



# Analysis of Tools for the Development of Conversational Agents <sup>†</sup>

Lamya Benaddi <sup>1,\*</sup> , Charaf Ouaddi <sup>1,\*</sup> , Ismaïl Khriss <sup>2</sup> and Brahim Ouchao <sup>1</sup>

<sup>1</sup> Software Engineering & Information Systems Engineering (GL-ISI) Team, Faculty of Sciences and Technics (FST), University of Moulay Ismail (UMI), Errachidia 52000, Morocco; ouchao@gmail.com

<sup>2</sup> Mathématique, Informatique et Génie, Université du Québec à Rimouski (UQAR), Rimouski, QC G5L 3A1, Canada; ismail\_khriss@uqar.ca

\* Correspondence: l.benaddi@edu.umi.ac.ma (L.B.); c.ouaddi@edu.umi.ac.ma (C.O.)

<sup>†</sup> Presented at the 3rd International Day on Computer Science and Applied Mathematics, Errachidia, Morocco, 13 May 2023.

**Abstract:** Conversational agents are being increasingly adopted in various domains, such as e-commerce and customer services, and as a direct communication channel between companies and end-users. Several tools have been developed to facilitate their definition and deployment. They exploit existing cloud infrastructures and artificial intelligence (AI) techniques to efficiently process users' input and extract conversational information. Major Information Technology (IT) companies, such as Google, IBM, Microsoft, and Amazon, have provided powerful tools to develop conversational agents. Still, choosing the most appropriate tool is not easy, as it may require high costs associated with automatic natural language processing (NLP) services and expertise in software engineering and AI. Therefore, this paper aims to analyze different tools to help developers and non-developers to choose the optimal tool for their specific scenario of creating a conversational agent.

**Keywords:** conversational agent; intelligence artificial; natural language processing



**Citation:** Benaddi, L.; Ouaddi, C.; Khriss, I.; Ouchao, B. Analysis of Tools for the Development of Conversational Agents. *Comput. Sci. Math. Forum* **2023**, *6*, 5. <https://doi.org/10.3390/cmsf2023006005>

Academic Editors: Abdeslam Jakimi and Mohamed Oualla

Published: 30 May 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Conversational agents (CA), virtual assistants, voice assistants, or chatbots are becoming part of our daily lives [1]. They provide users with various services via the use of natural language (NL). Users, for example, can inquire about the weather, ask questions, control home automation devices, such as coffee machines, book flights, and manage other essential tasks, such as emails and calendars.

As a result of the success of CAs, various technologies have been developed to create those systems. Tech Giants, such as Google, IBM, Microsoft, and Amazon, have released their CA creation tools, such as Dialogflow, Watson Assistant, bot framework, and Lex. Smaller companies, such as Rasa, Many-chat, FlowXO, and Pandorabots, have also proposed their tools. Those tools possess impressive capabilities, including NLP, Automatic Speech Recognition APIs (ASR), and speech synthesis. However, selecting the appropriate tool for a specific CA can be challenging due to the vast array of options. Moreover, operational factors, such as vendor lock-in and high costs, should also be considered.

Therefore, this work analyzes CA development tools to help developers and non-developers choose the most suitable solution for their scenario.

This paper is organized as follows: Section 2 presents an overview of how a CA works. Section 3 provides an analysis of agent development tools. Section 4 concludes our paper.

