



Installation and Documentation Evaluation of Recent (01 January 2020 to 15 February 2021) Chatbot Engines from Python Package Index (PyPI)

Kian Seng Sim and Maurice HT Ling*

School of Applied Sciences, Temasek Polytechnic, Singapore

***Corresponding Author:** Maurice HT Ling, School of Applied Sciences, Temasek Polytechnic, Singapore.

Received: June 23, 2021

Published: July 19, 2021

© All rights are reserved by **Kian Seng Sim and Maurice HT Ling.**

Abstract

Chatbots have its roots in the early days of computing and gain substantial popularity in recent years. The most critical component of a chatbot is the engine that accepts and responds in natural human language input. In this study, we evaluate the installation and documentation of 21 recent chatbot engines (01 January 2020 to 15 February 2021) indexed in the Python Package Index (PyPI). Fourteen engines can be installed and imported without warning or errors and four engines have rich documentation. Only three (ChatterBot, chatbotAI, and opsdroid) engines with rich documentation can be installed and imported without warnings or errors. This suggests that the majority of the available and recent Python chatbot engines are not ready for widespread usability.

Keywords: Systematic Review; Chatbot Engines; Python Package Index (PyPI); Anaconda

Introduction

Chatbots, also known as conversational systems or dialogue engines, are software agents providing access to services and information using natural human languages [1,2]. The roots of chatbots went back as early as 1950 when Alan Turing [3] conceptualized the Turing test in the form of a conversational system. The first chatbot, ELIZA, was developed by Joseph Weizenbaum in 1966 [4]. Since then, many chatbots had been developed in the pursuit of artificial general intelligence [5].

There have been many recent applications of chatbots; especially in education, customer service, robotics, and healthcare; with numerous deployments for industrial use [6]. In education, chatbots can be used for content recommendation for learners [7]. In customer service, chatbots aim to supplement or even displace human service agents but results have not been satisfactory [8]. In

robotics, chatbot aims to provide a natural language interface for physical/mobile robots [6]. In healthcare, a chatbot may be used as triage assistants [9] or supporting patients for chronic diseases self-management [10].

The core component of any chatbot application is the chatbot engine, such as AliMe Chat [11], which can be defined as a software component that accepts a natural human language input, processes, and respond with an output in a natural human language. In TIOBE Index published in June 2021, Python overtook Java to be the second most popular programming language after C programming language. In this study, we compare 21 recent chatbot engines (01 January 2020 to 15 February 2021) indexed in the Python Package Index (PyPI) in terms of installation and documentation. Only three (ChatterBot, chatbotAI, and opsdroid) engines with rich documentation can be installed and imported without warnings or errors.

Materials and Methods

Chatbot engine candidates: Potential evaluation candidates were isolated from Python Package Index (PyPI) using “Chatbot” as the search term (<https://pypi.org/search/?q=Chatbot>) on February 16, 2021; where release date of up to and inclusive of February 15, 2021. From the list of potential candidates, release date before 2020 (released or last updated before and inclusive of December 31, 2019) were excluded to yield a list of recent candidates. From the list of recent candidates, only packages with function(s) to accept an English sentence or phrase and reply with a corresponding English sentence or phrase, with no internet requirement (hence, standalone packages rather than connectors to third-party web services), are chatbot engine candidates. Functional and/or extendable chatbots were also excluded.

Test system

The following hardware and software were used in the tests. HP Pavilion Power Laptop 15-cb0xx with Intel[®] Core™ i7-7700HQ CPU @ 2.80GHz, 2 GPUs (GPU 1: Intel[®] HD Graphics 630, and GPU 2: NVIDIA GeForce GTX 1050), 8GB Samsung 2400MHz and 4GB SK Hynix 2400MHz, 1 terabyte SATA HGST Travelstar 7K1000 HT-S721010A9E630 (0J22423) 7200 RPM hard disk with 32MB Cache at 6.0 gigabyte per second. The operating system was Windows 10 Home 64-bit Version 19042.804. Anaconda3 2020.11 distribution with Python 3.8.8 was used as Python setup and Visual Studio Code 1.53.2 as the integrated development environment.

Evaluation criteria

Each of the chatbot engine candidates was evaluated using the following criteria on a standard laptop with Windows operating system using 64-bit Anaconda distribution for Python 3.8.8 (Anaconda3 2020.11) with a new default environment for each candidate (compared against a new default environment named “baseline”):

- Criteria 1: Installation
- Criteria 1A: Requirement of administrator’s rights.
- Criteria 1B: Successful installation using pip with version specificity preferred.
- Criteria 1C: Ability to import engine into Python environment.
- Criteria 2: Basic Usage

- Criteria 2A: Availability of tutorial/documentation to demonstrate basic use case (accepting input and providing response).
- Criteria 2B: Availability of example script(s) to demonstrate basic use case.

