Licy: A Chatbot Assistant to Better Understand and Select Open Source Software Licenses

Giorgos Shittas, Georgia M. Kapitsaki[a] and Maria Papoutsoglou[b]
Department of Computer Science, University of Cyprus, Aglantzia, Cyprus

Keywords: Open Source Software Licenses, Chatbot, License Recommendations.

Abstract: Open Source Software (OSS) carries licenses that specify the terms under which the software is made available for use. Various resources are available for software engineers online, in order to assist them in understanding and choosing among the available OSS licenses when creating their software projects. However, these resources lack in the provision of a sense of interactivity to user prompts, which would have been useful for providing guidance in a more familiar manner. In this work, we present our approach for *Licy*, a chatbot OSS licensing assistant for guiding users with information on specific OSS licenses and for choosing which OSS licenses to use in specific cases. A large number of licenses are supported by the chatbot using the license model offered by *choosealicense*, focusing on license permissions, limitations and conditions. We describe the design and implementation process of the chatbot and its preliminary evaluation results using chatbot design metrics and a user evaluation. We argue that the chatbot can serve as a starting point for similar interactive assistants for software engineers, and describe its value in that respect.

1 INTRODUCTION

Open Source Software (OSS) is widely used, while many software systems are made available as OSS, even by large vendors. OSS licenses specify the terms under which the original software is available for use by the licensees, the users of the software, and how it can be further modified and distributed. The OSS licensing landscape is complex, as it contains a wide variety of licenses that are linked with different combinations of permissions, constraints, and requirements (Laurent, 2004). A very large number of OSS licenses exists nowadays, with Software Package Data Exchange (SPDX)[1] listing 634 OSS and content licenses including their different versions (Stewart et al., 2010). Even among the more limited number of popular licenses, understanding the license content and the context of use is not trivial. The OSS licenses that are the most dominant today, according to Statista,[2] are the GNU General Public License (GPL), the Apache License, and the MIT License.].

Although there are available resources for OSS licenses online, e.g. *TLDRLegal*[3] and *choosealicense*[4], and prior research works that have focused on license recommendations and guidance (Kapitsaki and Charalambous, 2019; Kapitsaki et al., 2022; Xu et al., 2023), there is a lack of tools that are interactive when it comes to understanding and selecting OSS licenses.

In this work, we use as starting point the above to explore the potential of chatbots as a solution to the issue of OSS licensing assistants. Through simulating human-like conversations, chatbots deliver instant responses, and automate functions that once required a human touch, improving the user experience (Lokman and Ameedeen, 2019). We are introducing the design and implementation of an OSS licensing assistant chatbot, *Licy*, that aims at simplifying concepts found in OSS licenses and at giving users a conversational environment for finding the license that is most appropriate for their software project. The chatbot supports 58 licenses, and its prototype implementation has been evaluated using chatbot design metrics from a prior publication (Cañizares et al., 2022) and via the participation of 18 users with limited experience in OSS. The results of this evaluation show that *Licy* is straightforward to use, while it has the poten-

[a] https://orcid.org/0000-0003-3742-7123
[b] https://orcid.org/0000-0003-0658-5065

[1] https://spdx.org/licenses/
[2] https://www.statista.com/statistics/1245643/worldwide-leading-open-source-licenses/
[3] https://www.tldrlegal.com/
[4] https://choosealicense.com/

Shittas, G., Kapitsaki, G. M. and Papoutsoglou, M.
Licy: A Chatbot Assistant to Better Understand and Select Open Source Software Licenses.
DOI: 10.5220/0013336400003928

In *Proceedings of the 20th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2025)*, pages 573-581
ISBN: 978-989-758-742-9; ISSN: 2184-4895

tial to be expanded with additional features.

We argue that this work can serve as a starting point for similar conversational agents in the area of OSS licensing, especially for educational pusposes. This interactive approach allows the chatbot to ask specific questions, clarify users' intentions, and give recommendations, with which the user can interact in order to get more information on specific responses provided by *Licy*.

The remainder of the text is structured as follows. Section 2 presents OSS licenses and related work in the area. Section 3 is dedicated to the presentation of the chatbot design and implementation. The evaluation performed is detailed in section 4. Findings are further discussed in section 5, and section 6 concludes the work outlining directions of future work.

