FLOSS FAQ chatbot project reuse: how to allow nonexperts to develop a chatbot

Conference Paper · August 2019

DOI: 10.1145/3306446.3340823

CITATIONS
12

READS
781

2 authors, including:

Carla Rocha
University of Brasília
21 PUBLICATIONS **259** CITATIONS

SEE PROFILE

FLOSS FAQ chatbot project reuse - how to allow nonexperts to develop a chatbot

Arthur R. T. de Lacerda*
Carla S. R. Aguiar*
arthurrtl@gmail.com
caguiar@unb.br
UnB Faculty in Gama - University of Brasilia
Brasilia, Brazil

ABSTRACT

FAQ chatbots possess the capability to provide answers to frequently asked questions of a particular service, platform, or system. Currently, FAQ chatbot is the most popular domain of use of dialog assistants. However, developing a chatbot project requires a full-stack team formed by numerous specialists, such as dialog designer, data scientist, software engineer, DevOps, business strategist and experts from the domain, which can be both time and resources consuming. Language processing can be particularly challenging in languages other than English due to the scarcity of training datasets.

Most of the requirements of FAQ chatbots are similar, domain-specific, and projects could profit from Open Source Software (OSS) reuse. In this paper, we examine how OSS FAQ chatbot projects can benefit from reuse at the project level (black-box reuse). We present an experience report of a FLOSS FAQ chatbot project developed in Portuguese to an e-government service in Brazil. It comprises of the chatbot distribution service, as well as for analytics tool integrated and deployed on-premises. We identified assets that could be reused as a black-box and the assets that should be customized for a particular application. We categorized these assets in architecture, corpus, dialog flows, machine learning models, and documentation. This paper discusses how automation, pre-configuration, and templates can aid newcomers to develop chatbots in Portuguese without the need for specialized skills required from tools in chatbot architecture. Our main contribution is to highlight the issues non-English FAQ chatbots projects will likely face and the assets that can be reused. It allows non-chatbot experts to develop a quality-assured OSS FAQ chatbot in a shorter project cycle.

CCS CONCEPTS

• **Computing methodologies** → **Discourse, dialogue and pragmatics**; • **Software and its engineering** → **Open source model**; *Reusability*.

KEYWORDS

FLOSS, Open source, OSS, FLOSS FAQ chatbot, Black-Box reuse, Portuguese chatbot, experience report, e-government, conversational agents.

ACM Reference Format:
Arthur R. T. de Lacerda and Carla S. R. Aguiar. 2019. FLOSS FAQ chatbot project reuse - how to allow nonexperts to develop a chatbot. In *The 15th International Symposium on Open Collaboration (OpenSym '19), August 20–22, 2019, Skövde, Sweden.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3306446.3340823

1 INTRODUCTION

Chatbots are conversational software agents that must have natural language processing (NLP) to understand users intentions [18] and has become increasingly popular. They conduct conversations with humans, integrating services, users, and communication channels. Examples of chatbot applications are notification, FAQ [11, 23], virtual host [15], and personalized assistants [17].

The most common type of conversational agent is FAQ chatbots, which possesses the capability to provide answers to frequently asked questions of a particular service, platform or system [23]. A consequence of using FAQ chatbots in organizations services is the decrease of the demand on other communication channels, such as calls and e-mails [10].

OSS chatbots allow to keep all conversation data on-premises and facilitate their compliance with data regulations, such as the General Data Protection Regulation (GDPR) [9, 28]. They can benefit from collaborative modeling [21], and other bot

*Both authors contributed equally to this research.

OpenSym '19, August 20–22, 2019, Skövde, Sweden

ACM ISBN 978-1-4503-6319-8/19/08…$15.00
https://doi.org/10.1145/3306446.3340823

data, such as personality, vocabulary, and chit chat datasets. Chatbot architecture, composed by the integration of both building/creating a chatbot and distributing/messaging service, algorithms, and analytics is transparent in FLOSS. However, most of the FLOSS chatbot frameworks are domain-agnostic, where they handle one specific layer of the chatbot architecture and they leave the developer to configure algorithms and parameters for their given application. The documentation found in these communities is mostly technical, done by and to experts of the field. It thus discourages nonexperts practitioners to choose which FLOSS chatbot framework to reuse for a specific application reliably.