Results and Discussion

Ecosystem of chatbot packages: A total of 336 packages were found from Python Package Index (PyPI) using “Chatbot” as the search term (<https://pypi.org/search/?q=Chatbot>) on February 16, 2021. Of which, 158 (47%) packages were last updated on or before December 31, 2019 (Figure 1, see supplementary materials for package listing). Out of the remaining 178 packages, only 21 (11.8% of recent chatbot related packages or 6.25% of chatbot related packages) are chatbot engines, which are reviewed in this study.

This suggests that a vast majority (88.2%) of the recent chatbot related packages are not chatbot engines but other aspects of the chatbot ecosystem (Figure 2). Complete chatbots, such as Dingtalk-Chatbot, are at the user-facing end; which generally requires deployment channels, such as Facebook or Telegram, to provide the means of communication. The interface from deployment channels to the chatbot engines consists of deployment channel-specific APIs (application programming interfaces). For example, if the deployment channel of choice is Slack, then the deployment API can be slack-api and the human user interacts with the chatbot using Slack. Besides chatbot engines, there are third-party chatbot engines, such as Dialogflow which is hosted in Google Cloud. Depending on the third-party chatbot engines, chatbots can be built either on the third-party platform or communicate with the third-party chatbot engine through a communication API. As such, third-party chatbot engines can effectively be closed source and out of the control of chatbot developers as chatbot developers use these third-party chatbot engines in a Software as a Service (SaaS) model [12]. Chatbot engines generally require a set of supporting packages; such as corpus which contains a human text for the chatbot engine to learn [13], artificial intelligence (AI) or machine learning (ML) tools to generate an output from a conversational input, parsers to process a human language input, and persistence packages to store the learning into a data storage system.

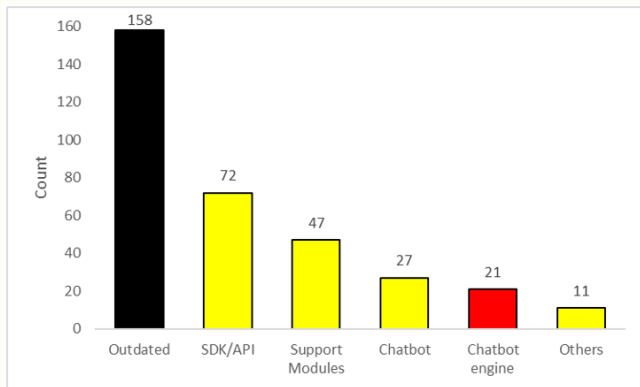


Figure 1: Distribution of packages.

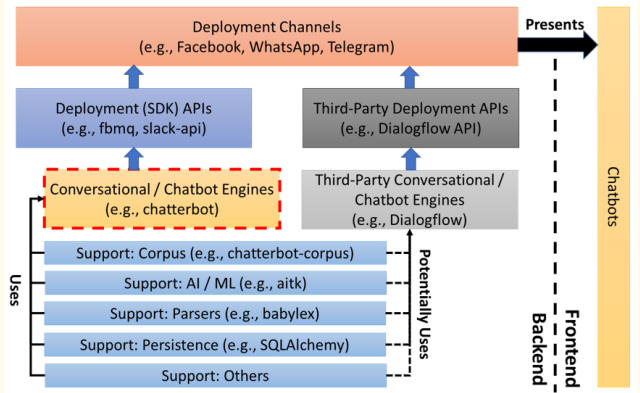


Figure 2: Ecosystem of chatbot packages.

Fourteen engines can be installed and imported without warnings or errors

Of the 21 engines (Table 1); 2 (9.5%) engines (flipgenic, and TChatBot) cannot be installed, 3 (14.3%) engines (coco-puppet, dazu, and deeppavlov) cannot be imported, and 2 (9.5%) engines (chatbot-creator, and emora-stdm) can be imported but with warning or errors. Of the remaining 14 engines that can be installed and imported without warning or errors, ChatterBot cannot be installed with a specific version and after successful non-version specific installation, pytz package required manual installation be-

fore ChatterBot can be imported. None of the packages requires administrator’s rights for installation. The required time to install an engine ranged from about 4 seconds (minette) to more than 10 minutes (ChatterBot), and the number of additional packages installed ranged from 2 (minette) to 106 (qary). Our experience supports prior experience that installation and execution of published tools can be challenging [14-16].

