

Building a Machine Learning Powered Chatbot for KSU Blackboard Users

Qubayl Alqahtani, Omer Alrwais

Information Systems, King Saud University, Riyadh, Saudi Arabia

Abstract—Chatbots have attracted the interest of many entities within the public and private sectors locally within Saudi Arabia and also globally. Chatbots have many implementations in the education field and can range from enhancing the e-learning experience to answer students' inquiries about course schedules and grades, tracking prerequisites information and elective courses. This work aim is to develop a chatbot engine that helps with frequently asked questions about the Blackboard system, which could be embedded into the Blackboard website. It contains a machine-learning model trained on Arabic datasets. The engine accepts both Arabic textual content as well as English textual content if needed; for commonly used English terminologies. Rasa framework was chosen as the main tool for developing the Blackboard chatbot. The dataset to serve the current need (i.e. Blackboard system) was requested from Blackboard support staff to build the initial dataset and get a sense of the frequently asked questions by KSU Blackboard student users. The dataset is designed to account for as many as possible of KSU Blackboard related inquiries to provide the appropriate answers and reduce the workload of Blackboard system support staff. Testing and evaluating the model was a continuous process before and after the model deployment. The model post-tuning metrics were 93.4%, 92.5%, 92.49% for test accuracy, f1-score and precision, respectively. The average reported accuracy in similar studies were near 90% on average as opposed to results reported here.

Keywords—Chatbot; RASA; conversational agents; machine learning

I. INTRODUCTION

A chatbot is an artificial intelligence (AI) software that can simulate a conversation (or a chat) with a user in natural language through messaging applications, websites, mobile apps or through the telephone [5]. It's an environment that receives questions from users in natural language, relates these questions with a knowledge base, and then answer based on pre-defined answers. Chatbots are more formally referred to in the literature as conversational agents or conversational assistants. The core principle of every conversational agent is to interact with humans using text messages and act as it were able to understand the user and replay with the appropriate message. The origin of computers talking to humans goes back to the start of the computer science field itself. Alan Turing defined a simple test referred to now as the Turing test back in 1950 where a human judge would have to predict if the entity they are communicating with via text is a computer program or not [6]. However, this test's scope is way greater than the case of chatbots, the main difference being that the domain knowledge of a chatbot is narrow compared to the Turing test.

Turing test assumes one can talk about any topic in mind with the agent.

Conversational agent environment consists of five different main parts [7]. Starting with user messages: they are a dynamic input received by the agent to process and replay. They contain a string representation of the actual text sent by the user, and a metadata that contains additional information like a reference to the session the conversation belongs to, and possibly the date and time the message was sent to the agent, on which platform the message was sent from if the agent is linked to more than one message, etc. The agent receives the message along with the information it contains in a read- only mode only with no possible means of making changes to it. The backend is one of the significant parts of the environment that the agent has access to. It contains additional information about the agent users and the database's states to store the user messages, their metadata, and keep track of the conversation events. The agent can view and update certain aspects of the backend. The chatbot can also obtain new information from the user if necessary, by asking the user to provide it.

RASA is a modular design framework proposed by [1] that consists of two main components, Rasa Core for dialog management and Rasa NLU for natural language understanding. Those are open- sourced python libraries for building machine-learning based conversational agents. They provide dialog management and NLU capabilities in an easy manner. By nature, a modular by design architecture allows for easier integration of modules with other systems and services. For instance, Rasa NLU can be used as a service in a different system other than rasa by exposing HTTP APIs for external requests and vice versa for Rasa Core. The code can be found by visiting: <https://github.com/RasaHQ>.

Although chatbots have been present for a long time, 2016, before the spring, was the true start of this technology. There are two main reasons for the renewed interest in chatbots (1) massive advances in artificial intelligence (AI) and (2) a major usage shift from online social networks to mobile applications such as WhatsApp, Telegram, Slack, and many more advances in AI holds a promise that intelligent chatbots are in fact, can be within reach. The increased usage of mobile applications attracts service providers to reach users through them. However, in spite of these advances, chatbot applications entail many challenges that need to be overcome in order to reach desired goals. Chatbots not only imply changes in the interface between users and technology; they imply changing user dynamics and usage patterns. A recent study indicated that 56% of chatbot users were interested in ordering meals from

restaurants using chatbots, while 34% had already ordered at least one meal [2]. Chatbots are considered to be beneficial for retailers in terms of customer service (about 95%), sales/marketing (about 55%), and order processing (about 48%) [3]. Generation Z and Millennials are more interested in using chatbots: 25% of a global sample aged 18 to 34 opted for a personal shopping chatbot [4], and the students using Blackboard system fall under this age range.

