A Chatbot Approach to Automating FAQ Responses in an Undergraduate Course Domain

Graciane X. Nobre[1], Gustavo Moraes[2], Wellington Franco[3], Leonardo O. Moreira[1]

[1]Virtual University Institute (UFC Virtual)
Federal University of Ceará (UFC) – Fortaleza, CE – Brazil

[2]Department of Computing
Federal University of Ceará (UFC) – Fortaleza, CE – Brazil

[3]Campus of the Federal University of Ceará in Crateús
Federal University of Ceará (UFC) – Fortaleza, CE – Brazil

gracianexavier@alu.ufc.br, gustavo.moraes@alu.ufc.br,
wellington@crateus.ufc.br, leoomoreira@virtual.ufc.br

Abstract. *Chatbots are web or mobile interfaces that allow the user to ask questions and retrieve information using natural language. Chatbots can be very useful as when used to automate some conversations between users. They can be used in several domains including education services. This work aims to save time and energy of an undergraduate course administrative staff by automating the answers to frequent questions. In this paper, we present the development process of the chatbot approach to automating FAQ responses in an undergraduate course domain, used technologies, the system architecture, a prototype and the chatbot evaluation.*

1. Introduction

Computers are used in many fields such as health, finance, e-commerce, education, among many others. From your smartphone in your pocket to your work laptop, they have become ubiquitous, that is, they can be used anytime and anywhere. In order to facilitate human communication with computers, conversational interfaces were developed. Conversational User Interfaces (CUI) or Conversational Agents are platforms that perform conversations with humans and *"enable people to interact with smart devices using spoken language in a natural way"* [McTear et al. 2016].

Two types of conversational agents are voice assistants and chatbots [Valtolina et al. 2018]. The former allows the user to use voice commands to interact with the interface; some commonly used voice assistants are Siri by Apple[1], Google Voice Assistant[2], among others. The latter, chatbots, are web or mobile interfaces that allow the user to ask questions and retrieve information using natural language. There are several types of chatbots. Some are developed for business, educational, information retrieval, or e-commerce purposes [McTear et al. 2016]. For communication purposes, chatbots can be very useful when used to automate some conversations between users. They are used

[1]Siri by Apple. Available at: https://www.apple.com/siri/. Accessed: August 2nd, 2019

[2]Google Voice Assistant. Available at: https://assistant.google.com/. Accessed: August 2nd, 2019

as customer service assistants, instant translators such as Instant Translator[3] on Facebook Messenger, moderator bots in messaging apps such as Telegram, among other functionalities.

Chatbots can not only retrieve information from databases, but they can also use artificial intelligence to make suggestions according to the user's input in the conversation. As an example of this, "And Chill"[4], is a chatbot developed for Facebook Messenger. It recommends movies based on preferences that the user sends to the application. These conversational agents are part of an area called Natural Language Understanding (NLU). According to Ovchinnikova (2012), the goal of an NLU system is to interpret an input text fragment. It is the process of a system to turn natural language into a computational representation. The author describes NLU as a subfield of Natural Language Processing.

As for administrative services in universities, chatbots can be very useful as well. Many universities have adopted the use of chatbots to respond to questions from students in order to improve the efficiency of administrative officers work and to enhance the experience of students [Lee et al. 2019].

The university administrative staff receives several e-mails, phone calls and messages everyday. Responding to every question, which are usually repetitive, may impact on the work of these assistants. By automating a Frequently Asked Questions (FAQ), the administrative officers are able to work on other priority tasks instead of responding to repetitive questions. Lee et al. (2019) examined the efficiency of chatbots in university administrative services, whether it would reduce the staff workload or not. Their results indicated that the chatbot reduced the amount of workload at a university administrative office.

In addition to that, students who have recently entered university do not know how the course works or may not know where to find the answer, therefore they may feel lost. As Dibitonto (2018) states, a chatbot can help students find information easier, in a natural way, proving it to be of great help during their university life. By using a chatbot, students do not need to browse websites looking for Frequently Asked Questions pages or call the administrative staff to ask questions. The use of the chatbot may facilitate their access to the information they need.

