cities containing activities, tourist attractions, heritage and culture, nature and environment, malls and shopping, restaurants, resorts and hotels. The total number of activities was more than 845 in all cities. To create the English dataset, the Arabic dataset was translated manually by a language expert with the aid of Google Translate.

B. System Implementation and Integration

The chatbots we proposed are closed domain, provide flexibility and efficiency of use offering buttons and text. The chatbot relies on pattern matching modeling. Moreover, one of them is developed using java language and IBM Watson. The data contained more than 845 activates. All chatbots implemented are deployed on telegram. In the following two sections we will be explaining the chatbots, Java-based and IBM Watson-based.

1) Java Based Chatbot

The chatbot was programmed using Java language on IntelliJ IDE, and the org.telegram.telegrambots library. On implementing the chatbot we had two choices, either allow the user to type in his request, or show the available options in a menu and the user selects the desired option by simply clicking on it. We preferred the latter as it complies with Nielsen usability heuristics by providing flexibility and efficiency of use offering buttons that minimizes time to type and minimizes errors [32]. After writing the Java code, we chatted with FatherBot on Telegram platform to receive the username and token (API) to create the three bots. FatherBot is the official bot created by Telegram to facilitate bot creation and management.

Article number	Approach	Dataset Model	Domain	Interaction		Data Source & Size	Deployment	Development	ML/DL models
				Input	Output				
[8]	PM	Retrieval-based	Closed (8 types of dialogue)	Text / Voice	Text	Manually 873 tagged sentences from Restaurant orders and airline ticketing	NA	Python	ML: LR, SVM, MNB, ET, and RF
[9]	NLP and ML	Retrieval-based	Open	Text	Text	Prophet biography book [15]	Telegram	Python: NLTK	ML: NA
[10]	NA	NA	Open	Text	Text	From MoH and Nigerian NGOs	Web interface and Facebook Messenger	NA	ML: StarSpace neural model
[11]	PM	Retrieval-based	Open	Text	Text	5.2k Twitter	NA	Python	DL: LSTM based on Seq2Seq, Transformer-based
[12]	AIML	Generative-based	Closed (Tourism recommendations)	Text	Text	160 examples from online survey	Android Application	Python: Rasa.ai framework	DL: LSTM
[13]	PM	Generative-based	Open	Text	Text	768 from an online survey	NA	NA	DL: CNN, LSTM AraVec
[16]	PM and AIML	Retrieval-based	Open	Text	Text	248 inputs / outputs from KSU IT students' accounts in Askme.com	Twitter, and Web interface	Pandorabots Playground	Not used
[18]	PM	Retrieval-based	Closed (student inquiries)	Text / Voice	Text / Voice	NA	NA	NA	Not used
[19]	PM	Retrieval-based	Closed (Qur'an holy book)	Text / Voice	Text / Voice	Qur'anic chapters and verses	Desktop application	Java: Core NLP plugin	Not used
[21]	PM	Generative-based	Closed (patients)	Text	Text	30 questions from WHO questionnaire	Telegram platform	IBM Watson	Not used
[23]	PM and STS	Retrieval-based	Closed (children with ASD)	Text	Text	Science curriculum for the 10-12 years age	Web interface	Java	Not used
[24]	PM	Retrieval-based	Closed (airline flight booking)	Text	Text	472 locations from Saudia airline and different sources	Telegram	Python: Wit.ai framework	Rule-based

[26]	AIML	Retrieval-based	Closed (Prophet Muhammad biography)	Text	Text	Islamic websites	Facebook	MongoDB and PHP with Apache solr	Not used
[27]	AIML	Retrieval-based	Closed (Dyslexia)	Text / Voice	Text / Voice	328 sample size from social media	Android Application	Java	Not used

TABLE 2. SUMMARY OF ARABIC CHATBOT RELATED WORK.

After that, the username and token were imbedded in the Java code to enable the connect to the bot. After testing the Java files locally, the chatbot was then deployed using Heroku platform [33]. Heroku is a platform that enables developers to build and run applications on the cloud. The Java files containing the code and the chatbot API to connect to the Telegram Platform are pushed to Heroku. Running the chatbot live on the cloud enables the users to chat with it 24/7. The communication architecture is shown in Fig. 1.