In this work, we investigate the assets necessary to FLOSS FAQ chatbot project to be reused as a black-box. We chose the application domain of FAQ to restrain the scope since the requirements specificity affects software positively reuse success [20]. Not only code reuse, but also architecture, content (dialogues, intentions), and machine learning conversation models. We present an experience report, where we conducted a project of a FLOSS FAQ chatbot developed during 15 months of a government-academia collaboration in Brazil, with best practices based on FLOSS ecosystems, agile and DevOps. Our main contribution is to identify and trace assets to maximize reuse in this application domain, regarding toolset, architecture, algorithms, training dataset, and documentation. We collect and analyze data from the project repository and data from analytics. We contribute to guide nonexperts practitioners to develop their FAQ chatbot in reduced project time and effort.

2 RELATED WORK

According to Lebeuf in [11], a chatbot project is composed of distribution and creation services. Distribution services are employed to manage the conversation data, it is the interface between the chatbot and the user, while creation services are used to develop the chatbot knowledge and behavior. Optionally, the business analytics service helps the improvement and evolution of the chatbot. The present work is focused on OSS chatbot distribution services.

The distribution service itself is composed of the following components [8]: Natural Language Understanding (NLU), communication channels, voice user interfaces, dialog management.

A vast number of chatbot frameworks is available in OSS, such as Botkit [11], Rasa [2], Botpress [3], among others. Typically, OSS chatbot frameworks are designed for developers and specialists, and an issue is to enable nonexperts to use them as a black box [28]. An exception is botpress, a complete chatbot architectural integrated solution, with a giving user interface that guides nonexperts from bot content creation to deployment. Limitations of this solution it is being both agnostic-domain and restrains the algorithms used. Dialog management that handles dialog context and decides the next action for the agent to take have drifted from pure rule-based to data-driven in recent years [8, 19].

Code reuse allows for previously tested and quality-assured code to be implemented in another system and it provides benefits regarding simply adding and enhancing system features [22]. In chatbot projects, both white-box and black-box reuse are common. However, most of the research done focus on the reuse of training datasets [7, 14, 19].

One crucial part of the chatbot is to understand user messages. According to Liu et al. [13], Natural Language Understanding Services perform similarly in both OSS and proprietary solutions. Another grand challenge is the effectiveness of chatbots in non-English speaking countries [14]. When the objective is to build a chatbot in other languages than English, dialogue datasets are the most significant difficulty [7], but it still possible as shown in [24], a German chatbot was built to guide an idea submission process. ParlAI is an open-source platform that aims to minimize this dataset issue by providing a unified framework for sharing, training, and testing dialog models [16]. They aim to reinforce reuse of training datasets by sharing a repository of the corpus, utterances, and machine learning models. Another alternative is the adoption of crowdsourcing to write utterances. Paetzel et al. in [19] evaluate how untrained crowd workers (crowdsourcing) can scale chatbots contents and still maintain coherence in the chatbot personality, affective behavior, and vocabulary.

3 THE EXPERIENCE REPORT

The project to develop an FAQ chatbot is a partnership between government and academia started in 2017 [1], and it is still ongoing. The Ministry of Citizenship decided to join the University of Brasília (UnB) to develop a FLOSS FAQ chatbot to help citizens to understand better their law of incentive to the culture and to answer the frequently asked questions of its service. However FAQ chatbot presence is growing in private sectors, it is still rare in government agencies [10].