In order to support the administrative staff of an undergraduate course and help students find information faster, we developed a FAQ chatbot to respond to the course related questions such as enrollment, professors contact information, graduation information, etc. The administrative assistants provided us with some frequent questions and their respective answers. The FAQ was added to the chatbot dataset to automate it. The objective is to save time and energy of the administrative staff by automating the answers to frequent questions.

The paper is organized as follows: Section 2 describes the definition of a conversational agent and explains some approaches used in the literature. Then we define DialogFlow by outlining some of its features. Section 3 presents related works and the different techniques used to develop a chatbot. Section 4 demonstrates our approach with

[3]Instant Translator. Available at: `https://botlist.co/bots/instant-translator`. Accessed: July 12th, 2019.

[4]And Chill. Available at: `http://www.andchill.io/`. Accessed: July 12th, 2019.

description of the system architecture, implementation aspects, and Proof of Concept (PoC). Finally, in Section 6, we conclude the paper and present future work.

2. Background

The idea of creating smart conversational agents started in the 1950s with Alan Turing paper [Turing 1950] in which he discusses the question "*Can machines think?*". He proposed the "*Imitation Game*", currently known as the Turing Test. A system would be considered intelligent if it could pretend to be human in a dialogue with a human interlocutor.

Turing's work opened the possibility of developing "dialogue machines" [Barros and Azevedo 2016], later known as *Chatterbots* or *Chatbots* [Mauldin 1994]. Such systems, also called *Conversational Agents* are designed to interpret the user's question and offer direct answers, seeking to maintain the illusion that the user is talking to another human being.

The following subsections describe important concepts we use in this work.

2.1. Proof of Concept (PoC)

According to Neto (2018), a Proof of Concept (PoC) is a research practice which serves as an instrument of study to understand, validate, and analyze objects in a given area of knowledge. A PoC is used to prove a concept through a practical model, with the aim of promoting knowledge of the objects under study [Neto et al. 2018].

2.2. Conversational Agent

A conversational agent or chatbot is a computer program that attempts to simulate a human being in conversation with someone [Barros and Azevedo 2016]. The goal is to answer a set of questions so that the user has the impression they were talking to another person and not to a computer program. After receiving the user questions in a natural language format, the system searches for an example sentence in a knowledge base and then returns an answer in an attempt to mimic human behavior. These kinds of robots are not only built to mimic human conversation, and entertain users, they can also be used as tutors in distance education, customer service, information retrieval, business, and e-commerce [Shawar and Atwell 2007].

In the literature, there are references to three approaches of chatbots [Mauldin 1994]. Such approaches use distinct techniques regarding their implementation [Mitkov 2003]. The first implementation uses pattern matching and grammatical rules, represented by the software Eliza [Weizenbaum 1966]. Although this approach works quite well for a small knowledge base, it is impracticable for larger systems due to the large number of rules necessary to cover all sentences.

The second approach uses a neural network model to generate answers from input patterns, implemented in the software Julia [Mauldin 1994]. However, the computational cost to train the network is very high when the knowledge base has a great number of sentences. The most popular approach is based on a markup language called Artificial Intelligence Markup Language (AIML) [Wallace 2003] and it is necessary for a parser to read the user's input and to search for an answer iterating over the file.

Being an XML file, it is very difficult to read and when the file is big, the management can become very complicated, mainly when adding, removing or editing a sentence. Moreover, if the tags are not in the right format or in the right place, it will not work at all. Either way, the building of chatbots can be a complex process. It has a great operational cost, because involves a great number of processes, variation of the application's domain and problems of natural language processing. To avoid these drawbacks, we built a system on DialogFlow, powered by Google.

2.3. DialogFlow

DialogFlow[5], previously known as api.ai, is a development cloud-based platform based on Natural Language Understanding owned and maintained by Google. It incorporates Google's machine learning algorithms, supports different programming languages and has different built-in integration with other chatbot-based platforms such as Telegram, Facebook Messenger, Twitter, Skype, among other apps. The platform has a console which is used to create, build, manage and test the agents. An agent is "a virtual agent that handles conversations with your end-users".