2 BACKGROUND AND RELATED WORK

2.1 Open Source Software Licenses

The essence of software licensing is to protect the intellectual property rights of developers, as well as to specify the rights and restrictions given to licensees. Software licenses range in general from strict proprietary to permissive OSS licenses, each tailored to individual goals. When using software that is made available under an OSS license, the user has the opportunity to add enhancements and distribute the modifications made, usually under the same licensing conditions as the original software. Understanding and being compliant with OSS licenses is vital in respecting the rights of software creators. It also makes users of software conscious of any limitations the software they are using has, like prohibitions to use derivative works for commercial purposes or conditions to share modifications under the same license. Unlawful use of software resulting from the lack of indicating any license or from the violation of licensing terms can cause legal disputes, so proper selection and use of OSS licenses is vital (Reddy, 2009). OSS licenses are divided in the following main categories:

- **Permissive Licenses:** Licenses in this category do not impose tight restrictions on how software can be used, modified or distributed. Usually, they require that certain conditions are met when the software is redistributed, such as inclusion of the original license notices, copyrights and trademarks. Examples of permissive licenses include the Apache license, the BSD (Berkeley Source Distribution) licenses and the MIT License.
- **Weak Copyleft Licenses:** Licenses in this category allow users to choose any licensing model for their own (added) code provided that the original software is not modified. LGPL (Lesser General Public License) licenses and MPL-2.0 (Mozilla Public License 2.0) are the most typical examples of this category.
- **Strong Copyleft Licenses:** They guarantee that any distributed version or modification of the licensed original software, or any project that uses it, will be made available under the same licensing terms. This means that any software product that employs source code licensed under a strong copyleft license also needs to carry a strong copyleft license, with examples including GPL and AGPL (GNU Affero General Public License).

2.2 OSS Licensing Assistants

Prior works have focused on assisting users in choosing the most appropriate license for their software. *findOSSLicense* is a recommender system that assists developers in selecting an appropriate license for their software project (Kapitsaki and Charalambous, 2019; Kapitsaki et al., 2022). The users can indicate a number of requirements, resulting from their responses to specific questions, including rights they want to provide to users of the software, other licenses they are already using in libraries in their source code, and technical properties of the software, whereas licenses recommended to other users of the recommender system are also considered, when making recommendations. License compatibilities are also taken into consideration using FOSSology for scanning the source code (Gobeille, 2008). *LicenseRec* is a recommendation tool that is based on compatibility checks on OSS projects' code and dependencies (Xu et al., 2023). It uses a compatibility matrix for compatibilities between licenses, whereas the source code is scanned via the Scancode[5] tool to find licenses in project files.

When it comes to license terms, *OSLiFe-DiSC* is a tool that helps OSS developers in understanding open source licenses terms (Sassi, 2024). *OSLiFe-DiSC* provides users the list of OSS licenses approved by the Open Source Initiative (OSI),[6] it allows users to identify which licenses support a specific set of features, while it can make comparisons between two licenses. At the same time, online resources provide more information on the terms of licenses. The aforementioned *TLDRLegal* is based on legal analysis of licenses and presents what the licensee can, cannot

[5]https://github.com/aboutcode-org/scancode-toolkit
[6]https://opensource.org/

and must do with each license, while *choosealicense* offered by GitHub provides a basic set of licenses the user can choose from based on his/her needs and also lists the main terms of a large number of licenses.

2.3 Chatbot Applications

Chatbots have been used in various areas, including customer service, healthcare and education. In customer service, chatbots are employed to respond to queries, solve problems, and lead users through troubleshooting, such as in the case of social media, where a chatbot trained using Twitter conversations between users and agents was created (Xu et al., 2017). Within healthcare, chatbots help with appointments, schedules and patient education (Shinde et al., 2021). In education, chatbots can act as virtual tutors and learning assistants, providing students with interactive learning environments (Clarizia et al., 2018). Chatbots have also been useful for scientific applications, such as for querying multidimensional big data (Franciscatto et al., 2022), and for software engineering (Lebeuf et al., 2017), whereas the wide spread of Large Language Models (LLMs) have offered new directions for integration with chatbots (Dam et al., 2024). In the area of software licenses, Microsoft License Advisor[7] is a custom version of GPT-4 that assists users with advice on Microsoft licenses.

Relation to Previous Works. Existing online resources and platforms on OSS licensing offer a static user experience, missing the interactive aspect, which is available with a chatbot interface targeted in the framework of the current work. The conversational nature and variety of licenses supported in the current work make the chatbot an enhancement upon static approaches to OSS licensing assistance, with *Licy* building upon the existing static resources, e.g. the licensing terms of *choosealicense* have been used for license modeling purposes.