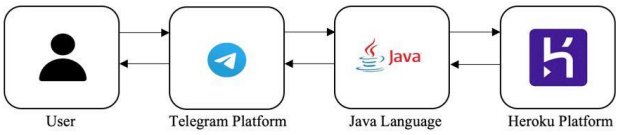


Fig. 1. Java based chatbot architecture.

A user can start chatting with the chatbot using the Telegram Platform as shown in the in Fig. 2. To start chatting the user can search for the chatbot handle, Saudi_Rahhal_Bot, in this case, and click it. After that, a chatting screen appears for the user to click start, Fig. 2a. then the chatbot will display lists or buttons and options to the user to help him choose. Also, when the user press "who are you button"? the chatbot will response with a description of Rahhal bot, as shown in Fig. 2a.

When the user presses "Rahhal be my tour guide", Fig. 2a., the chatbot will display buttons and options that contain all areas that he or she can select from. When an \ an area is selected, Fig. 2b, a list of cities is displayed then on selecting a city, activities in that city are displayed. The user then selects the type of activity he desires, then the chatbot will show all available options under that activity, whether parks, entertainment venues, markets or even restaurants, as shown in Fig. 2c. in the final step, the activity information is displayed along with a link for extra information.

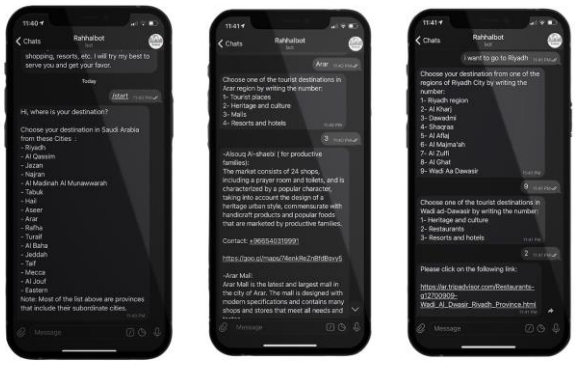
2) IBM Watson-Based Chatbot

To provide international accessibility in the chatbot, we implemented two versions: SaudiRahhalBot in Arabic and EngRahhalBot in English. Both are implemented using IBM Watson, which provides artificial intelligence means that support catching different user intents and

(a) (b) (c)

Fig. 2. Saudi_Rahhal_Bot screenshots where (a) is the start (b) are regions of Saudi Arabia and (c) are the place options.

entities. To implement the chatbot, we have two options. The first is to use the similarity feature that enables the user to enter a wide range of possible inputs, and the system recognizes those entries by similarity measures. This is a useful feature that tolerate spelling mistakes and typos. However, it still requires the user to enter a considerable number of characters, and this is costly in terms of time. In addition, one of the Nielsen usability heuristics is flexibility and efficiency of use, which basically refers to providing the user with accelerators like keyboard shortcuts that speed up user entraction [32]. Thus, we implemented the second option, which provides menus with numbered options.



(a) (b) (c)

Fig.3. EngRahhalbot screenshots where (a) is the start, (b) is the destination, and (c) is the region of Riyadh.

The user type in the number of the option they want to select instead of typing whole sentences and words. The chatbots provide the user with topics they can chat about; then, based on the user input, the bots forward the request to the IBM Watson conversation cloud.

The conversation slots take user input and provide them with a relevant answer after checking their intent, entity, and condition of the conversion.