In the console, the developer (any person with previous or zero background in programming) can create agents, intents which represent the purpose of a user input, entities to extract useful data from user input, control conversation paths with contexts, add events that are triggered by occurrences outside of the conversation, integrate with other conversational platforms, implement fulfillment to connect the service when using integration, analyze agent performance, and test the agent via the simulator on the console. For the development of our chatbot, we created intents and some entities which will be explained in Section 4.

3. Related Work

In this section a search was made for works that are solutions for FAQ using chatbot and approaches based on dialog systems. Shawar et al. (2005) adapted a chatbot-training program to the FAQ in the School of Computing at University of Leeds, England, producing the FAQ chat system. It retrieves information by matching keywords pattern without using any linguistic tool and by using the most significant word approach. The system replies to queries using a FAQ database from the university.

Ranoliya et al. (2017) implemented an interactive chatbot for university-related FAQs to meet the academic needs of the visitors. The application answers the queries from students or parents about college admission among other questions related to Manipal University in India. The chatbot is based on Artificial Intelligence Markup Language (AIML) language which is an XML based markup language meant to create artificial intelligent applications and uses the pattern matching technique.

Bala et al. (2018) developed a chatbot for a college management system using artificial language algorithms. Their system is a web application which provides answers to the analyzed queries of the user. The user register to the system and has access to pages in which they can chat by asking questions related to college activities.

[5]DialogFlow Documentation. Available at: `https://dialogflow.com/docs`. Accessed at: October 10th, 2019

Table 1. The Descriptive Taxonomy For The Six Nlu Platform Examined

Platform	*Usability*	*Languages*	*Programming Languages*	*Pre-build Entities*	*Pre-build Intents*	*Default Fallback Intent*	*Automatic Context*	*Composition Mode*	*Online Integration*	*Webhook/ SDK Availability*	*All-in Platform*	*Linkable Intents*	*Price*
Dialog-Flow	High	15, from English to Chinese	11, from Java to Ruby	60, from addresses to colors	34, from small talks to currency converters	Yes	Yes	Form-based	14, from Telegram to Alexa	Webhook and SKDs	Yes	Yes	Free
wit.ai	Medium	50, from Albanian to Ukrainian	3: Node.js, Python, and Ruby	22, from location to email	Zero	No	No	Form-based	Zero	SDK	No	No	Free, contact heavy usage
LUIS	Medium	10, from English to Chinese	4: Android, Python, Node.js, and C#	13, from numbers to geography	20, from calendar to fitness	Yes	No	Form-based	Zero	Webhook and SDK	No, other services are in Azure	No, other services are in Azure	Free up to 10k requests per month
Watson Conversation	High	12, from English to Chinese	6, from Node.js to Java	7, from time to person	Zero	Yes	Yes	Form and block-based	Zero	SDK	No, other services are in Bluemix	Yes	Free up to 10k requests per month
Amazon Lex	Low	1: English	9, from Java to Go	93, from Alexa	15, from Alexa	No	Yes	Form-based	3: Twilio SMS, FB Messenger, and Slack	SDK	No, other services are in AWS	Yes	Free for the 1st year (with limits)
Recast.ai	Medium	16 at the standard level	7, from Python to Go	31, from colors to distance	3, several from the community	No	No	Form and block-based	Zero	Webhook and SDK	Yes	Yes	Free

These three works use different approaches for building a chatbot. While Shawar et al. (2005) uses the matching keywords pattern approach, Ranoliya et al. (2017) uses AIML and a pattern matching technique. Both approaches work well but the matching keywords pattern one works well for a small knowledge base, but not for a large number of rules. As for the AIML language, if the tags are not well formatted or in the right place, it will not work. Besides, it needs a parser to read the user's input and to search for an answer. Bala et al. (2018) use Natural Language Processing (NLP) techniques and a web-based system with user registration and login. Their system, though, is intended to answer queries related to any college related activities.

Our approach, on the other hand, uses DialogFlow which allows us to rapidly and efficiently create our conversational agent as well as integrate it to Telegram. It also uses NLP techniques to process the sentences added to the system. According to Canonico and Russis (2018), from a descriptive point of view, DialogFlow is the most complete NLU platform. In their study, they compared and analyzed six NLU platforms focusing on 13 different facets. Table 1 shows the results of their comparison. They concluded that IBM Watson was better at detecting the intent they created, whereas DialogFlow was the only tool which correctly detected the default intent for all sentences.