3 DESIGN AND IMPLEMENTATION

3.1 License Modeling

The chatbot supports the licenses listed in Table 1. The abbreviation of the license name follows the naming convention of SPDX (Stewart et al., 2010) that has been utilized in prior works on OSS licensing and are used this way in the text (Kapitsaki and Kramer, 2014). Although not intended for software licensing, we have also added the following Creative Commons licenses due to their popularity, as they are also licenses that gather the attention of developers and are listed by SPDX: CC0-1.0, CC-BY-4.0, CC-BY-NC, CC-BY-ND-4.0, CC-BY-SA-4.0, CC-NC-ND, CC-NC-SA, CC-ND, CC-SA. For the same reasons, we have also added the content licenses GFDL-1.3 (GNU Free Documentation License v1.3) and OFL-1.1 (SIL Open Font License 1.1) (Lin et al., 2009).

For each license, we adopted the license terms available in *choosealicense* in order to be able to present to the user what the licensee is allowed to do (can or permissions), what he/she is not allowed to do (cannot or limitations) and which are his/her obligations (must or conditions). These terms are also used when recommending licenses to users. An example for MPL-2.0 license as provided by *choosealicense* is shown in Table 2.

3.2 Overall Chatbot Workflow

We have chosen to design *Licy* as a conversation-centric chatbot that mimics human discussions, as this type of chatbots are considered better in natural conversations (compared to other types of chatbots) (Moore and Arar, 2018). The functionality of the chatbot concerning licensing has as goals to:

1. Simplify the license selection process for users by recommending appropriate licenses.
2. Provide explanations on OSS licenses and answer relevant users' questions.

The workflow for the user was designed to cover the two above areas, as depicted in Figure 1, that includes also the Rasa[8] process stack[9] (at the bottom part of figure) that is the framework used for *Licy*'s implementation. The Rasa framework has been chosen due to its popularity and since it has an Open Source edition (Cañizares et al., 2022). After the user provides his/her input, the *intention checker* is used by the chatbot to analyze the user's input and decide on the actual intention. This is an important stage where the chatbot determines what the users intend to achieve with their query. When it comes to recommending licenses, the chatbot is intended for beginner (*Beginner Questionnaire* in the figure) or advanced users (*Advanced Questionnaire* in the figure), where beginners users are shown a more limited set of questions. If license recommendation is detected as the user's intent, these relevant questions are shown to the user (the user indicates whether he/she will be treated as a beginner or an advanced user).

[7] https://chatgpt.com/g/g-p7NpdszNq-ms-license-advisor

[8] https://rasa.com/

[9] https://kavindasenarathne94.medium.com/rasa-architecture-for-clever-chatbots-4a36b0b0ffcc

Table 1: OSS licenses covered by *Licy*.

	License abbreviation and full name
1.	0BSD (BSD Zero Clause License)
2.	AFL-3.0 (Academic Free License v3.0)
3.	AGPL-3.0 (GNU Affero General Public License v3.0)
4.	AML (Apple MIT License)
5.	Apache-2.0 (Apache License 2.0)
6.	APL-1.0 (Adaptive Public License 1.0)
7.	APSL-2.0 (Apple Public Source License 2.0)
8.	Artistic-2.0 (Artistic License 2.0)
9.	BSD-2-Clause (BSD 2-Clause ”Simplified” License)
10.	BSD-3-Clause (BSD 3-Clause ”New” or ”Revised” License)
11.	BSD-4-Clause (BSD 4-Clause ”Original” or ”Old” License)
12.	BSL-1.0 (Boost Software License 1.0)
13.	CATOSL-1.1 (Computer Associates Trusted Open Source License 1.1)
14.	CDDL-1.0 (Common Development and Distribution License 1.0)
15.	CECILL-2.1 (CeCILL Free Software License Agreement v2.1)
16.	CPL-1.0 (Common Public License 1.0)
17.	CUA-OPL-1.0 (CUA Office Public License v1.0)
18.	Cube (Cube License)
19.	ECL-2.0 (Educational Community License v2.0)
20.	EFL-1.0 (Eiffel Forum License v1.0)
21.	EFL-2.0 (Eiffel Forum License v2.0)
22.	EPL-1.0 (Eclipse Public License 1.0)
23.	ErlPL-1.1 (Erlang Public License v1.1)
24.	EUDatagrid (EU DataGrid Software License)
25.	EUPL-1.1 (European Union Public License 1.1)
26.	EUPL-1.2 (European Union Public License 1.2)
27.	GPL-2.0 (GNU General Public License v2.0 only)
28.	GPL-3.0 (GNU General Public License v3.0 only)
29.	ISC (ISC License)
30.	LGPL-2.1 (GNU Lesser General Public License v2.1 only)
31.	LGPL-3.0 (GNU Lesser General Public License v3.0 only)
32.	LPPL-1.3c (LaTeX Project Public License v1.3c)
33.	MIT (MIT License)
34.	MIT-0 (MIT No Attribution)
35.	MPL-2.0 (Mozilla Public License 2.0)
36.	MS-PL (Microsoft Public License)
37.	MS-RL (Microsoft Reciprocal License)
38.	MulanPSL-2.0 (Mulan Permissive Software License, Version 2)
39.	NCSA (University of Illinois/NCSA Open Source License)
40.	ODbL-1.0 (Open Data Commons Open Database License v1.0)
41.	OSL-3.0 (Open Software License 3.0)
42.	PostgreSQL (PostgreSQL License)
43.	Unlicense (The Unlicense)
44.	UPL-1.0 (Universal Permissive License v1.0)
45.	Vim (Vim License)
46.	WTFPL (Do What The F*ck You Want To Public License)
47.	Zlib (zlib License)