King Saud University (KSU) is the largest public university in Saudi Arabia and at any time encompass thousands of students whom some of them use and struggle with the online course and learning management system (blackboard), as such the customer support staff (which is not more than three employees) at KSU is overwhelmed with enquiries which can cause great user dissatisfaction and affect the adaptation of Blackboard system by KSU students. The goal of this work is to build a Minimum Viable Product (MVP) for an Arabic Chatbot which is intended to serve users of Blackboard system from King Saud University students by answering their frequently asked questions about the Blackboard system to reduce the load of answering repeated, one answered questions and allow customer service staff to focus on more dynamic issues that require human intervention.

II. RELATED WORK

A revision for several chatbot related papers that highlights the usage of chatbots in the education field was conducted, along with other chatbot usages in different areas such as retail and government entities. A review and summarization of those implementations are discussed in the next paragraphs.

A. Chatbots in Education

A chatbot named EASElective that was built to advise students on what to choose as an elective course was proposed in [22]. EASElective is a conversational agent that was built to supplement existing academic advising systems. It has an interactive, online interface that supports basic official course information to informal students' opinions about that course. Its major components included intent detection, conversational management routines, dialogue design, course information management, and a collection of analyzed students' peers' opinions. In this study, a survey was conducted to capture students' perceptions of the chatbot. The subjects were briefed about the chatbot's purpose and instructed on how to use it and were given up to a half-hour to interact with it and then fill the surveys. The survey results showed that many students preferred to either ask their friends for course information. Around 22% preferred to ask the program leader or use the official university website instead of the chatbot. There were a number of limitations, including the chatbot not having enough interactions to learn from before going live. And also, the chatbot patterns usages are neither recorded nor pre-defined in advance to prepare the appropriate responses.

Another chatbot implementation to enhance the LMS experience was proposed by [23]. This model classifies the main keywords that could be asked by students using R programming language, and this classification is then used in an Artificial Intelligent Markup language (AIML) script as a query. If this query was unsuccessful, it would run against SQL

lite. If neither AIML nor SQL lite worked, then the student query will be transferred to a human agent to take over and answer the query. Although the implementation of AIML scripting language is easy and also free to use as a scripting language, this model is a rule-based model and is less tolerant to changes in users' input and, thus, harder to capture the user intent.

Another study for developing a chatbot for university inquiries was put forward by [24]. This study discussed the development of a deep-learning based chatbot using RASA framework. RASA has many connectors to be used in integrating it with communication platforms. One of them is for Facebook. This chatbot is integrated with FB as the majority population is using FB as their main social media channel. This chatbot uses Long Short-Term Memory (LSTM), which is a recurrent neural network architecture that is used in deep learning. This architecture is included in RASA framework. Although the chatbot performed well in terms of intent classification and provided the appropriate replays, there was a platform limitation as they had to perform platform-specific steps to run the chatbot on Facebook, which can result in some limitations to the interaction with the chatbot.

A chatbot for instantly answering students' questions to reduce teacher's workload was proposed in [25]. It supports multiple common social platforms, including Telegram, Facebook Messenger, and Line. The chatbot can reply to commands and natural language questions. Once the instructors transfer the course-related data to an internet database, the chatbot can reply to questions about the course materials and logistics (e.g., course plan). It also supports student login to provide profile-based answers such as the schedule of student registered courses.

B. Chatbots in other Fields

Chatbots also have many usages in other fields besides education. Some of those applications are in healthcare, such as self-diagnoses based on symptoms, using chatbots as a communication means in e-commerce websites, providing account data and paying bills in banking, etc. Below are some of the related works of chatbots.

A text-to-text chatbot engages patient's medical issues were proposed by [26]. It's a medical chatbot that diagnoses diseases using AI. This chatbot was built to reduce medical costs and improve patient's accessibility to medical knowledge. In this chatbot, a series of questions about the patient's symptoms are asked to give suggestions that help in clarifying the disease. The accurate disease is found based on the user reply to those series of questions, and in case of major diseases, a doctor is suggested to be consulted. The patient's past responses are recorded, and in order to reach an accurate diagnosis, the patient is asked more specific questions. There are three main components of the system, which are (1) user validation and symptoms extraction from the conversation, (2) mapping of extracted, potentially ambiguous symptoms to their corresponding database codes, and (3) personalized diagnosis and referring the patient to a specialized doctor if required. The sole focus of this system is extracting symptoms by analyzing natural language using NLG components, which in term makes it easier and less technical for the end-user.

Another example of chatbot usage in e-commerce to support customers in their website journey is called “SuperAgent” [27]. This chatbot scrapes public e-commerce websites' content of products description, user questions and answers, and product reviews and feeds them to its knowledge base. It uses NLP techniques to understand users' text and machine learning techniques to predict responses to it, including opinion mining for product reviews, fact QA for product information, and FAQ search for customer reviews and chit-chat for greetings and goodbyes.