4. Our Approach

This section begins by presenting the system architecture that has been defined to lead the implementation of the proposed approach. Thus, all components of the architecture and their interactions are commented. After that, the implementation aspects of the proposed approach are discussed and commented taking into account the system architecture and justifying the technologies used in the implementation.

4.1. System Architecture

Figure 1 shows the system architecture. It starts with the input message from the user. Telegram API receives the query and sends it to DialogFlow which processes the sentence by its NLP algorithms. In the DialogFlow platform, intents were created for every question and answer. In the intents, there are training phrases and responses to them. After the query is matched with an intent, DialogFlow outputs the corresponding response that was stored in the intent.

As Figure 2 shows, for every agent, there may be different intents and for each intent, there are training phrases - in which we created sentences that the end-user might ask in different ways for the same question, for instance, an end-user could ask when the enrollment day would be. They could ask this question in different ways, i.e., *"When will the enrollment day be?"* or *"When is the day to enroll?"*. In DialogFlow documentation, they say that for an intent to work efficiently well, there should be created at least 10 training phrases so the agent can recognize a variety of end-user expressions.

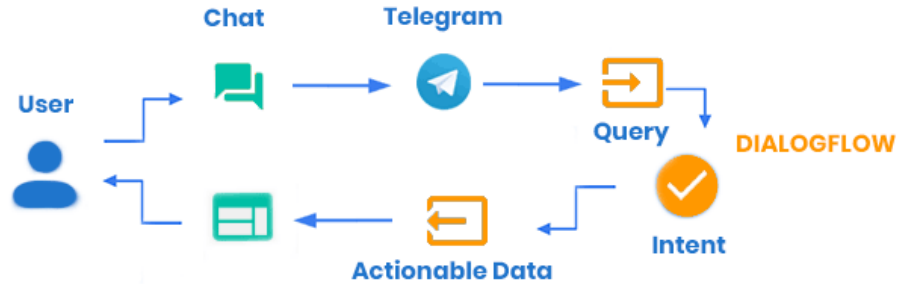


Figure 1. SMDbot system architecture

To validate and analyze our objective of reducing the workload of the administrative staff and helping students find information faster, we developed the application SMDbot as a PoC. SMDbot was developed in DialogFlow which has integration with Telegram, a messaging application. We chose Telegram as the user interface because it has easy integration with DialogFlow and because the university staff has been using the messaging app to respond to student's queries.

The conversational agent answers questions related to the FAQ provided by the university staff which contains 34 questions and answers. Figure 3(a) has an initial screen of the chatbot with all information about itself to the user. While Figure 3(b) presents an image of SMDbot responding to query performed in natural language in Portuguese through Telegram.

We added all the questions and corresponding answers to intents in the SMDbot agent. Typing questions and answers variations in the DialogFlow platform can take long depending on the size of the dataset. Since we began with a small one, it did not take