Also, the bots can extract and specify the purpose of what the users want or the objective behind a user's input. For example, if the user typed "hi, hello," the chatbot will translate this to the "Greetings" intent. The city names will be translated into the "Cities" intent. The bot's response depends on what the user types or chooses from the options. For example, after the user chooses a city to visit, the chatbot builds a list of activities or tourist destinations. Moreover, the entities are available, which is focused on defining the topic the user is talking about by an important keyword extracted from a sentence. For example, if the user typed "I want to go to Jeddah", the word "Jeddah" is detected as an entity, and hence the bot switches to the tourist destinations in Jeddah city topic, as shown in Figure 3c. We created up to 111 entities to build our dialogue model. The entity list was accompanied by a list of synonyms to add flexibility for entity detection. This is important to detect user intents and sentiments from their conversation and provide the right response to their questions. Finally, to handle fallbacks where the bots have no clue about the response, we designed the "anything else" dialogue to handle unhandled intents.

The fact that chatbots are supported by Watson and easily accessible 24/7 makes them a convenient assistant for tourists with zero cost. Our chatbot can handle up to 1,000 requests per month. A user can start chatting with the chatbots using the Telegram platform. To start using the chatbot, a user can search for the chatbot handles, then a chatting screen appears for the user to click start, as shown in Fig. 3a. Then, the chatbot dialogue is activated, as shown in Figures 3b and 3c. The same is applicable for the Arabic chatbot SaudiRahhalBot.

IBM Watson supports different deployment platforms, such as Slack, Facebook Messenger, WhatsApp with Twilio, and more. We tried to deploy the chatbots in Facebook Messenger, but since Facebook is committed to providing appropriate data privacy, security, and governance solutions, we had to submit a Subject Access Request through the IBM Cloud Service Portal to Facebook by creating a new case. The reply took longer than we expected we deployed our chatbots using the Telegram platform via Node-RED [34], as shown in Fig. 4.

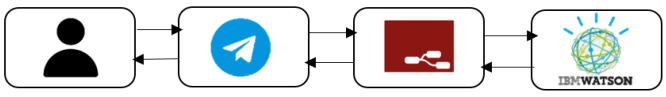


Fig. 1 IBM Watson-based chatbot architecture.

IV. Evaluation Criteria

To ensure the usability of Rahhal chatbots, we asked 300 users to use these chatbots and provide feedback by answering a web-based questionnaire.

In the questionnaire, we asked the users questions to ascertain the criteria of efficiency, satisfaction, and effectiveness, and hence, the usability of the chatbots. To evaluate effectiveness, we focused on performance, such as flexibility to input and speed of response. To evaluate efficiency, we were concerned with the linguistic accuracy of the outputs and the ability to respond to specific questions. To evaluate satisfaction, we focused on greetings and ease of communication with Rahhal. Thus, we divided our questions into several categories:

- The methodology of Rahhal in conducting conversation;
- Quality of information provided by Rahhal;
- Rahhal's abilities;
- Conversation with Rahhal in general.

V. Results and Discussion

To test the usability of our chatbots, we designed a survey inspired by Al-Ghadhban et al.[16]. Namely, we tested the usability of the Arabic Java-based chatbot Saudi_Rahhal_Bot as well as the Arabic and English-based IBM Watson chatbots SaudiRahhalBot and EngRahhalbot. For all surveys, the number of participants was 100. The overall evaluation of the ease of use of all three chatbots was easy. That is, both Arabic chatbots were evaluated as 100% easy to use, whereas 4% of the English chatbot users evaluated it as difficult. This could be attributed to the fact that many of the users who tested the English chatbot speak English as a second language. Although the English chatbot is intended for people who do not speak Arabic, the participant sample of the usability test could not be completed with only those people. The interfaces of the chatbots are somewhat different. While the Java-based chatbot interface offers options as buttons, the IBM Watson-based chatbots require the user to type in the number of the option. Having options as buttons in the interface instead of requiring the user to type in those options minimizes the required effort from the user. This difference is prominent in the users' impressions of the overall ability of the chatbot. That is, 87% of users evaluated the Java-based chatbot's ability to conduct a conversation as excellent. In contrast, the IBM Watson chatbots were evaluated at 79% for the English chatbot and 64% for the Arabic chatbot (Table 3). This difference also affected the users' satisfaction when chatting with the chatbots; these evaluations were 83% for the English IBM Watson, 81% for the Arabic IBM Watson, and 90% for the Java-based chatbot. The difference in satisfaction between the typing interfaces provided by IBM Watson and the button interface in the Java-based chatbot was up to 7%.