Package Name (Version)	Time to install chatbot engine (seconds)	Number of Libraries in Virtual Environment	Importing Library into Python
Baseline	0.0	11	Not applicable
Agt (0.6.1)	41.3	32	Import as agt
Bani (0.6.3)	689.7	56	Import as Bani
chatbotAI (0.3.0.0)	12.6	16	Import as chatbot
Chatbot-creator (0.0.5)	459.0	90	Import as ChatbotCreator with warning
Chatbotmaker (0.0.13)	6.6	14	Import as chatbotmaker
ChatterBot (1.0.8)	737.6	28 (Unable to install specific version)	Unable to import - pytz module not found. Import as chatterbot after manual installation of pytz
Cobe (3.0.1)	30.8	23	Import as cobe
Coco-puppet (0.4.3)	18.8	28	Unable to import as cocopuppet or coco_puppet
Commandinte-grator (1.3.0)	7.3	17	Import as commandintegra-tor
Dazu (0.1.0)	16.8	23	Unable to import - yaml module not found and unable to install yaml through pip
Deeppavlov (0.14.0)	113.9	38	Unable to import - module not found
Deeppavlov-agent (2.2.0)	28.6	35	Import as deeppavlov_agent

Emora-stdm (1.95)	116.2	51	Import as emora_stdm with error loading Spacy
Flipgenic (0.5.0)	5.67	11 (Unable to install, dependencies not satisfied)	Not applicable
Iokobot (0.6)	45.9	17	Import as iokobot
Jigbot (0.1.1)	109.4	44	Import as jigbot
Minette (0.4.3)	4.2	14	Import as minette
Msgflow (0.6.0)	34.3	23	Import as msgflow
Opsdroid (0.20.2)	78.2	68	Import as opsdroid
Qary (0.6.22)	267.5	117	Import as qary
TChatBot (0.1.0)	2.0	11 (Unable to install, dependencies not satisfied)	Not applicable

Table 1: Installation and import of chatbot engines.

Four engines contains rich documentation

The quality of documentation is classified into 4 categories: (a) None or no documentation. (b) Limited documentation, which is usually 1-page documentation (such as a readme file) with no images and consisting of little to no tutorials. (c) Good documentation where documentation is substantial enough for multiple sections with tutorials provided but limited use of images used for explanation. Generally, the package is relatively being used widely. (d) Rich documentation with numerous sections of documentation supported by many images to assist in understanding. Tutorials are also provided. Generally, the package is widely used.

In our survey, 14 (67%) engines have limited or no documentation with only 4 (19%) having rich documentation (Table 2). Moreover, 3 engines (cobe, deeppavlov-agent, and jigbot) do not contain installation information on PyPI page. This led to significant difficulties in installation and using these engines, especially for persons attempting to try out chatbot development. For example, deeppavlov-agent should be installed using “pip install deeppavlov-agent” instead of “pip install deeppavlov-agent” (underscore vs hyphenation) as listed on its PyPI page (Table 1). In addition, deeppavlov has rich documentation (Table 2) but cannot be im-

ported (Table 1). The importance of documentation cannot be understated [17]. Even developers themselves want to have documentation that they can refer to and code samples that they can use [18]. Yet, most developers reported more enjoyment in writing code compared to writing documentation [19]. As a result, most open-sourced software packages have been plagued with poor and incomplete documentation [20], which can present as barriers to contribution and usage.

Package Name	Basic Installation on PyPI Page	Source Code/ Documentation on GitHub	Documentation Level
Agt	Yes	Yes	Limited
Bani	Yes	Yes	Limited
ChatbotAI	Yes	Yes	Rich
Chatbot-creator	Yes	Yes	Limited
Chatbotmaker	Yes	Yes	Limited
ChatterBot	Yes	Yes	Rich
Cobe	No	Yes	None
Coco-puppet	Yes	Yes	Limited
Commandintegrator	Yes	Yes	Limited
Dazu	Yes	No	Limited
Deeppavlov	Yes	Yes	Rich
Deeppavlov-agent	No	Yes	Limited
Emora-stdm	Yes	Yes	Good
Flipgenic	Yes	Yes	Limited
Iokobot	Yes	Yes	Limited
Jigbot	No	No	None
Minette	Yes	Yes	Good
Msgflow	Yes	Yes	Limited
Opsdroid	Yes	Yes	Rich
Qary	Yes	No	Limited
TChatBot	Yes	Yes	Good

Table 2: Documentation of chatbot engines.