One of the significant project requirement that it should be based on existing FLOSS, and the Ministry technical staff should handle its evolution and maintenance. The technical staff is composed mainly by software engineers, journalists, and experts in cultural law, and none of them had any prior experience in developing chatbots. Therefore, project documentation, configurations, and automation should be designed to shorten the team learning curve and to render unnecessary the need for chatbot specialists in the team. The main characteristics of this project are depicted in Table 1.

Table 1. Characteristics of our FLOSS chatbot project.

Team Members from all needed expertise	14 members
Releases in the last 12 months	14 releases
Number of Intentions	72 annotated intents
Number of Utters	147 utters
Original Size of the FAQ	35 questions

Throughout the project, we develop incrementally and employ a set of best practices based on FLOSS, agile and DevOps values, with lessons learned from previous government-academia collaboration projects [27]. We focused on facilitating reuse by the design of a modular architecture, modular corpus, and integration of highly dynamic OSS projects, up-to-date documentation, containerization, and heavy use of continuous integration tools to orchestrate several automated actions that, together, enabled the deployment pipeline.

Based on our practical experience, we present the FLOSS chatbot for FAQ e-government services, the lessons learned, and how the black-box reuse can reduce the time of new projects of FAQ chatbots to mature.

4 THE CHATBOT PROJECT

In the following sections, we present our solution and how significant decisions were made toward reuse. While throughout the project, we had experts in the teams or developed the necessary abilities in-house, this work serves as an experience report that enables black-box reuse to empower nonexperts in developing FAQ chatbots.

Full-stack Team

Choosing the most appropriated frameworks, tools, algorithms and dialogue models require expertise in fields such as machine learning, DevOps, language, User Experience (UX), project management, business strategy, among others [5].

A typical full-stack bot team can be divided into organization, development and beta test members, and each group has one particular involvement with the project (Figure 1). It is essential to have context specialist to give the necessary content and business specialists to guide towards the business goals. Also, it is crucial to have IT specialists involved.

In the present project, the team was formed by undergraduate interns, IT professionals and professors. The DevOps role is necessary to manage automation, services integration, and continuous deploy configuration. The UX Specialists will plan all the dialog flows, chatbot personality, and vocabulary. Finally, Data Scientists choose the techniques to process natural language, dialogues and calibrate the hyperparameters of such models.

Beta testers are essential to validate new features of the chatbot, provide feedback before it is deployed into production.

Although we have established this team structure, an essential requirement was to guarantee that a "traditional" team structure could do the maintenance, as depicted in Figure 1 (D),(E),(F). The following sections will detail how it was achieved with a focus on black-box reuse.

Overview

The complete overview of our project is depicted in Figure 2, and every component is OSS and integrated. It guarantees that the entire solution is on-premises, and how data is processed, stored, and distributed are transparent and customized.

Figure 2(A) represents the Communication Channel, and it is the interface where the user will send messages to the chatbot, and it will receive answers. Rocket.Chat [1] is a communication tool used. It enables a higher degree of automation, and the capability to store all conversations. Figure 2(B) is the core of the chatbot, where both natural language understanding and dialog management is performed. Rasa NLU and Rasa Core [2] are our distribution services. They implement state-of-art NLP algorithms and aim to bridge the gap between research and application. The quality of annotated intents is asserted by analyzing the dataset in Jupyter Notebooks. In Figure 2(C) is displayed the business analytics, the stack elasticsearch and kibana provide dashboards to gather all information about the users, like the most asked questions, chatbot access, timestamp, and unmapped questions, which can be interpreted by the Organization and the Chatbot Team. We did not use any creation service, and the entire bot knowledge and behavior were done directly on markdown files of the distribution services.