TABLE 3. RAHHAL CHATBOTS USABILITY TEST RESULTS SHOWN AS THREE LINES, (1) THE FIRST LINE IS THE ENGRAHHALBOT CODED IN IBM WATSON TO CHAT IN ENGLISH, (2) THE SECOND LINE IS THE SAUDIRAHHALBOT CODED IN IBM WATSON TO CHAT IN ARABIC, AND (3) THE THIRD LINE IS THE SAUDI_RAHHAL_BOT CODED IN JAVA TO CHAT IN ARABIC.

Category	Question	Results			
Methodology	How was the experience of your communication with Rahhal chatbot?	**Easy** 96% 100% 100%	**Difficult** 4% 0% 0%		
	Have you been welcomed?	**Yes** 98% 99% 100%	**No** 2% 1% 0%		
Quality of information	How would you evaluate the level of information you got from Rahhal chatbot?	**Bad** 2% 0% 0%	**Acceptable** 11% 6% 3%	**Good** 14% 29% 14%	**Excellent** 73% 65% 83%
	Was Rahhal chatbot accurate in giving answers?	**Not accurate** 3% 2% 2%	**Somewhat accurate** 15% 37% 15%	**Very precise** 82% 61% 83%	
	Have Rahhal chatbot answers been appropriate to the context of your request?	**Not appropriate** 5% 3% 1%	**Suitable** 19% 39% 26%	**Very suitable** 76% 58% 73%	
Abilities	How would you rate the language used in Rahhal chatbot in terms of accuracy and linguistic accuracy?	**Bad** 0% 0% 0%	**Acceptable** 8% 8% 3%	**Good** 18% 22% 12%	**Excellent** 74% 70% 85%
	Did Rahhal chatbot interact with your conversation?	**Yes** 96% 96% 95%	**No** 4% 4% 5%		
	In general, was Rahhal chatbot able to conduct a conversation well?	**Bad** 6% 1% 1%	**Good** 15% 35% 12%	**Excellent** 79% 64% 87%	
	Has Rahhal chatbot replied promptly?	**Slow** 4% 6% 2%	**Fast** 96% 94% 98%		
Conversation	How was chatting with Rahhal chatbot?	**Bad** 4% 1% 1%	**Acceptable** 19% 29% 11%	**Good** 77% 70% 88%	
	How satisfied are you with Rahhal chatbot?	**Dissatisfied** 3% 0% 0%	**Somewhat satisfied** 14% 19% 10%	**Totally satisfied** 83% 81% 90%	
	Will you chat with Rahhal chatbot again?	**Yes** 94% 96% 99%	**No** 6% 4% 1%		

VI. Conclusion

This paper presents three different implementations on a tourism chatbot. Those three versions are Arabic and English chatbots implemented in IBM Watson and an Arabic chatbot implemented in Java. Also, it provides a survey on Arabic chatbots covering 14 different studies. The studies were presented and evaluated based on the implementation technique, the conversation interaction, domain, and the model used for the chatbot dataset. The evaluation shows that most of the chatbots under review are retrieval-based 18 opposed to four only that are Generative-Based dataset models. The ways used to build Arabic chatbot is Pattern Matching, AIML. The usability test of our chatbots show a relatively high satisfaction of the users, as they evaluated them as 99% easy on average. Also, it shows that users prefer the interface that offers buttons instead of the interface that requires them to type in requests with a difference of at least 6% in the evaluation of conversation to the favor of the button interface.

References

[1] S. A. Al-Ghamdi, J. Khabti, and H. S. Al-Khalifa, "Exploring NLP web APIs for building Arabic systems," in *2017 Twelfth International Conference on Digital Information Management (ICDIM)*, Sep. 2017, pp. 175–178. doi: 10.1109/ICDIM.2017.8244649.

[2] B. Liu *et al.*, "Content-Oriented User Modeling for Personalized Response Ranking in Chatbots," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 1, pp. 122–133, Jan. 2018, doi: 10.1109/TASLP.2017.2763243.