ChatPy is one of the chatbot implementations in the wholesale business [28]. “Mundirepuestos” is a wholesaler

automotive spare company. This company is an SME company that started operating in 1992 that specialized in the distribution and sales of Volkswagen, Skoda, and Audi automotive parts. ChatPy is a conversational agent built mainly using a tool called Dialogflow. This tool makes use of intents, actions with parameters, entities, voice-to-text, and text-to-speech with automatic learning. A major reason for choosing this tool was its compatibility with the most known messaging platforms. A summary of chatbot related works in different fields is shown in Table I.

TABLE I. SUMMARY OF RELATED RESEARCH

Title	Field	Method	Limitations
Developing a Chatbot for College Student Programme Advisement [22]	Education	Intent detection, conversational management routines, dialogue design, course information management.	Minimal interactions to learn from before going live. Patterns usages are neither recorded nor pre-defined in advance to prepare the appropriate responses.
Enhancing LMS Experience through AIML Base and Retrieval Base Chatbot using R Language [23]	Education	Artificial Intelligent Markup language (AIML) script as a query, SQLite.	Using a rule-based model only.
Developing FB Chatbot Based on Deep Learning Using RASA Framework for University Enquiries [24]	Education	Long Short Term Memory (LSTM) a recurrent neural network architecture included in RASA framework.	Facebook-specific implementation to run the chatbot which limits customization.
Using a Multiplatform Chatbot as an Online Tutor in a University Course [25]	Education	DialogFlow's ML engine and knowledge base.	Low number of participants for evaluating the model.
A Self-Diagnosis Medical Chatbot Using Artificial Intelligence [26]	Healthcare	Symptom recognition, String Searching Algorithm.	No fallback policy for unknown symptoms.
SuperAgent: A Customer Service Chatbot for E-commerce Websites [27]	E-commerce	Opinion mining for product reviews, fact QA for product information, and FAQ search for customer reviews and chit-chat conversations.	No intent detection.
ChatPy: Conversational agent for SMEs: A case study [28]	Business	DialogFlow's ML engine and knowledge base.	Facebook-specific implementation to run the chatbot which limits customization.

To avoid issues in [22], the system needs to be internally deployed and used by diverse students' backgrounds while recording their usage patterns and interactions which is recorded by default in Rasa framework. Rule-based chatbots like the one presented in study [23] cannot learn which is not the case for Rasa as it has interactive learning capabilities which allows it to learn and refrain from making the same mistake in the future. Unlike the cases presented in study [24] and [28], this work is going to use Rasa API to communicate with the chatbot which remove the platform specific limitations and allows for more customizations. There needs to be a fit number of participants to fairly evaluate the chatbot to overcome the limitation in study [25]. As opposed to the case presented in study [26], Rasa provides a fallback policy that can be triggered when the prediction of the action to be taken is below a specified threshold; this fallback could be used to ask the user to rephrase or show some buttons for the user to choose from. Knowing the user intent can help greatly in providing the right answer to the user question and also helps in performing actions based on the user intent; Rasa uses deep learning embeddings to detect user intention which is not the case in study [27].

III. SYSTEM DESIGN

Fig. 1 below gives an overview of Rasa open-source architecture that consists of two main components which are the Rasa NLU and Dialog management (Core). Rasa NLU is

responsible for predicting intents, extracting entities and retrieving responses. It uses the saved model in the filesystem. The Core component is responsible for choosing the appropriate next action with regards to the conversation context and uses Tracker store to store the conversation states, messages and metadata.

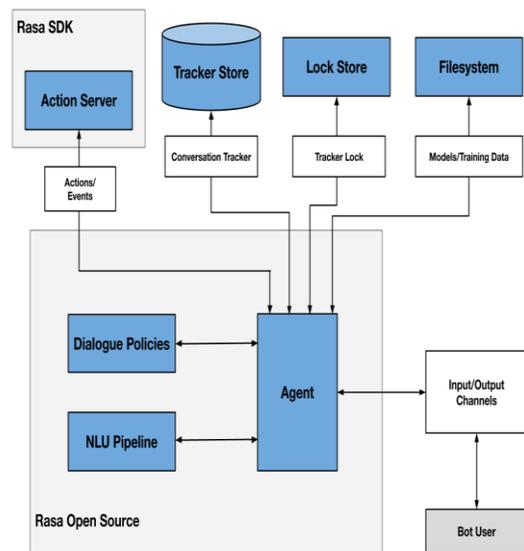


Fig. 1. System architecture overview.