[1] https://rocket.chat
[2] https://rasa.com

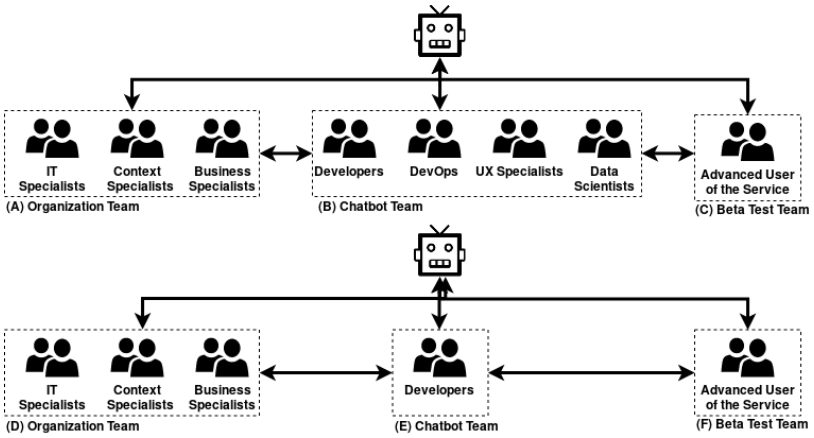

Figure 1: Chatbot team with experts (a),(b),(c) and non experts with project reuse (d),(e),(f).