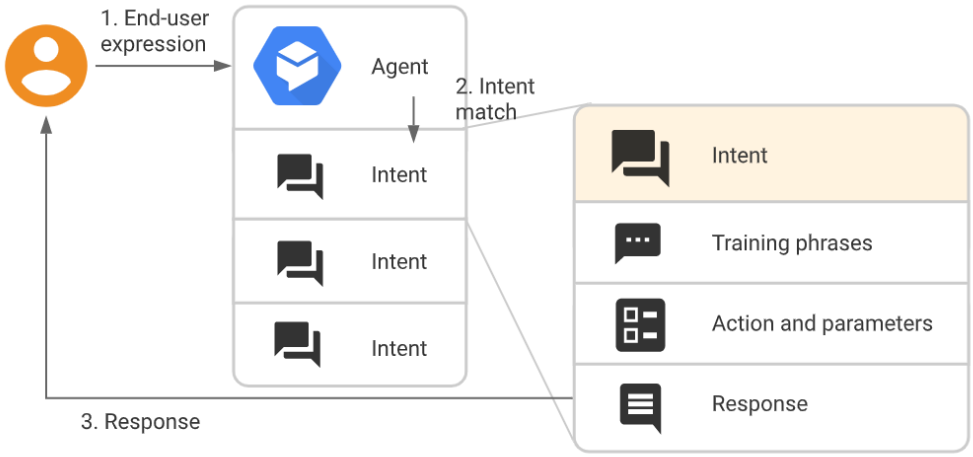
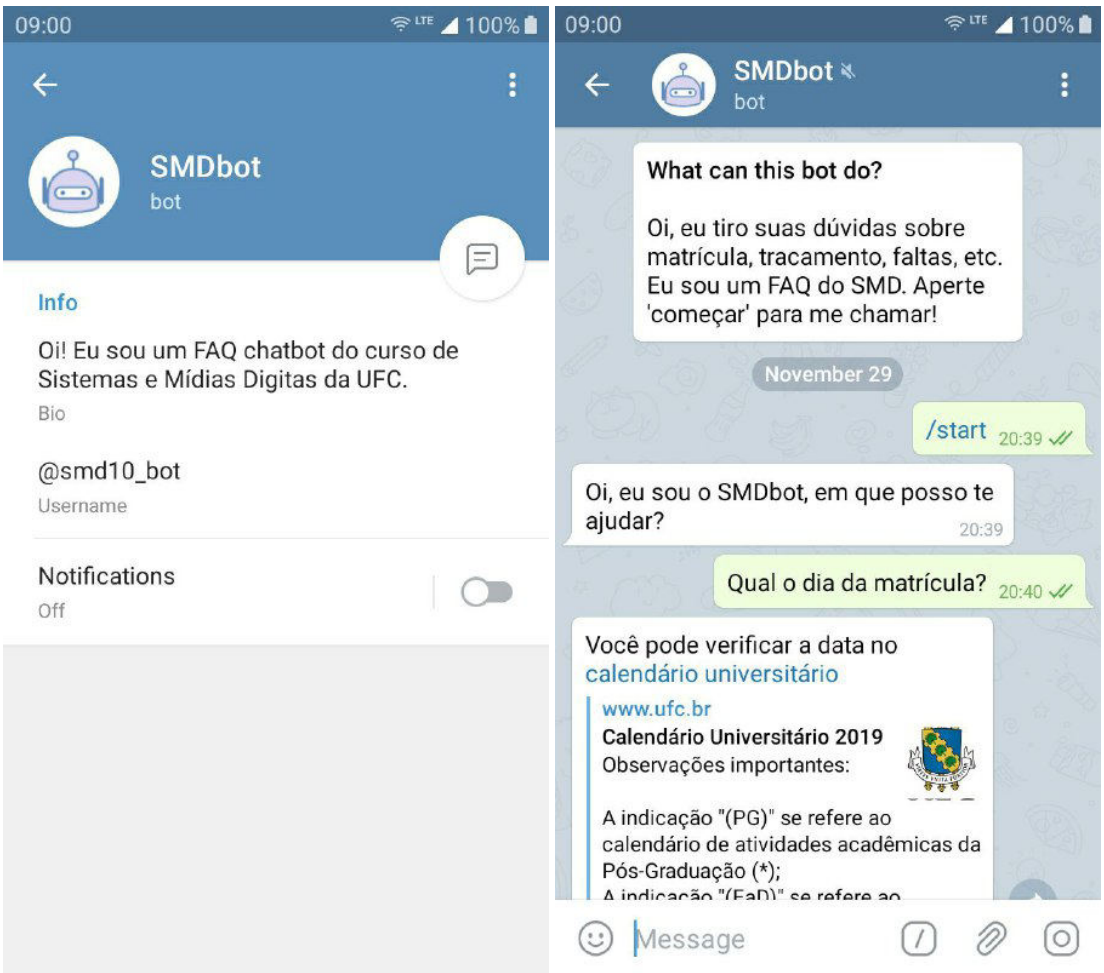


Figure 2. Workflow of the DialogFlow platform [DialogFlow 2019]