The questions shown to beginner users have been adopted from the main choices offered by *choosealicense*. We chose this option in order to keep the available choices for beginner users limited and simple.

Table 2: MPL-2.0 license modeling based on choosealicense terms.

Key	**Value**
title	Mozilla Public License 2.0
spdx-id	MPL-2.0
description	Permissions of this weak copyleft license are conditioned on making available source code of licensed files and modifications of those files under the same license (or in certain cases, one of the GNU licenses). Copyright and license notices must be preserved. Contributors provide an express grant of patent rights. However, a larger work using the licensed work may be distributed under different terms and without source code for files added in the larger work.
permissions	commercial-use modifications distribution patent-use private-use sublicense
conditions	disclose-source include-copyright same-license
limitations	liability trademark-use warranty

The questions shown to advanced users are more: 11 in total but based on the reply of the user to a given question, some questions might be skipped. In order to define these questions, we relied on prior works on recommending licenses, that we adapted considering the expanded list of licenses covered in the current work in relevance to *findOSSLicense* (Kapitsaki and Charalambous, 2016; Kapitsaki and Charalambous, 2019). The questions added for advanced users are listed in Table 3, whereas relevant explanations are provided to the users for each of them (via the chatbot's User Interface - UI). The questions refer to basic license terms, including whether modifications to the software are allowed (question 1), whether the use of a different compatible license instead of the license of the original software is allowed for derivative works (question 4 - for instance, this is allowed under the MIT License but not under GPL licenses), and whether interactions over a network are covered (question 7 - covering mainly AGPL licenses).

Other requests, e.g. asking for information for specific licenses, including permissions and restrictions or asking for more general aspects of OSS licenses, are processed by the chatbot. In order to answer a user's request, the chatbot performs the next steps after the user types a message. These are typical steps found in existing chatbot platforms, such as Rasa or DialogFlow:[10]

- *Input Pre-Processing.* The message of the user is preprocessed, including text normalizing procedures, such as converting to small letter cases, removal of punctuation, as well as correction of mistakes in the message.
- *Intent Recognition.* Natural Language Understanding (NLU) techniques are used, utilizing machine learning