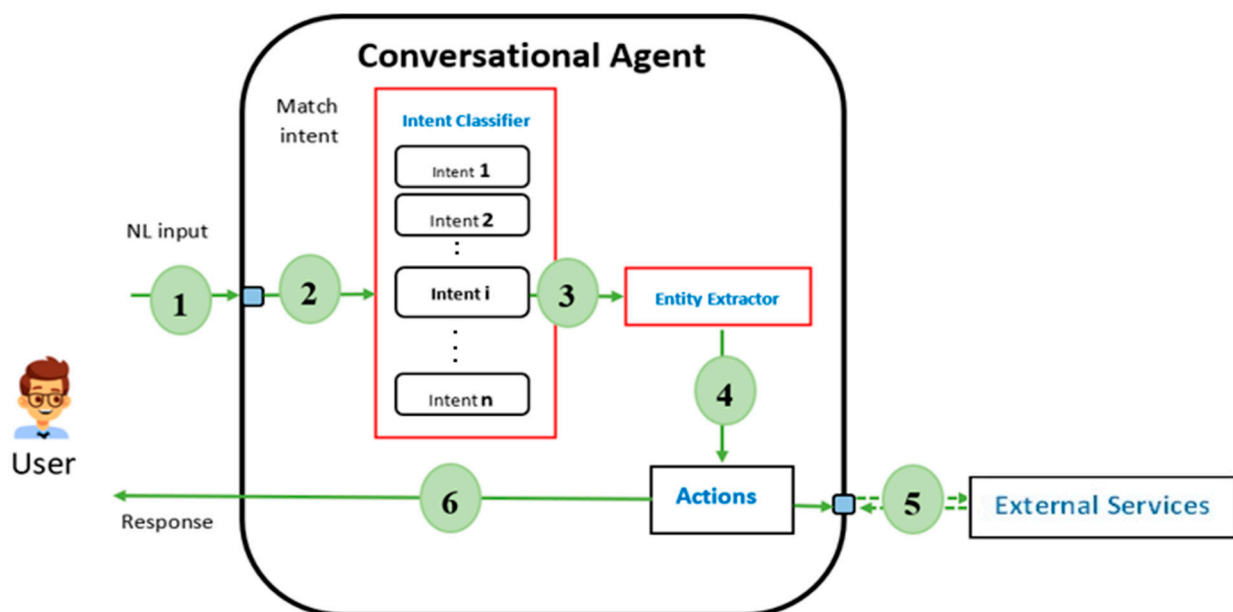
## 2. Building a Conversational Agent: An Overview

CAs are a good illustration of the progress in NLP [2]. Indeed, a CA is a computer program that is able to converse with users using NL. It can, thus, understand requests formulated using NL, process them, trigger actions, and formulate answers. CAs are

attracting increasing interest as they provide access to various services, such as flight booking or weather checking, via mobile applications, websites, or social networks, such as Telegram, Twitter, or Slack. This approach allows users to benefit from these services without the need to install new applications, and their interactions with the service are facilitated via an NL text or voice conversation [3].

CAs can be classified into two main categories [4]: proactive agents, which can initiate the conversation with users in a given context without an explicit request from them (e.g., by alerting them), and reactive agents, which can interact with users regarding specific tasks, such as a hotel or flight booking.

Figure 1 shows a simplified diagram of how a CA works [4,5]. A standard method of CA design is based on the use of “intents”, which essentially correspond to the objectives or goals that the user wishes to achieve when they are initiating a conversation with the CA.



**Figure 1.** Conversational agents working scheme (adapted from Refs. [4,5]).

Firstly, the CA receives the user’s input in NL (e.g., “I want to book a car from 1 March 2023 to 15 March 2023”, label 1 in Figure 1). It then tries to match the sentence to a specific intention (e.g., intention: book, label 2). Intents can be extracted from the text using different AI-based techniques, such as rule-based or machine learning-based ones. This task is essential in NLP and consists of identifying specific text elements and classifying them into predefined categories called “entities”. These entities can be names of people, dates, phone numbers, email addresses, currencies, etc. For our above request, the CA extracts the following entities: start date: 1 March 2023; end date: 15 March 2023 (label 3).

Then, the CA triggers an appropriate action by responding to the request (label 4). Actions may include sending a text response, performing an online task, or interacting with external services (label 5). Finally, it generates an NL response from the result of the action (label 6).

### 3. Analysis of the CA Development Tools

Various platforms, frameworks, and services are available to create CAs. These tools enable their creation for different messaging platforms, mobile applications, websites, and connected home devices.

This section analyzes several CA development tools, which are listed in Table 1. This analysis mainly focuses on the features offered by these tools and the concepts used to develop an agent.