[3] B. A. Shawar, "A Chatbot as a Natural Web Interface to Arabic Web QA," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 6, no. 1, Art. no. 1, Mar. 2011.

[4] N. Y. Habash, "Introduction to Arabic Natural Language Processing," *Synthesis Lectures on Human Language Technologies*, vol. 3, no. 1, pp. 1–187, Jan. 2010, doi: 10.2200/S00277ED1V01Y201008HLT010.

[5] S. Al-Humoud, A. Asma, and W. Aldamegh, "Arabic Chatbots: A Survey," *International Journal of Advanced Computer Science and Applications*, vol. 9, Jan. 2018, doi: 10.14569/IJACSA.2018.090867.

[6] T. Hak and J. Dul, "Pattern Matching," Social Science Research Network, Rochester, NY, SSRN Scholarly Paper ID 1433934, 2009. Accessed: Sep. 08, 2021. [Online]. Available: https://papers.ssrn.com/abstract=1433934

[7] G. D. Gasperis, "Building an AIML Chatter Bot Knowledge-Base Starting from a FAQ and a Glossary," *Journal of e-Learning and Knowledge Society*, vol. 6, no. 2, pp. 75–83, 2010.

[8] A. Joukhadar, H. Saghergy, L. Kweider, and N. Ghneim, "Arabic Dialogue Act Recognition for Textual Chatbot Systems," in *Proceedings of The First International Workshop on NLP Solutions for Under Resourced Languages (NSURL 2019) co-located with ICNLSP 2019-Short Papers*, 2019, pp. 43–49.

[9] S. Mahamat Yassin and M. Khan, "SeerahBot: An Arabic Chatbot About Prophet's Biography," *International Journal of Innovative Research in Computer Science & Technology (IJIRCST)*, May 2021, doi: 10.21276/ijircst.2021.9.2.13.

[10] A. B. E. Mabrouk, M. B. H. Hmida, C. Fourati, H. Haddad, and A. Messaoudi, "A Multilingual African Embedding for FAQ Chatbots," *arXiv:2103.09185*, Mar. 2021, Accessed: Oct. 07, 2021. [Online]. Available: http://arxiv.org/abs/2103.09185

[11] T. Alshareef and M. A. Siddiqui, "A seq2seq Neural Network based Conversational Agent for Gulf Arabic Dialect," in *2020 21st International Arab Conference on Information Technology (ACIT)*, 2020, pp. 1–7. doi: 10.1109/ACIT50332.2020.9300059.

[12] R. Alotaibi, A. Ali, H. Alharthi, and R. Almehamdi, "AI Chatbot for Tourist Recommendations: A Case Study in the City of Jeddah, Saudi Arabia," *International Journal of Interactive Mobile Technologies*, vol. 14, no. 19, pp. 18–30, 2020.

[13] A. M. Bashir, A. Hassan, B. Rosman, D. Duma, and M. Ahmed, "Implementation of A Neural Natural Language Understanding Component for Arabic Dialogue Systems," *Procedia Computer Science*, vol. 142, pp. 222–229, Jan. 2018, doi: 10.1016/j.procs.2018.10.479.

[14] J. Wang, *Encyclopedia of Business Analytics and Optimization (5 Volumes)*. IGI Global, 2014. Accessed: Jan. 31, 2022. [Online]. Available: https://www.igi-global.com/book/encyclopedia-business-analytics-optimization/www.igi-global.com/book/encyclopedia-business-analytics-optimization/90651

[15] M. Al-Azmi, *Al Luwlu Al maknun fi Sirat Al nabii Al Mamun "Dirrasa Muhaqaqa Lil Siyra AL Nabawia,"* vol. Third, 4 vols. Dar Alsomaie, 2013. Accessed: Sep. 03, 2021. [Online]. Available: https://ar.islamway.net/book/16850

[16] D. Al-Ghadhban and N. Al-Twairesh, "Nabiha: An Arabic dialect chatbot," *Int. J. Adv. Comput. Sci. Appl*, vol. 11, no. 3, pp. 1–8, 2020.