[10] https://cloud.google.com/dialogflow

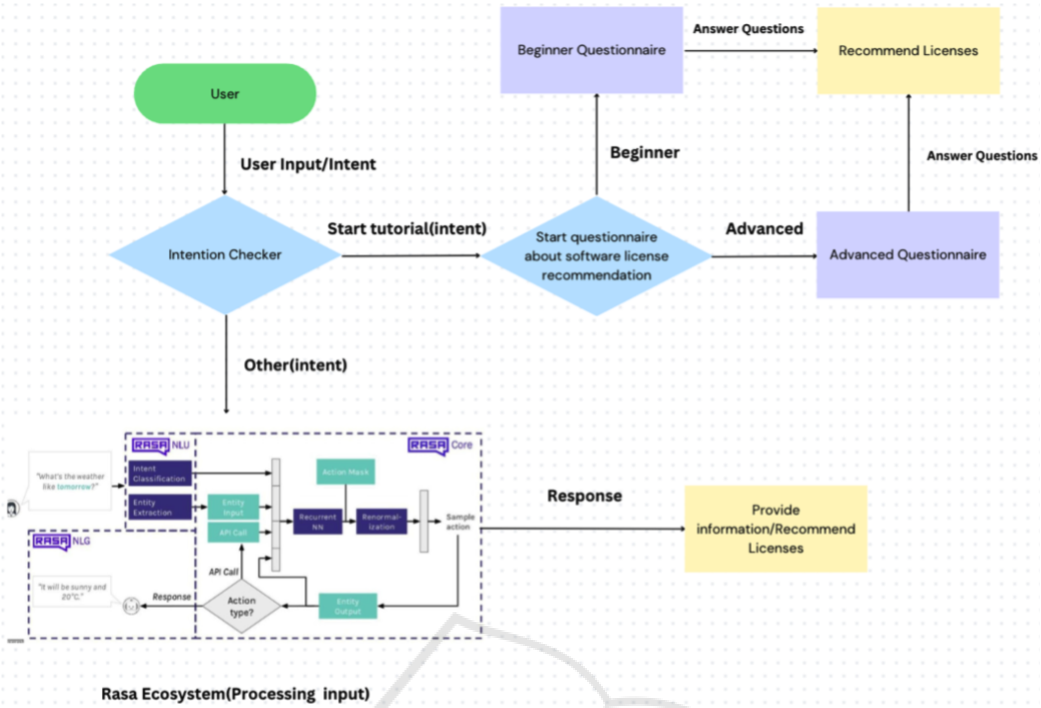

Figure 1: General workflow of the *Licy* license assistant chatbot.

Table 3: Chatbot questions for license recommendations for advanced users.

1.	Do you want the license to allow modifications to your software?
2.	Do you want to allow the licensed software and derivatives to be used for commercial purposes?
3.	Do you want to allow your software to be used in proprietary (closed-source) applications?
4.	Do you want to allow sublicensing of your software, e.g. addition of a different license?
5.	Do you want modifications to your software to be released under the same license (when distributing the licensed software)?
6.	Do you want to specify that any modifications made to your software must be indicated when redistributed?
7.	Do you want to allow users that interact with the licensed software via a network to receive a copy of the source code?
8.	Are you concerned about copyright, patents, or trademarks in your work?
9.	Do you want the original copyright to be retained?
10.	Do you want to give users the right to practice patent claims of contributors to the code?
11.	Do you want to allow the use of trademarks in your work?

models that have been trained to analyze the text and classify the user's intent.

- *Entity Extraction.* Entities are extracted from the user's message. Entities are important details and names mentioned in the user's message, like names of software licenses, license category, or other relevant information.
- *Dialogue Management.* The dialogue manager determines which response should be provided to handle user's input by considering the context of the conversation and the training data of the chatbot.
- *Action Selection.* An action to be performed next is chosen in this step. Actions can be to return a response to the user, to ask the knowledge base or to invoke a custom action. More details on the chatbot's knowledge base that drives this step are provided in the next section.
- *Response Generation.* A reply is formed using templates or responses defined in the training data. The response is customized in line with the user's request and utilizes any information which was found in the user's message.
- *Follow-Up Actions.* It is determined whether any additional actions are needed after performing an action. This is done by asking the user to provide more details or revising the user's query. For instance, the chatbot may ask the user to verify the license name he/she is referring to (to ensure it was correctly recognized by *Licy* as users may use slightly modified terminology to refer to a specific license).

3.3 Chatbot's Training Data and Knowledge Base

In order to assist the chatbot in performing the steps described previously, relevant data have been created. We have created initially, training data that can be used to understand user's intent (*Intent Recognition* step in section 3.2). The data have been created taking into consideration the type of questions that can be asked for each license, using as basis the license modeling (assuming that users will focus mainly on these aspects of the license text), and the variations in the license names that may be employed.