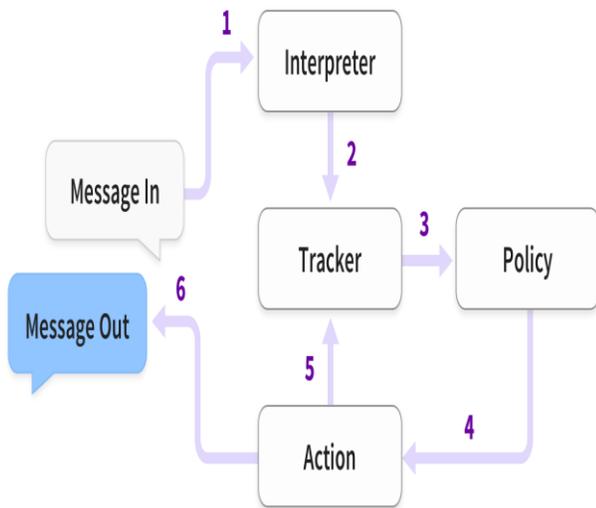


Fig. 2. Rasa architecture.

Rasa ensures that messages are being processed in the right order using the lock store. Actions are running on the Action Server and executed when called by the Core component. Fig. 2 shows the message flow and how Rasa architecture works.

The user first types in a message; this message is then passed to the interpreter in which NLU is used to extract user intention, intent for short, and any entities contained in the text. The conversation state is then saved in a Tracker object, and an event is created, i.e., new message arrival. The state is then received by the policy, and the next action is chosen by the policy to be taken. The action is also logged into the Tracker and then implemented, which could be a response that is based on an external API call or a simple text response that is sent back to the user.

A. Data Collection

To gather the required data for training the chatbot to answer most frequent questions asked by students, the LMS Blackboard team admin was contacted to provide this data. This data is in the form of a Word document format and will be used to manually generate the training data and build chatbot stories to train the chatbot. It contains 17 of the frequently asked questions by the Blackboard users. Examples are shown in Appendix 1.

There are two datasets to be built, one for the NLU model which contains examples for each user intent along with labeled entities. The data for this project are in MSA, Modern Standard Arabic and some common English words. The second dataset is for the Core model or the Dialog Management model, it has all the possible flows for the conversation (intents with their corresponding actions). The latter might not be needed when using mapping policy which maps each intent to an action or a template. The dataset to serve the current need (i.e., Blackboard system) was requested from Blackboard support staff to build the initial dataset, and get a sense of the frequently asked questions by KSU Blackboard student users. This data will be then increased by synthesizing text that could

be asked by the chatbot users to increase the chances of understanding students' questions about the Blackboard system. For chatbots systems, the datasets should be continuously updated after deployment for continuous enhancement. The data formats for both NLU and Core are written in a user-friendly format to make it easier to build, revise and edit. For the NLU model, examples for each intent along with labeled entities are created. There are two available formats for building the dataset, either as a json or a markdown format. Markdown formats are the most used as it can be rendered by most text editors.

B. Building NLU Corpus

The main goal of building this corpus is to make the chatbot see many examples of what the user might say regarding a specific intention of the user. There are two available formats for building the dataset, either as a json or a markdown format. Markdown formats are the most used one as it can be rendered by most text editors. Below is an example of markdown NLU dataset record.

```
1. ## intent: FAQ_login_issue
2. -(system)[نظام إدارة التعلم] لا أستطيع الدخول على
```

The other format is the json format; it's not sensitive for whitespaces and better in exchanging data among applications. The actual NLU corpus can be found in Appendix 2.

```
1. {
2. "text": " لا أستطيع الدخول على نظام إدارة التعلم",
3. "intent": "FAQ_login_issue",
4. "entities": [{
5. "start": 19,
6. "end": 33,
7. "value": "نظام إدارة التعلم",
8. "entity": "system"
9. }]}
```

C. Building Stories

Stories are a type of data that is used in order to teach the chatbot the possible messaging flow with user. Markdowns are used to specify the conversation paths i.e., stories. Below is an example for Dialog management model training data.

```
1. ## story_84854865
2. * greet
3. - utter_ask_howcanhelp
4. - utter_ask_issue
5. * inform("system": "نظام إدارة التعلم")
6. - action_replay_issue
```

The naming convention for stories is to start with two hashes, followed by the story name. Actions are events that start with a dash. The actual Core model corpus can be found in Appendix 3.

D. Implementation

Rasa environment requires a list of hard and software requirements for running Rasa on Docker. Although there are minimum hardware requirements on Rasa official website, the hardware requirements depend on the size of the model and training data as the training time and the size of the NLU data are positively correlated. Those requirements need to be met to

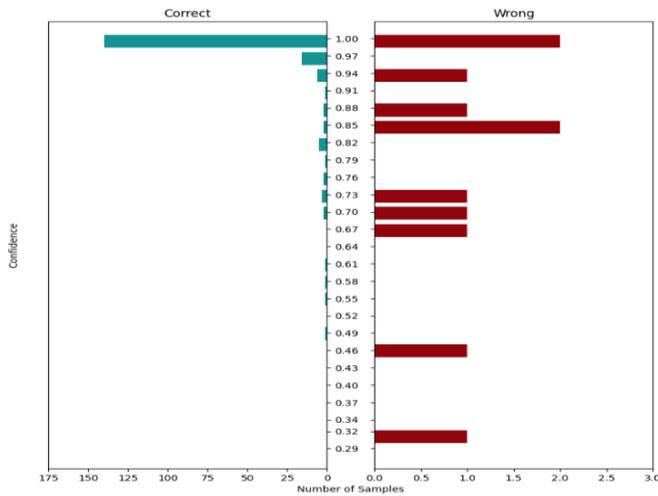


Fig. 5. NLU intent confidence histogram.