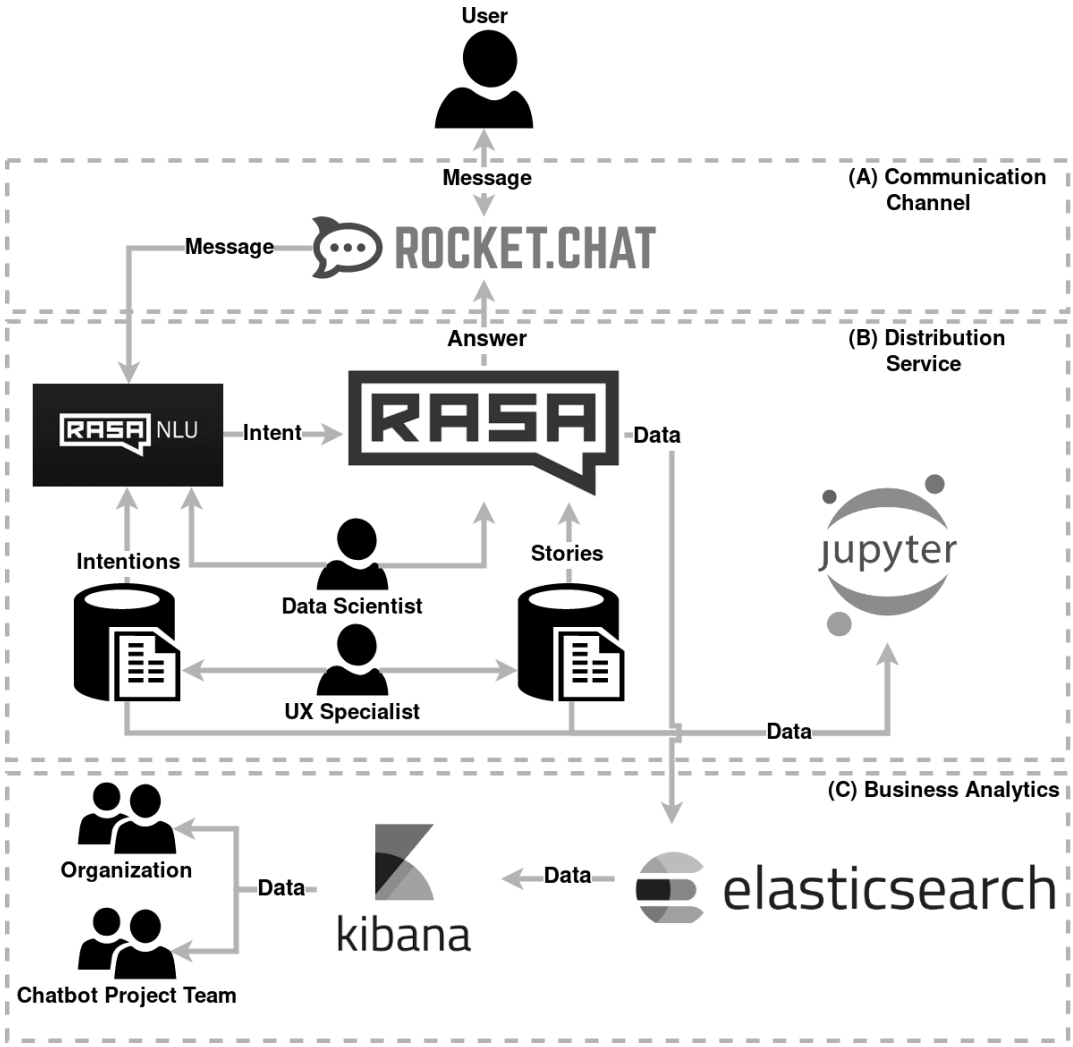

Figure 2: Project overview - technologies, artifacts, and stakeholders.

The automation of the deployment pipeline facilitates reuse of the architecture solution. We employ containerization to each service separately, and configuration files to customize production, homolog, and training environments. Finally, each component of the architecture is loosely coupled and well-encapsulated so that it can be both customized or even replaced.

Interaction Design

Designing the corpus of the chatbot knowledge base and strategy to manage the dialog is time-consuming, hand-authored by UX and interaction designer experts, which does not scale to large domains [19]. While crowdsourcing can help to overcome the problem of scale, it is only possible when it is clear the chatbot personality, vocabulary and writing style.

We addressed two main interaction design challenges: (a) a guideline of how to convert FAQ texts into annotated intents, dialog arcs and dialog flows, and (b) separate chatbot specific dataset content from general dialogue elements, such as chit-chat, general intents, fallback strategy. This later could be black-box reused in any FAQ chatbot, and interaction design would be reduced to create training data to the specific FAQ domain, following provided tutorials.

We have documented the entire interaction design process, and we create tutorials, best practices, and guidelines to write annotated intents from FAQs, suggested a personality, vocabulary, and repeated the same procedure to create dialogue contents. Finally, chatbot utterances were separated in files to explicit domain-specific content from general bot behavior.

Natural Language Understanding and Dialog Management

Chatbots are based on machine learning techniques that (a) process natural language from input message from users, (b) identify the message meaning (Natural Language Understanding), and (c) select the most appropriate next action to perform (Dialog management) [8]. The choice of machine learning techniques, their pre and post-processing impact the user experience directly. For example, it influences how well the chatbot identifies user intent on a given language, its ability to infer context, to treat noncooperative behavior, and to manage complex dialog flows [8][10]. Therefore, the proper choice of Natural Language Understanding (NLU), dialog management pipeline and hyper-parameters impose restrictions on what can be done by the chatbot.

FAQ chatbots basically must identify users intents and select the correct action to execute from a closed list. The choice of the most suitable NLU and dialog management techniques depends on the amount of dataset available for training and validation. This dataset is composed by the list of utterances annotated with intents, entities, language annotated dataset, and dialogues examples [16]. It implies that techniques execution pipeline can be borrowed across projects with similar requirements, like FAQ chatbots, as a form of generalization and reuse.

We have chosen data-driven algorithms that worked well for Brazilian Portuguese language and medium-size utterances (over 1000). For intent classification, we chose the supervised embed model Starspace [2], and we utilize annotated dialogues to train a Long short-term memory (LSTM) network [25]. This pipeline can be reused as a black-box to any FAQ chatbot in Portuguese, where the only configuration necessary is the calibration of the hyperparameters.

To validate the hypothesis of algorithms pipeline reuse, we wrote a tutorial on hyperparameters calibration, and ask a group of software engineering undergraduates, with no prior knowledge in machine learning, to develop FAQ chatbots to the University using the same pipeline. In three months, they were able to develop a chatbot and correctly calibrate the hyperparameters, guided by our tutorials.