**Table 1.** A comparative study of CA development tools.

Features and Concepts																			
	Type of Tool	Open-Source	Type of CA That Can Be Built		Pricing Model	Number of Languages Supported	NLP Service		Input Modalities			Conversion Assets				Deployment		Integration Channels	
Tools	(Library, Framework, Platform, Or Service)		Reactive	Proactive	Free version with all the basic features (f). Paid version: pay-as-you-go (p). Paid version: fixed amount (a).Pricing based on chosen features (c).	Represented by L4: more than 40. L3: more than 10. L2: less than 10.L1: 1.	Proprietary	Third-Party	Text	Voice	Intent	Training phrases	Regular expressions	Entities	Response: text, image, or httpRequest	Conversation flow structure	Cloud	On Premises	Integration with mobile apps/social networks/websites. C3: more than 10. C2: less than 10. C1: some.
Watson Assistant [6]	P	No	Yes	Yes	f, p, c	L2	Yes	No	Yes	Yes <sup>1</sup>	Yes	Yes	Yes	Yes	Yes	Context	Yes	Yes	C2
Rasa Framework <sup>2</sup> [7]	F	Yes	Yes	Yes	f, c	L4	Yes	No	Yes	No	Yes	Yes	Yes	Yes	Yes	Tree	No	Yes	C3
Wit.ai [8]	P	No	Yes	No	f	L4	Yes	No	Yes	Yes	Yes	Yes	-	Yes	Yes	Tree	Yes	No	C1
Manychat [9]	P	No	Yes	No	f, a, c	L2	Yes	Yes	Yes	No	Yes	No	No	Yes	Yes	Tree	Yes	No	C1
Chatfuel [10]	P	No	Yes	No	f, a	L3	Yes	No	Yes	No	No	No	No	No	Yes	Tree	Yes	No	C2
Chatterbot [11]	L	Yes	Yes	No	f	L4	Yes	No	Yes	No	No	No	No	Yes	No	Context	No	Yes	C3
FlowXO [12]	P	No	Yes	No	f, a	L3	No	Yes	Yes	No	Yes	No	Yes	Yes	Yes	Tree	Yes	No	C2
LUIS <sup>3</sup> [13]	S	No	Yes	Yes	p	L4	Yes	No	Yes	No	Yes	Yes	Yes	Yes	Yes	Tree	Yes	No	C3
QnA Maker [14]	S	No	Yes	No	f, p, c	L4	Yes	No	Yes	No	Yes	Yes	No	No	Yes	Tree	Yes	No	C3
Bot Framework [15]	F	Yes	Yes	Yes	f, p, c	L4	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Tree	Yes	Yes	C3
Xatkit [16]	P	Yes	Yes	Yes	f, a	L2	No	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Tree	No	Yes	C2
Botsify [17]	P	No	Yes	No	f, a	L4	Yes	Yes	Yes	No	Yes	Yes	No	Yes	Yes	Tree	Yes	No	C2
SmartLoop [18]	P	No	Yes	Yes	f, p	L2	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Tree	Yes	No	C2
Dialogflow [19]	P	No	Yes	Yes	f, p	L4	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Context	Yes	No	C3
Amazon Lex [20]	P	No	Yes	Yes	f, p, c	L2	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Context	Yes	No	C3
Botkit [21]	F	Yes	Yes	Yes	f, c	L4	No	Yes	Yes	Yes	Yes	Yes	-	Yes	Yes	Tree	Yes	Yes	C3
Amelia [22]	P	No	Yes	-	f	L4	No	Yes	Yes	Yes	Yes	Yes	-	Yes	Yes	Context	Yes	No	C3
Smartly [23]	P	No	Yes	No	f, a	L4	Yes	No	Yes	Yes	Yes	Yes	-	Yes	Yes	Context	Yes	No	C2
Pandorabots [24]	P	Yes	No	No	f, c	L4	-	Yes	Yes	Yes	No	No	No	No	Yes	Tree	Yes	Yes	C1
Houndify [25]	P	No	Yes	No	f, c	L3	No	Yes	Yes	Yes	No	-	-	-	Yes	Context	Yes	Yes	C1

<sup>1</sup> Voice offered through Watson Speech to Text. <sup>2</sup> Rasa framework is downloadable and open source. <sup>3</sup> Natural language processing through LUIS will still be cloud-based. - means that the information is not available.

With there being many tools available to create CAs, choosing the best one can be challenging for developers. It is important to consider several factors, such as the features offered, the project's complexity, the developers' expertise, the budget, and the type of application.

These tools can be open source (such as Rasa framework), which allows more flexibility and customization, or closed source (such as Dialogflow and Amazon Lex), which are often easier to use and provide more comprehensive technical support. Closed source tools are offered on a subscription- or usage-based pricing model, which can vary considerably depending on the complexity of the tool, its functionality, and the level of support the vendor provides.

Most tools use user interfaces to create training sentences (such as Dialogflow, Bot Framework with LUIS, and Watson Assistant), allowing developers to define intentions and entities from these training sentences. On the other hand, some CA development tools use regular expressions to detect patterns in text. In contrast to training sentences, regular expressions allow the identification of more complex patterns in a sentence. They are beneficial for spotting keywords in a sentence that can reveal the user's intent. Regular expressions can, therefore, complement training sentences to improve the CA's ability to understand users' requests.

Some CA development tools support two input modes, text and voice, while others focus solely on one or the other. Text-based CA can be integrated with popular messaging apps, such as Facebook Messenger, WhatsApp, or Slack. In contrast, a voice-based CA can be integrated with devices such as voice assistants, for example, Amazon Alexa, Google Assistant, and Apple Siri.