From Table III, we can see that all actions were predicted correctly with a value near to 1 for F1-score, precision, and accuracy. The reason for such high results is that the dialog management model is classifying actions based on the results of the intent classifier. If there are no errors in predicting the intention of the user, the prediction of the next action becomes easier and hence, result in a high hit rate.

TABLE III. DIALOG MANAGEMENT RESULTS

Metric	Score
Correct	335/338
F1-score	0.991
Precision	0.992
Accuracy	0.991

As mentioned earlier, we will try to optimize the NLU model by adding more examples and removing examples that might mislead the model into falsely predicting intents. We will also change some of the NLU model configurations to see if those changes yield better results (Table IV).

TABLE IV. PRE-TUNING VS. POST-TUNING METRICS

Metric	Pre-Tuning Score	Post-Tuning Score
Train Accuracy	1	1
Train F1-score	1	1
Train Precision	1	1
Test Accuracy	0.924	0.934
Test F1-score	0.911	0.925
Test Precision	0.922	0.925

Although those are minor changes, they do have an effect and it means that it's possible to further optimize the model by adding more data and tuning the model parameters to find the ones that best fit the data. The average reported accuracy in similar case studies mentioned in [22], [24], [27] is near 90% as opposed to our results which is slightly higher.

V. CONCLUSION

This work intended to develop a chatbot engine that helps with frequently asked questions about Blackboard system, which could be embedded into Blackboard website. It contains a machine-learning model trained on Arabic datasets. The engine accepts both Arabic textual content as well as English textual content if needed; for commonly used English terminologies. The interactions with the chatbot, as well as the users' evaluations, are stored and used for optimizing the chatbot model to improve future interactions. Chatbot systems development entails many challenges in terms of preparing the training dataset in a way that incorporate as much as possible of users' inquiries without confusion, preprocessing it before feeding it to the NLU model to try to normalize the data and remove unnecessary words and symbols that could confuse the model, and deploy and maintain the model to be used. Rasa framework was chosen to as the main tool for developing the Blackboard chatbot.

The actual chatbot implementation started by preparing the datasets required for Rasa NLU and Core models. The dataset is designed to account for as many as possible of KSU Blackboard- related inquires to provide the appropriate answers and reduce the workload of Blackboard system support staff. When the data was ready, the model training and tuning began along with a number of experimentations to find the best model pipelines that fits the data. The chatbot is built using a combination of tools such as Python for programming, YAML as the markup language.

For future work, the chatbot should be deployed using Docker and Docker-compose for running the chatbot service. The chatbot can also be deployed in a distributed cluster either on cloud or on- premise to handle the workload and make the chatbot system scalable.

REFERENCES

- [1] T. Bocklisch, J. Faulkner, N. Pawlowski, and A. Nichol, "Rasa: Open source language understanding and dialogue management," arXiv. 2017.
- [2] M. L. ATKINSON, "Voice is the Thing in 2018 - Order automation for restaurants - Orderscape," 2018. <https://orderscape.com/blog/voice-is-the-thing-2018-order-automation-restaurants/> (accessed Oct. 28, 2020).
- [3] Mindbrowser, "Global Chatbot Trends Report – 2020 - Mindbrowser," 2017. <https://www.mindbrowser.com/chatbot-market-survey-2017/> (accessed Oct. 28, 2020).
- [4] Chatbots Magazine, "Chatbot Report 2019: Global Trends and Analysis | by BRAIN [BRN.AI] CODE FOR EQUITY | Chatbots Magazine," 2019. <https://chatbotmagazine.com/chatbot-report-2019-global-trends-and-analysis-a487afec05b> (accessed Oct. 28, 2020).
- [5] Expert System Team, "Chatbot: What is a Chatbot? Why are Chatbots Important? - Expert.ai | Expert.ai," 2020. <https://www.expert.ai/blog/chatbot/> (accessed Dec. 05, 2020).
- [6] A. M. Turing, "Computing machinery and intelligence," in Machine Intelligence: Perspectives on the Computational Model, 2012.
- [7] A. : Peters, F. Promoteur, and L. Wehenkel, "Master thesis : Design and implementation of a chatbot in the context of customer support." Accessed: Dec. 06, 2020. [Online]. Available: <http://matheo.uliege.be>.
- [8] "(PDF) A Literature Review On Chatbots In Healthcare Domain." https://www.researchgate.net/publication/334836867_A_Literature_Review_On_Chatbots_In_Healthcare_Domain (accessed Dec. 06, 2020).
- [9] L. Batista and L. Alexandre, "Text Pre-processing for Lossless Compression," 2008, doi: 10.1109/dcc.2008.78.