Analytic and Improvements

According to Michaud [15], monitoring is an important step while developing a chatbot. Analytic tools track messages, distinguishes guests and users text behavior, and show in dashboards an infinity of graphics that display metrics and data from the actual chatbot use. This information enables the management of the knowledge base and guides the development efforts aligned with the business strategy.

The lesson learned from adding analytics to our chatbot project is the importance of determining the quality of the utterances and dialogue contents [15], in providing accurate data on users real questions and real behavior. This work is unfeasible manually. Metrics guide the development team to improve the quality of the interaction, and they also guide managers to draw data-driven business strategies. Table 2 summarizes some of the data provided by analytics, showing some results from our chatbot data after sixty days in the production environment.

5 DISCUSSION

Throughout this paper, we presented a report of a FLOSS FAQ chatbot project. The objective is to present some of the issues engineers, experts, managers, or researchers will likely be confronted with when developing an OSS FAQ chatbot, some core concerns they should have, and to evidence how they can benefit of black-box reuse of similar project.

Although white-box reuse is an essential practice in Open Source Projects, chatbots can benefit from black-box reuse regarding (a) machine learning models, and techniques pipeline,

Table 2. Business analytics data collected after 60 days of the chatbot in production.

Average Number of Messages sent by single user	10.2
Number of users interacting with the chatbot per day	44
Number of Default fallback occurrences in 60 days	72
Average number of timeseach user asks for help per conversation	2
Messages per day on week days	738
New questions not present in utterances	121
Number of intents added after analysing the data	14

(b) datasets to train both intent and dialog flows, (c) automation, configuration files, and (d) analytics tools and dashboards. Consequently, with proper documentation, tutorials, and guidelines, untrained workers/ nonexperts could build a mature FLOSS FAQ chatbot from scratch in a short period. Table 3 a list of assets that could be reused as black-box, and the ones that should be customized if our project is used to develop a FAQ chatbot in Portuguese in a different context than ours.

A common problem in chatbot projects is to find, to chose, and to configure the most appropriate set of technologies. In OSS, this problem is accentuated, once most projects only provide advanced technical documentation [4], which creates an adoption barrier to nonexperts. Additionally, some of the most advanced OSS frameworks cover only partially a chatbot architecture, and it imposes an integration effort which is also an adoption barrier to nonexperts. Finally, the common dialogue flows like greeting and chit chat are typically re-implemented every new application. Although it is readily available this corpus in English, the same is not correct to other languages. Therefore, all these barriers make a typical FAQ chatbot project to be time-consuming, filled with mistakes, waste, and to deploy immature chatbots in the production environment. We believe these projects errors can shadow the efficacy of FAQ chatbots, and by poor user experience lead by immature chatbots [14].

In 1996, Weizenbaum [26] called Eliza as a program, in 1997 Lieberman [12] called Letizia as a computer agent, but both Eliza and Letizia are classified today as chatbots. Nowadays, there is still some divergence to similar chatbot concepts, such as in [6] that evidences synonyms like conversational systems and virtual agents, chatbot. The lack of a common vocabulary added disturbance to this study. The discussion and definition of these concepts is an opportunity for academic contribution.

This study has a few apparent limitations. Around 5 groups of much more inexperienced teams, composed by undergraduate students, developed a chatbot to answer FAQ of the administration services at the university in 3 months, reusing ours as presented in the paper. However, this validation should be done to a broader number of FAQ projects, and data about collected to validate our hypotheses fully. Future work is needed to explore more sophisticated approaches to this problem, and compare reuse in contexts other than FAQ chatbots.

6 CONCLUSIONS

In this paper, we presented the experiment report of a FLOSS FAQ chatbot project developed in the context of e-government services. We highlight the main challenges encountered when developing a chatbot distribution service and analytics for non-English dialogue agents. Several OSS projects compose the chatbot architecture. Their selection is the first difficulty, and automation is fundamental to integrate these services and enable continuous delivery. The choice of Natural Language Understanding (NLU) and Dialogue Management techniques demands the expertise of data scientists, and we found that the OSS community gives little support to non-English applications. We configured a set of techniques that performed well for the Portuguese Language with small training datasets. We provided tutorials to facilitate the reuse and hyperparameters calibration to different contexts and scopes. These automation, customizations, and tutorials enable to use this project in other FAQ chatbots in black-box reuse.