Implementing a CA involves choosing the tool that best suits the conversational scenarios and the client's needs. The goal is to converse with many customers via social media platforms. In that case, a tool that supports the development of multilingual CA that can use different communication channels is necessary. Additionally, if the developer does not have the resources to host the infrastructure, tools offering hosting services are the best choice.

#### 4. Conclusions

CA development has become increasingly accessible due to the variety of tools available on the market and advancements in AI. However, choosing a suitable tool is not easy, as it depends on the specific needs and requirements of the project, including the required functions, supported languages, deployment mode, and pricing. In this paper, we present an analysis of 20 tools that can help developers choose the right tool to support the development of their CAs in conformance with their specific requirements.

**Author Contributions:** Conceptualization, L.B., C.O. and I.K.; methodology, L.B. and C.O.; software, L.B. and C.O.; validation, I.K.; formal analysis, L.B., C.O., I.K. and B.O.; investigation, L.B. and C.O.; resources, L.B. and C.O.; data curation, L.B. and C.O.; writing—original draft preparation, L.B. and C.O.; writing—review and editing, I.K.; visualization, L.B., C.O., I.K. and B.O.; supervision, I.K. and B.O.; project administration, I.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Center for Scientific and Technical Research (CNRST) under a Moroccan project called "AI-Khawarizmi program in AI and its Applications".

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

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

## References

1. Hoy, M.B. Alexa, Siri, Cortana, and more: An introduction to voice assistants. *Med. Ref. Serv. Q.* **2018**, *37*, 81–88. [CrossRef] [PubMed]
2. Nagarhalli, T.P.; Vaze, V.; Rana, N. A review of current trends in the development of chatbot systems. In Proceedings of the 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 6–7 March 2020; IEEE: New York, NY, USA, 2020; pp. 706–710.
3. Brandtzaeg, P.B.; Følstad, A. Why people use chatbots. In Proceedings of the Internet Science: 4th International Conference, INSCI 2017, Thessaloniki, Greece, 22–24 November 2017; pp. 377–392.
4. Sarikaya, R. The technology behind personal digital assistants: An overview of the system architecture and key components. *IEEE Signal Process. Mag.* **2017**, *34*, 67–81. [CrossRef]
5. Adamopoulou, E.; Moussiades, L. An overview of chatbot technology. In Proceedings of the Artificial Intelligence Applications and Innovations: 16th IFIP WG 12.5 International Conference, AIAI 2020, Neos Marmaras, Greece, 5–7 June 2020; pp. 373–383.
6. IBM Watson Assistant. Available online: <https://www.ibm.com/cloud/watson-assistant/> (accessed on 10 October 2022).
7. RasaOpen Source. Available online: <https://rasa.com> (accessed on 10 October 2022).
8. WIT.ai. Available online: <https://wit.ai/> (accessed on 10 October 2022).
9. Manychat. Available online: <https://manychat.com/> (accessed on 10 October 2022).
10. Chatfuel. Available online: <https://chatfuel.com/> (accessed on 10 October 2022).
11. ChatterBot. Available online: <https://chatterbot.readthedocs.io/en/stable/> (accessed on 10 October 2022).
12. FlowXO. Available online: <https://flowxo.com/> (accessed on 10 October 2022).
13. LUIS. Available online: <https://www.luis.ai/> (accessed on 22 October 2022).
14. QnA Maker. Available online: <https://www.qnamaker.ai/> (accessed on 22 October 2022).
15. Microsoft Bot Framework. Available online: <https://dev.botframework.com/> (accessed on 22 October 2022).
16. Daniel, G.; Cabot, J.; Deruelle, L.; Derras, M. Xatkit: A Multimodal Low-Code Chatbot Development Framework. *IEEE Access* **2020**, *8*, 15332–15346. [CrossRef]
17. Botsify. Available online: <https://botsify.com/> (accessed on 14 November 2022).
18. SmartLoop. Available online: <https://smartloop.ai/> (accessed on 14 November 2022).
19. Dialogflow. Available online: <https://dialogflow.com/> (accessed on 14 November 2022).
20. Amazon Lex. Available online: <https://aws.amazon.com/en/lex/> (accessed on 14 November 2022).
21. Botkit. Available online: <https://botkit.ai/> (accessed on 2 December 2022).
22. AMELIA. Available online: <https://amelia.ai/conversational-ai/> (accessed on 2 December 2022).
23. Smartly. Available online: <https://www.smartly.ai/> (accessed on 2 December 2022).
24. Pandorabots. Available online: <https://pandorabots.com> (accessed on 2 December 2022).
25. SoundHound. Available online: <https://www.soundhound.com/voice-ai-products/nlu/> (accessed on 2 December 2022).

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