[10] Singh and V. Gupta, "Text stemming: Approaches, applications, and challenges," ACM Comput. Surv., 2016, doi: 10.1145/2975608.

[11] A. Oussous, F. Z. Benjelloun, A. A. Lahcen, and S. Belfkih, "ASA: A framework for Arabic sentiment analysis," J. Inf. Sci., 2020, doi: 10.1177/0165551519849516.

[12] N. Lee, K. Kim, and T. Yoon, "Implementation of robot journalism by programming custombot using tokenization and custom tagging," 2017, doi: 10.23919/ICACT.2017.7890154.

[13] T. Jiang, H. Yu, and Y. Jam, "Tibetan word segmentation system based on conditional random fields," in ICSESS 2011 - Proceedings: 2011 IEEE 2nd International Conference on Software Engineering and Service Science, 2011, pp. 446-448, doi: 10.1109/ICSESS.2011.5982349.

[14] J. R. Bellegarda, "Part-of-speech tagging by latent analogy," IEEE J. Sel. Top. Signal Process., 2010, doi: 10.1109/JSTSP.2010.2075970.

[15] L. Yang, M. Zhang, Y. Liu, M. Sun, N. Yu, and G. Fu, "Joint POS Tagging and Dependence Parsing with Transition-Based Neural Networks," IEEE/ACM Trans. Audio Speech Lang. Process., 2018, doi: 10.1109/TASLP.2017.2788181.

[16] B. Chen and D. Ji, "Chinese semantic parsing based on dependency graph and feature structure," 2011, doi: 10.1109/EMEIT.2011.6023005.

[17] Z. Li, M. Zhang, W. Che, T. Liu, and W. Chen, "Joint optimization for Chinese POS tagging and dependency parsing," IEEE Trans. Audio, Speech Lang. Process., 2014, doi: 10.1109/TASLP.2013.2288081.

[18] L. Gao and H. Chen, "An automatic extraction method based on synonym dictionary for web reptile question and answer," 2018, doi: 10.1109/ICIEA.2018.8397745.

[19] S. Qin, J. Song, P. Zhang, and Y. Tan, "Feature selection for text classification based on part of speech filter and synonym merge," 2016, doi: 10.1109/FSKD.2015.7382024.

[20] N. Chetty, K. S. Vaisla, and N. Patil, "An Improved Method for Disease Prediction Using Fuzzy Approach," 2015, doi: 10.1109/ICACCE.2015.67.

[21] B. Xu, Y. Ma, and H. Lin, "A hybrid deep neural network model for query intent classification," J. Intell. Fuzzy Syst., 2019, doi: 10.3233/JIFS-182682.

[22] C. Chun Ho, H. L. Lee, W. K. Lo, and K. F. A. Lui, "Developing a Chatbot for College Student Programme Advisement," 2018, doi: 10.1109/ISET.2018.00021.

[23] V. K. Shukla and A. Verma, "Enhancing LMS Experience through AIML Base and Retrieval Base Chatbot using R Language," 2019, doi: 10.1109/ICACTM.2019.8776684.

[24] Y. Windiatmoko, A. F. Hidayatullah, and R. Rahmadi, "Developing FB Chatbot Based on Deep Learning Using RASA Framework for University Enquiries," Sep. 2020, Accessed: Nov. 22, 2020. [Online]. Available: <http://arxiv.org/abs/2009.12341>.

[25] L. K. Lee, Y. C. Fung, Y. W. Pun, K. K. Wong, M. T. Y. Yu, and N. I. Wu, "Using a Multiplatform Chatbot as an Online Tutor in a University Course," 2020, doi: 10.1109/ISET49818.2020.00021.

[26] S. Divya, V. Indumathi, S. Ishwarya, M. Priyasankari, and S. Kalpana Devi, "A Self- Diagnosis Medical Chatbot Using Artificial Intelligence," J. Web Dev. Web Des., 2018.

[27] L. Cui, S. Huang, F. Wei, C. Tan, C. Duan, and M. Zhou, "Superagent: A customer service chatbot for E-commerce websites," 2017, doi: 10.18653/v1/P17-4017.

[28] P. Perez, F. De-La-Cruz, X. Guerron, G. Conrado, P. Quiroz-Palma, and W. Molina, "ChatPy: Conversational agent for SMEs: A case study," in Iberian Conference on Information Systems and Technologies, CISTI, Jun. 2019, vol. 2019-June, doi: 10.23919/CISTI.2019.8760624.

[29] "Docker Compose Installation." <https://rasa.com/docs/rasa-x/installation-and-setup/install/docker-compose/#docker-compose-manual-install> (accessed Apr. 02, 2021).