[17] L. Wu, A. Fisch, S. Chopra, K. Adams, A. Bordes, and J. Weston, "StarSpace: Embed All The Things!," presented at the The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18), Sep. 2017.

[18] Y. Almurtadha, "LABEEB: Intelligent Conversational Agent Approach to Enhance Course Teaching and Allied Learning Outcomes Attainment.," *Journal of Applied Computer Science & Mathematics*, vol. 13, no. 27, pp. 9–12, 2019.

[19] F. Bendjamaa and T. Nora, "A Dialogue-System Using a Qur'anic Ontology," in *Second International Conference on Embedded Distributed Systems (EDiS)*, Nov. 2020, pp. 167–171. doi: 10.1109/EDiS49545.2020.9296437.

[20] T. Slimani, "Ontology Development: A Comparing Study on Tools, Languages and Formalisms," *Indian Journal of Science and Technology*, vol. 8, pp. 1–12, Feb. 2015, doi: 10.17485/ijst/2015/v8i1/54249.

[21] A. Fadhil and A. AbuRa'ed, "OlloBot - Towards A Text-Based Arabic Health Conversational Agent: Evaluation and Results," in *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP)*, Varna, Bulgaria, Sep. 2019, pp. 295–303. doi: 10.26615/978-954-452-056-4_034.

[22] "Obesity and overweight." https://www.who.int/news-room/fact-sheets/detail/obesity-and-overweight (accessed Oct. 07, 2021).

[23] S. Aljameel, J. O'Shea, K. Crockett, A. Latham, and M. Kaleem, "LANA-I: An Arabic Conversational Intelligent Tutoring System for Children with ASD," 2019, pp. 498–516. doi: 10.1007/978-3-030-22871-2_34.

[24] A.-H. Al-Ajmi and N. Al-Twairesh, "Building an Arabic Flight Booking Dialogue System Using a Hybrid Rule-Based and Data Driven Approach," *IEEE Access*, vol. 9, pp. 7043–7053, 2021, doi: 10.1109/ACCESS.2021.3049732.

[25] D. Salber and J. Coutaz, "Applying the Wizard of Oz technique to the study of multimodal systems," in *Human-Computer Interaction*, Berlin, Heidelberg, 1993, pp. 219–230. doi: 10.1007/3-540-57433-6_51.

[26] O. El-Saadawi, A. El-Saadawi, and M. El-Saadawi, "A Proposed Framework of an Intelligent Arabic Chatbot for Teaching Islamic History," *International Journal on Islamic Applications in Computer Science And Technology*, vol. 5, no. 4, pp. 14–24, 2019.

[27] N. Aljojo *et al.*, "Arabic Alphabetic Puzzle Game Using Eye Tracking and Chatbot for Dyslexia," *International journal of interactive mobile technologies*, vol. 12, no. 5, pp. 1865–7923, 2018, doi: 10.3991/IJIM.V12I5.8957.

[28] "IBM Watson," May 27, 2021. https://www.ibm.com/watson (accessed Aug. 19, 2021).

[29] "Saudi Tourism Tour." https://play.google.com/store/apps/details?id=com.jadara.gawla&hl=ar&gl=US (accessed Sep. 04, 2021).

[30] G. Holding, "Gathern." https://gathern.co/site/index (accessed Sep. 04, 2021).

[31] "Hotels, book cheap hotels & get discounted prices on Almosafer," *Almosafer*. https://www.almosafer.com/en (accessed Sep. 04, 2021).

[32] W. L. in R.-B. U. Experience, "Usability Inspection Methods: Book by Jakob Nielsen," *Nielsen Norman Group*. https://www.nngroup.com/books/usability-inspection-methods/ (accessed Aug. 18, 2021).

[33] "Cloud Application Platform | Heroku." https://www.heroku.com/ (accessed Aug. 18, 2021).

[34] "Node-RED." https://nodered.org/ (accessed Aug. 18, 2021).