(a) Initial Screen (b) Example of Query in Portuguese on Telegram

Figure 3. SMDbot - Screens

long to feed the platform with the data. As soon as a new sentence was created in the platform, it automatically trained it and could already be tested. For every question and answer, an intent was created. DialogFlow already comes with two predefined intents. They are the default welcome intent to greet the users in natural language, and the default fallback intent helps catch all the questions that the bot does not understand and returns a response such as *"Sorry, I did not understand your question"* to respond to the user when the question does not match any intents. An entity is a property that DialogFlow uses to answer the request from the user. It is usually a keyword such as a name, date, location, etc. DialogFlow has pre-build entities. Two of them are *@sys.date* and *@sys.location* where the former stores date values and the latter stores location words from the trained phrases.

After the creation of all intents, we deployed the chatbot by integrating it to Telegram which is easy to do. First, we created a chatbot with BotFather (a Telegram bot to create your own chatbots) on Telegram itself. BotFather provides some info about the chatbot such as the token (a sequence of characters including numbers and letters) which gives us access to the bot. We enabled the Telegram integration on the DialogFlow platform and added the access token to it. By doing so, the bot could be started and could already respond to queries. The implementation of the chatbot from learning how to use DialogFlow to inserting and training the dataset took us three weeks in total. As for the insertion of the intents with questions and answers, it took us two days since the dataset was small.

5. Evaluation

This section aims to show the results of the evaluation of the chatbot to check if its responses are correct. For evaluation purposes, we asked 5 questions included in the dataset provided by the administrative staff. The Table 2 shows the response for every question.

The number 1 questions are sentences that are exactly the same as in the dataset we trained the chatbot with. The number 2 questions are a variant of the sentence, in order to test if the chatbot recognizes and matches it with the right answer. Words in the Answers column that are in bold and in italics are hyperlinks. So the user just needs to click on it, and they will be redirected to the page with the information.

The chatbot successfully responded to the questions as expected. In the sentence *"tem algum laboratrio livre"*, we typed the word *"laboratrio"* incorrectly and we did not add a question mark on purpose. The chatbot still recognized the sentence correctly and sent the right output. The sentence *"vou me graduar?"* uses a different verb than *"formar"* (it means "to graduate" in Portuguese) used in its related question number 1. The conversational agent was still able to match the right answer because we created an entity (which allows us to create synonyms for a word) for the verb *"formar"*. The user could type either *"formar"*, or *"graduar"* and the SMDbot would identify both as "graduate".

After testing other sentences that are in the dataset, we concluded that the results are positive and that the training of the sentences took only a few seconds with a small dataset. For better results, more questions and answers can be easily added to the chatbot platform by the administrative staff using the DialogFlow console.

Table 2. Questions and answers from the SMDbot dataset

Questions	**Answers**
1. Não sei se hoje haverá aula. Como saber? 2. como saber se hoje vai ter aula	Geralmente o cancelamento de aulas são divulgados pelo SIGAA.
1. Não poderei ir na colação de grau. O que fazer? 2. não posso ir na colação de grau	Você pode nomear um procurador para assinar a ata por você. Alguns dias antes da cerimônia, a coordenação sempre envia as demais orientações.
1. Como vou saber se irei me formar? 2. vou me graduar?	Você pode verificar no SIGAA => portal do discente-ensino-consultas do discente-pendências de conclusão. Se ainda assim estiver com dúvidas, pode nos procurar aqui na coordenação.
1. Como pedir aproveitamento de uma disciplina? 2. quero aproveitar disciplina	Você pode preencher o formulário ***Aproveitamento de Estudos*** e entregar na coordenação juntamente com o histórico e as ementas já cursadas.
1. Tem laboratório livre para estudo? Como saber? 2. tem algum laboratrio livre	Através do ***mapa de salas***, atualizado de tempo em tempo.

6. Conclusion and Future Work

The present work has been conducted with an aim to save time and energy of the administrative staff as well as facilitate access to information for students of an undergraduate course. To achieve this goal, we developed a chatbot, a system able to respond to requests in natural language in Portuguese via Telegram.