Conclusion

Of the 21 recent chatbot engines examined, only three (ChatterBot, chatbotAI, and opsdroid) engines with rich documentation can be installed and imported without warnings or errors, suggesting

that majority of the available and recent Python chatbot engines are not ready for widespread usability.

Supplementary Materials

Supplementary materials can be downloaded at <http://bit.ly/ChatbotEnginesA>.

Data Availability

Data files for this study can be downloaded at <http://bit.ly/ChatbotEnginesB>.

Acknowledgement

Kian Seng Sim is funded by the Temasek Polytechnic Career Kick-starter Programme.

Disclaimer

Views presented in this article are that of the authors and not the institution.

Conflict of Interest

The authors declare no conflict of interest.

Bibliography

1. Skjuve M., et al. "My Chatbot Companion - a Study of Human-Chatbot Relationships". *International Journal of Human-Computer Studies* 149 (2021): 102601.
2. Brandtzaeg PB and Følstad A. "Chatbots: Changing User Needs and Motivations". *Interactions* 25.5 (2018): 38-43.
3. Turing AM. "Computing Machinery and Intelligence". *Mind* LIX.236 (1950): 433-460.
4. Weizenbaum J. "ELIZA - A Computer Program for the Study of Natural Language Communication Between Man and Machine". *Communications of the ACM* 9.1 (1966): 36-45.
5. Yampolskiy RV and Fox J. "Artificial General Intelligence and the Human Mental Model". In: *Singularity Hypotheses*. Springer (2012): 129-145.
6. Adamopoulou E and Moussiades L. "Chatbots: History, Technology, and Applications". *Machine Learning with Applications* 2 (2020): 100006.
7. Smutny P and Schreiberova P. "Chatbots for Learning: A Review of Educational Chatbots for the Facebook Messenger". *Computer Education* 151 (2020): 103862.
8. Adam M., et al. "AI-Based Chatbots in Customer Service and Their Effects on User Compliance". *Electronic Markets* (2020).
9. Reiswich A and Haag M. "Evaluation of Chatbot Prototypes for Taking the Virtual Patient's History". *Studies in Health Technology and Informatics* 260 (2019): 73-80.
10. Hernandez JPT. "Network Diffusion and Technology Acceptance of A Nurse Chatbot for Chronic Disease Self-Management Support : A Theoretical Perspective". *Journal of Medical Investigation* 66.1.2 (2019): 24-30.
11. Qiu M., et al. "AliMe Chat: A Sequence to Sequence and Rerank based Chatbot Engine". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vancouver, Canada: Association for Computational Linguistics (2017): 498-503.
12. Mohammed CM and Zeebaree SRM. "Sufficient Comparison Among Cloud Computing Services: IaaS, PaaS, and SaaS: A Review". *International Journal of Business* 5.2 (2021): 17-30.
13. Maslowski I., et al. "In-the-wild Chatbot Corpus: From Opinion Analysis to Interaction Problem Detection". In: *International Conference on Natural Language and Speech Processing* (2017).
14. Peng B., et al. "Genetic Simulation Resources and the GSR Certification Program". *Bioinformatics* 35.4 (2019): 709-710.
15. Dange MC., et al. "Evaluation of Freely Available Software Tools for Untargeted Quantification of 13C Isotopic Enrichment in Cellular Metabolome from HR-LC/MS Data". *Metabolic Engineering Communications* 10 (2020): e00120.
16. Yousif A., et al. "NASQAR: A Web-Based Platform for High-Throughput Sequencing Data Analysis and Visualization". *BMC Bioinformatics* 21.1 (2020): 267.
17. Chomal VS and Saini JR. "Significance of Software Documentation in Software Development Process". *International Journal of Engineering Innovation and Research* 3.4 (2016): 410-416.
18. Bottomley C. "What Part Writer? What Part Programmer? A Survey of Practices and Knowledge Used in Programmer Writing".

- ing". In: Proceedings of International Professional Communication Conference 2005. Limerick, Ireland: IEEE (2005): 802-812.
19. Geiger RS, *et al.* "The Types, Roles, and Practices of Documentation in Data Analytics Open Source Software Libraries: A Collaborative Ethnography of Documentation Work". *Computer Supported Cooperative Work* 27.3-6 (2018): 767-802.
20. Cereceda O and Quinn DEA. "A Graduate Student Perspective on Overcoming Barriers to Interacting with Open-Source Software". *FACETS* 5.1 (2020): 289-303.

Volume 3 Issue 8 August 2021

© All rights are reserved by Kian Seng Sim and Maurice HT Ling.