Finally, we have outlined a set of good practices, tutorials, and documentation to empower untrained workers/ nonexperts to build a FLOSS FAQ chatbot from scratch in a short period for non-English contexts.

7 ACKNOWLEDGEMENTS

The authors are indebted to the users and collaborators of this chatbot project for providing invaluable feedback and creating a supportive ecosystem. Special acknowledgment is owed to the Lab team and external contributors. The up-to-date list of contributors may be visited at https://github.com/lappis-unb/tais/graphs/contributors.

The Brazilian Ministry of Citizenship has financially supported this research in the TAIS project. The authors are

Table 3. Black-box reuse in FLOSS FAQ Chatbot - list of assets.

Activity	Assets to be reused as black-box	Assets to be customized for each application
Interaction Design	Language corpus General dialogue contents Dialogue tutorials of how to convert FAQ file into intents and utters Chatbot personality tutorials	Write intents, utters and dialogue flows to the desired context Follow dialogue tutorials Follow personality tutorials
Architecture	Distribution service Creation service Analytics service Continuous integration Development containers Experimental environment for dataset validation Configuration tutorials Architecture customization tutorials	Follow configuration tutorials Follow architecture customization tutorials
Natural Language Understanding	Techinics pipeline to brazilian portuguese NLU hyperparameters calibration tutorial	Follow tutorial to calibrate NLU hyperparameters
Dialog Management	Dialog Management policy techniques Dialog Management policicy hiperparameters calibration tutorial	Follow tutorial to calibrate the hyperparameters of the dialogue management policy
Monitoring	Dashboards templates Dashboards customization tutorials documentation	Follow dashboards customization tutorials

grateful for the stimulating collaboration and support from colleagues and partner organization.

REFERENCES

[1] Research Advanced Laboratory of Production and Innovation in Software Engineering (LAPPIS/UnB). 2019. Tais - a FLOSS FAQ Chatbot for the Ministry of Culture. https://github.com/lappis-unb/tais

[2] Tom Bocklisch, Joey Faulker, Nick Pawlowski, and Alan Nichol. 2017. Rasa: Open Source Language Understanding and Dialogue Management. *CoRR* (12 2017).

[3] Botpress. 2019. Botpress - A Chatbot Maker & Development Framework. https://botpress.io/

[4] Noel Carroll, Lorraine Morgan, and Kieran Conboy. 2018. Examining the Impact of Adopting Inner Source Software Practices. In *Proceedings of the 14th International Symposium on Open Collaboration (OpenSym '18)*. ACM, New York, NY, USA, Article 6, 7 pages.

[5] Jessica Falk, Steven Poulakos, Mubbasir Kapadia, and Robert Sumner. 2018. PICA: Proactive Intelligent Conversational Agent for Interactive Narratives. 141–146.

[6] B Filipczyk. [n.d.]. Chapter 12 - Success and failure in improvement of knowledge delivery to customers using chatbotâĂŤresult of a case study in a Polish SME. ([n. d.]), 15.

[7] Mingkun Gao, Wei Xu, and Chris Callison-Burch. 2015. Cost Optimization in Crowdsourcing Translation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2015)*. Denver, Colorado.

[8] J. Harms, P. Kucherbaev, A. Bozzon, and G. Houben. 2019. Approaches for Dialog Management in Conversational Agents. *IEEE Internet Computing* 23, 2 (March 2019), 13–22.

[9] Eric von Hippel and Georg von Krogh. 2003. Open Source Software and the "Private-Collective" Innovation Model: Issues for Organization Science. *Organization Science* 14, 2 (March 2003), 209–223.