[30] "What is Python? Executive Summary | Python.org." <https://www.python.org/doc/essays/blurb/> (accessed Apr. 02, 2021).

APPENDICES

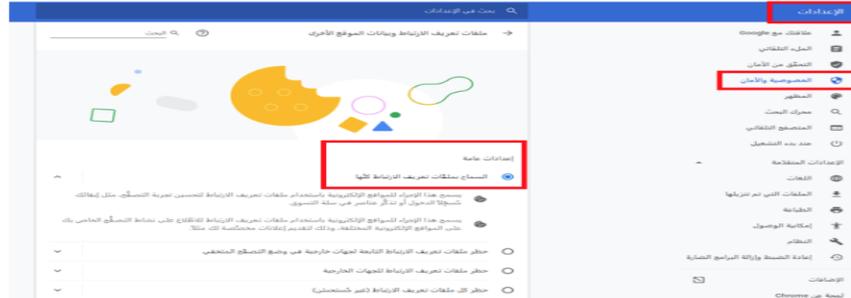
Appendix 1: Raw Data: Below is a snippet of the raw data used in building the chatbot. For the full file please download it through this link: <https://drive.google.com/drive/folders/1zKxniGzjqYUCYAcHgUM1I8o-Ma-sg8Ag?usp=sharing>.

أسئلة الطلاب الشائعة لنظام إدارة التعلم LMS

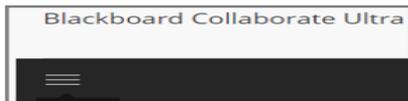
س/ يوجد لدي مشكلة بالصوت داخل جلسة الفصل الافتراضي؟
هم بتفعيل خيار الوصول إلى الميكروفون من المتصفح المستخدم ، وذلك من خلال النقر على علامة الففل بجانب اسم الموقع بالأعلى ثم إعدادات الموقع ثم اختر السماح للميكروفون

س/ عند الدخول على نظام إدارة التعلم يوجد لدي مشكلة بالصوت؟
لا تستخدم المتصفح من خلال الحوالم، بل عليك أن تقوم بتحميل تطبيقين على حوالم أو الجهاز اللوحي وهما:
Blackboard -1
Collaborate -2

س/ عند الضغط على قائمة الفصول الافتراضية يظهر شاشة سوداء ولا تظهر لي الجلسات؟
اذهب إلى علامة الففل lock بجانب اسم الموقع بالأعلى ثم إعدادات الموقع ثم الخصوصية والأمان ثم فعل السماح بملفات تعريف الارتباط كلها.



س/ كيف أصل إلى تسجيلات المحاضرات السابقة؟
اذهب إلى الفصول الافتراضية ثم علامة الإعدادات ثم انقر على التسجيلات وانبع الخطوات.



Appendix 2: NLU Corpus: Below is a snippet of the NLU data used in building the chatbot intent classifier. For the full file please download it through this link: <https://drive.google.com/drive/folders/1TN3yluifFYDXm7FNwdNn-AbthC70hWYg?usp=sharing>.

```
- intent: greet
  examples: |
    - السلام عليكم
    - السلام عليكم ورحمة الله وبركاته
    - هلا اخوي
    - الله يحبك
    - اهلا وسهلا
    - حياك الله ياخي
    - سلاام عليكم
    - سلام
    - هلا
    - أهلا
    - صباح الخير
    - مساء الخير
    - صباحك الله بالخير
    - مساءك الله بالخير
    - هلا اخوي
    - ايدا
    - هلا والله
    - صباحالخير
    - مساءالخير
    - يسعد مسالكك
    - يسعد صباحك
- intent: FAQ_how_to_see_screen_recordings
  examples: |
    - كيف اشوف تسجيل المحاضرات
    - كيف اشوف المحاضرات السابقة
    - وين القى المحاضرات السابقة
    - وين القى المحاضرات الماضية
    - كيف اشوف الجلسات المسجلة
    - الدكتور كان يسجل المحاضره كيف
    - وين القى الجلسات المسجلة
    - وين احصل تسجيل المحاضرات
    - ماعرفت اطلع تسجيل المحاضرات
```

Appendix 3: Core Model Corpus: Below is a snippet of the Core (Dialog) data used in building the chatbot Dialog model. For the full file please download it through this link: <https://drive.google.com/drive/folders/1uzF31NWfO6sAv5VzIAkcrUD6hIv0c-Y8?usp=sharing>.

```
- story: FAQ_in_lms_sound_issue path
  steps:
    - intent: greet
    - action: utter_greet
    - action: action_listen
    - intent: FAQ_in_lms_sound_issue
    - action: utter_FAQ_in_lms_sound_issue
    - action: action_listen
    - action: utter_goodbye
- story: FAQ_how_to_change_password path
  steps:
    - intent: greet
    - action: utter_greet
    - action: action_listen
    - intent: FAQ_how_to_change_password
    - action: utter_FAQ_how_to_change_password
    - action: action_listen
    - action: utter_goodbye
- story: FAQ_how_to_see_alerts_in_lms path
  steps:
    - intent: greet
    - action: utter_greet
    - action: action_listen
    - intent: FAQ_how_to_see_alerts_in_lms
    - action: utter_FAQ_how_to_see_alerts_in_lms
    - action: action_listen
    - action: utter_goodbye
```