This paper showed the process of the SMDbot development using DialogFlow, a cloud-based platform which showed to be an easy and fast tool to create chatbots. The initial dataset was small (34 questions and answers) but enough to train the chatbot to correctly respond to the frequent asked questions. It also took only a few seconds to train the chatbot and deploy it. By testing some questions and their variants, we concluded that the chatbot answers were correct and able to deliver efficient and accurate responses, helping students to locate relevant information quickly without the need to reach the administrative officers to ask questions that were already responded by the chatbot.

As future work, we intend to develop a strategy to connect data from a spreadsheet file to DialogFlow, so the administrative staff can create more questions and answers as needed by creating data in the spreadsheet. Besides, we plan to evaluate the chatbot with both the administrative staff and the students to increase the number of its dataset, as well as integrate the chatbot with other messaging apps that DialogFlow supports.

References

Bala, K., Kumar, M., Hulawale, S., and Pandita, S. (2018). Chat-bot for college management system using a.i. pages 2395–0056.

Barros, F. and Azevedo, P. (2016). *Agentes Inteligentes Conversacionais: Conceitos Básicos e Desenvolvimento*, pages 169–218.

Canonico, M. and Russis, L. D. (2018). A comparison and critique of natural language understanding tools.

DialogFlow (2019). DialogFlow documentation. Available at: `https://dialogflow.com/docs`. Accessed: 2019-11-06.

Dibitonto, M., Leszczynska, K., Tazzi, F., and Medaglia, C. M. (2018). Chatbot in a campus environment: Design of lisa, a virtual assistant to help students in their university life. In *HCI*.

Lee, K., Jo, J., Kim, J., and Kang, Y. (2019). *Can Chatbots Help Reduce the Workload of Administrative Officers? - Implementing and Deploying FAQ Chatbot Service in a University*, pages 348–354.

Mauldin, M. L. (1994). Chatterbots, tinymuds, and the turing test: Entering the loebner prize competition. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (Vol. 1)*, AAAI '94, pages 16–21, Menlo Park, CA, USA. American Association for Artificial Intelligence.

McTear, M., Callejas, Z., and Griol, D. (2016). *The Conversational Interface: Talking to Smart Devices*. Springer Publishing Company, Incorporated, 1st edition.

Mitkov, R. (2003). *The Oxford Handbook of Computational Linguistics (Oxford Handbooks in Linguistics S.)*. Oxford University Press, Inc., New York, NY, USA.

Neto, A. J. R., Borges, M. M., and Roque, L. (2018). A preliminary study of proof of concept practices and their connection with information systems and information science. In *Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality*, TEEM'18, pages 270–275, New York, NY, USA. ACM.

Ranoliya, B. R., Raghuwanshi, N., and Singh, S. (2017). Chatbot for university related faqs. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1525–1530.

Shawar, B. A. and Atwell, E. (2007). Chatbots: Are they really useful? *LDV Forum*, 22:29–49.

Shawar, B. A., Atwell, E., and Roberts, A. (2005). Faqchat as in information retrieval system. In *Human Language Technologies as a Challenge for Computer Science and Linguistics: Proceedings of the 2nd Language and Technology Conference*, pages 274–278. Poznan : Wydawnictwo Poznanskie : with co-operation of Fundacja Uniwersytetu im. A. Mickiewicza.

Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, 49:433–460.

Valtolina, S., Barricelli, B. R., Gaetano, S. D., and Diliberto, P. (2018). Chatbots and conversational interfaces: Three domains of use. In *CoPDA@AVI*.

Wallace, R. (2003). The elements of aiml style. *Alice AI Foundation*, 139.

Weizenbaum, J. (1966). Eliza — a computer program for the study of natural language communication between man and machine. *Commun. ACM*, 9(1):36–45.

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

N672c Nobre, Graciane Xavier.
A Chatbot Approach to Automating FAQ Responses in an Undergraduate Course Domain / Graciane Xavier Nobre. – 2019.
11 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Instituto UFC Virtual, Curso de Sistemas e Mídias Digitais, Fortaleza, 2019.
Orientação: Prof. Dr. Leonardo Oliveira Moreira.
Coorientação: Prof. Me. José Wellington Franco da Silva.

1. Chatbot. 2. DialogFlow. 3. Conversational Agents. I. Título.

CDD 302.23