[10] P. Kucherbaev, A. Bozzon, and G. Houben. 2018. Human-Aided Bots. *IEEE Internet Computing* 22, 6 (Nov 2018), 36–43.

[11] C. Lebeuf, M. Storey, and A. Zagalsky. 2018. Software Bots. *IEEE Software* 35, 1 (January/February 2018), 18–23.

[12] Henry Lieberman. 1997. Autonomous interface agents. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '97*. ACM Press, Atlanta, Georgia, United States, 67–74.

[13] Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. 2019. Benchmarking Natural Language Understanding Services for building Conversational Agents. *arXiv* 1903.05566 (mar 2019), 13.

[14] R Lowe, N Pow, Iulian Serban, L Charlin, C.-W Liu, and J Pineau. 2017. Training end-to-end dialogue systems with the Ubuntu Dialogue Corpus. *Dialogue and Discourse* 8 (01 2017), 31–65.

[15] L. N. Michaud. 2018. Observations of a New Chatbot: Drawing Conclusions from Early Interactions with Users. *IT Professional* 20, 5 (Sep. 2018), 40–47.

[16] Alexander H. Miller, Will Feng, Adam Fisch, Jiasen Lu, Dhruv Batra, Antoine Bordes, Devi Parikh, and Jason Weston. 2017. ParlAI: A Dialog Research Software Platform. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (2017).

[17] Mycroft. 2019. Mycroft AI Open Source Voice Assistant. https://mycroft.ai

[18] M. Nuruzzaman and O. K. Hussain. 2018. A Survey on Chatbot Implementation in Customer Service Industry through Deep Neural Networks. In *2018 IEEE 15th International Conference on e-Business Engineering (ICEBE)*. 54–61.

[19] Maike Paetzel, James Kennedy, Ginevra Castellano, and Jill Lehman. 2018. Incremental Acquisition and Reuse of Multimodal Affective Behaviors in a Conversational Agent. In *International Conference on Human-Agent Interaction 2018*. 92–100.

[20] Maria-Eleni Paschali, Apostolos Ampatzoglou, Stamatia Bibi, Alexander Chatzigeorgiou, and Ioannis Stamelos. 2017. Reusability of open source software across domains: A case study. *Journal of Systems and Software* 134 (2017), 211 – 227.

[21] S. Perez-Soler, E. Guerra, and J. de Lara. 2018. Collaborative Modeling and Group Decision Making Using Chatbots in Social Networks. *IEEE Software* 35, 6 (November/December 2018), 48–54.

[22] Ruben Prieto-Diaz. 1993. Status Report: Software Reusability. *Software, IEEE* 10 (06 1993), 61 – 66.

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

[24] Navid Tavanapour and Eva A C Bittner. 2018. Automated Facilitation for Idea Platforms: Design and Evaluation of a Chatbot Prototype. *Conference: Thirty Ninth International Conference on Information Systems (ICIS)* (2018), 9.

[25] Vladimir Vlasov, Akela Drissner-Schmid, and Alan Nichol. 2018. Few-Shot Generalization Across Dialogue Tasks. *CoRR* (2018).

[26] Joseph Weizenbaum. 1966. ELIZA&Mdash;a Computer Program for the Study of Natural Language Communication Between Man and Machine. *Commun. ACM* 9, 1 (Jan. 1966), 36–45.

[27] Melissa Wen, Paulo Meirelles, Rodrigo Siqueira, and Fabio Kon. 2018. FLOSS Project Management in Government-Academia Collaboration. In *International Conference on Open Source Systems*. 15–25.

[28] Mairieli Wessel, Bruno Mendes de Souza, Igor Steinmacher, Igor S. Wiese, Ivanilton Polato, Ana Paula Chaves, and Marco A. Gerosa. 2018. The Power of Bots: Characterizing and Understanding Bots in OSS Projects. *Proc. ACM Hum.-Comput. Interact.* 2, CSCW, Article 182 (Nov. 2018), 19 pages.