Appendix 4: Domain File: Below is a snippet of the Domain file used in building the chatbot. For the full file please download it through this link: <https://drive.google.com/drive/folders/1GPqHIRBFg9slW5ddSGUZhQqxIIERYzvK?usp=sharing>.

```
intents:
- greet:
  use_entities: true
- goodbye:
  use_entities: true
- at_fms:
  use_entities: true
- help:
  use_entities: true
- FAQ_in_lms_sound_issue:
  use_entities: true
- FAQ_no_session_black_screen:
  use_entities: true
- FAQ_how_to_see_green_recordings:
  use_entities: true
- FAQ_changes_in_edu_gate_didnt_appear_in_lms:
  use_entities: true
- FAQ_courses_in_lms_are_not_sorted:
  use_entities: true
- FAQ_how_to_change_password:
  use_entities: true
- FAQ_cant_login_to_lms:
  use_entities: true
- FAQ_how_to_see_alerts_in_lms:
  use_entities: true
- FAQ_how_to_resubmit_document:
  use_entities: true
- FAQ_some_options_doesnt_appear:
  use_entities: true
- FAQ_courses_doesnt_appear_in_mobile:
  use_entities: true
- FAQ_saudi_digital_library_login_issue:
  use_entities: true
- FAQ_virtual_sessions_doesnt_appear_in_side_panel:
  use_entities: true
- FAQ_im_the_only_one_in_session:
  use_entities: true
- FAQ_my_submission_is_pending:
  use_entities: true
- FAQ_submit_button_is_not_working:
  use_entities: true

- bot_challenge:
  use_entities: true
entities: []
slots: {}
responses:
utter_greet:
- text: أهلا عزيزي الطالب، كيف أقدر أخدمك؟
utter_did_that_help:
- text: هل تمت خدمتك بشكل صحيح؟
utter_goodbye:
- text: إلى اللقاء!
utter_iamabot:
- text: من مطلاب جامعة الملك سعود LMS أنا بوت مطور لخدمة مستخدمي
utter_FAQ_in_lms_sound_issue:
- image: https://i.ibb.co/QnffkXz/Screen-Shot-2021-04-05-at-9-48-57-AM.png
text:
  قم بتحميل خيار الوصول إلى الميكروفون من المتصفح المستخدم ، وذلك من خلال النقر على علامة القفل بجانب اسم الموقع بالأعلى ثم إعدادات الموقع ثم
  اختر السماح للميكروفون
  لا تستخدم المتصفح من خلال الجوال، بل عليك أن تقوم بتحميل تطبيقين على جوالك أو الجهاز اللوحي وهما
  Collaborate
utter_FAQ_no_session_black_screen:
- image: https://i.ibb.co/1rpgqxy/no-session.png
text:
  بجانب اسم الموقع بالأعلى ثم إعدادات الموقع ثم الخصوصية والأمان ثم فعل السماح بملفات تعريف الارتباط كلها lock اذهب إلى علامة القفل
  بجانب اسم الموقع بالأعلى ثم إعدادات الموقع ثم الخصوصية والأمان ثم فعل السماح بملفات تعريف الارتباط كلها lock
```

Appendix 5: Test Stories: Below is a snippet of the Test Stories used for testing the chatbot. For the full file please download it through this link: https://drive.google.com/drive/folders/1WtU-_ph30Kkjf2ycC2tpVRx1843Y0puh?usp=sharing.

```
stories:
- story: FAQ_in_lms_sound_issue test story 1
  steps:
  - user: |
    أهلا
    intent: greet
  - action: utter_greet
  - action: action_listen
  - user: |
    عندي مشكلة فالصوت ما ادري وش السبب
    intent: FAQ_in_lms_sound_issue
  - action: utter_FAQ_in_lms_sound_issue
  - action: action_listen
  - user: |
    شكرا
    intent: goodbye
  - action: utter_goodbye

- story: FAQ_cant_login_to_lms test story 2
  steps:
  - user: |
    سلام عليكم
    intent: greet
  - action: utter_greet
  - user: |
    lms ما اقدر ادخل على ال
    intent: FAQ_cant_login_to_lms
  - action: utter_FAQ_cant_login_to_lms
  - action: action_listen
  - user: |
    يعطيك الف عافيه
    intent: goodbye
  - action: utter_goodbye
```