The above are reflected in the knowledge base of *Licy*, whose main goal is to cover a wide range of user queries in the area of license information and license recommendations. Via the knowledge base, the chatbot is able to understand the intention of user's inputs. Most chatbots are designed around intents. The license modeling has been used in order to create the possible intents, combining different license names (and variations) with what the license terms indicate. Some examples of potential intents of the user when asking about licenses are provided next. Each case corresponds to a specific user's intent, with the text in square brackets (*[]*) corresponding to the entity's text (parameter from the user's text) and the text in parenthesis (*()*) corresponding to the entity's name, e.g. *[MIT](license_name)*. The first example provided concerns variants of user's intent to ask for information about a specific license, while the second variants of asking about a specific permission a specific license may offer. The complete list of options is available in the software repository of *Licy* (Licy, 2024). The scope of what the chatbot can understand and how it should respond to different user inputs is captured in a separate part of the configuration, so the intents and entities shown below are indicated in this configuration (*domain.yml* in Rasa framework):

```
-intent: ask_for_license
  examples: |
   -What do you know about [AGPL-3.0](license_name)
   -Give me some info on [Vim](license_name)
   -Give me some info on [CC-SA](license_name)
   -Give me some info on [EUDatagrid](license_name)
   -I want to know more about [Unlicense](license_name)
   -I want to know more about [EPL-1.0](license_name)
   -I want to know more about [LPPL-1.3c](license_name)
   -What is [CC-BY-4.0](license_name)
   -Have you heard about [EUPL-1.1](license_name)
   -[Artistic-2.0](license_name)
   -[BSD-3-Clause](license_name)
   [...]
-intent: ask_for_license_permission
  examples: |
   -Does [CPL-1.0](license_name) [allow](choice)
   [private-use](permission)?
   -Is [sublicensing](permission) [allowed](choice)
   for the [MIT](license_name)?
   -Does [GPL-2.0](license_name) [permit](choice)
   [modifications](permission) to the source code?
   -Does [APSL-2.0](license_name) [let](choice) the
   licensed material and derivatives to be used for
   [commercial purposes](permission)?
   -[Can](choice) I [modify](permission) the source code
   of [CC-BY-NC](license_name)
   -[Can](choice) I use [commercially](permission)
   the software with the [Eclipse Public License 1.0]
   (license_name)?
   -Does [CC-BY-NC](license_name) [forbid](choice)
   [sublicense](permission)?
   -[EUPL-1.2](license_name) [deny](choice)
   [commercial-use](permission)?
   [...]
```

In addition to intents, flows are used to define example conversations that a user might have with the chatbot. These flows guide the chatbot on how to respond to different user inputs based on the sequence of intents and actions. Essentially, they serve as training data for the chatbot, helping it to learn how to manage dialogues by predicting the next action based on the current state of the conversation (*Dialogue Management* and *Action Selection* steps in section 3.2). Each flow represents a potential path that a conversation can take, capturing various user interactions and the chatbot's responses. By providing example conversations (or stories in the terminology of Rasa), the Rasa framework trains its dialogue model to understand and predict the appropriate actions to take during real user interactions. Each step in a flow includes an intent that represents the user's message or action. Some example stories (i.e. flows) used in the chatbot are listed below (as for intents, the whole list is available in the online repository of *Licy*) (Licy, 2024). The *action* defines the chatbot's response to that intent, which can be a simple reply, a custom action, or any other predefined response. The names of the intents indicated correspond to intents already defined in the knowledge base, as in the previous intent examples:

```
-story: Ask for software license definition
  steps:
    -intent: ask_for_license_definition
    -action: utter_license_definition
-story: User agrees on license
  steps:
    -intent: ask_for_license
    -action: action_get_license
    -intent: affirm
    -action: action_get_license_info
-story: User asks for license permissions
  steps:
    -intent: ask_for_license_permission
    -action: action_check_permission
```

3.4 Implementation Details

As aforementioned, the Rasa framework was employed for implementation purposes, that uses Python, markdown and YAML. The intents examples are used by the NLU algorithms of Rasa, and are defined in the respective file (*nlu.yml)* with the training data that assist Rasa in identifying users' intents. Flows in the conversation are defined in the *stories.yml* file. The machine learning models present in Rasa are trained using the knowledge base on how to comprehend human language. For implementation purposes, all license terms (for the license modeling of section 3.1) were encoded in JSON – JavaScript Object Notation – format. In addition, the Firebase[11] cloud database was used for data storage purposes, and Flask[12] Python web framework was used for back-end routing. Standard web technologies were employed for the front-end implementation, along with a number of Python libraries used for NLU and Natural Language Processing (NLP), e.g. `PyPI`.

For demonstration purposes, two screenshots from the prototype of *Licy* are depicted in Figure 2 and Figure 3, with the first showing a case of asking information on a specific license and the second triggering the questionnaire on license recommendations. From the first example, the chatbot extracts two main parameters: the license name (*GPL2*) and what the user is interested in (*info*). The chatbot provides also the possibility to display more information on the licenses, including relevant links (e.g. *TLDRLegal* website).

Chatbot Availability. The chatbot source code and instructions for its installation are available in a software repository online (Licy, 2024). The complete knowledge base of

[11] https://firebase.google.com/

[12] https://flask.palletsprojects.com/en/3.0.x/

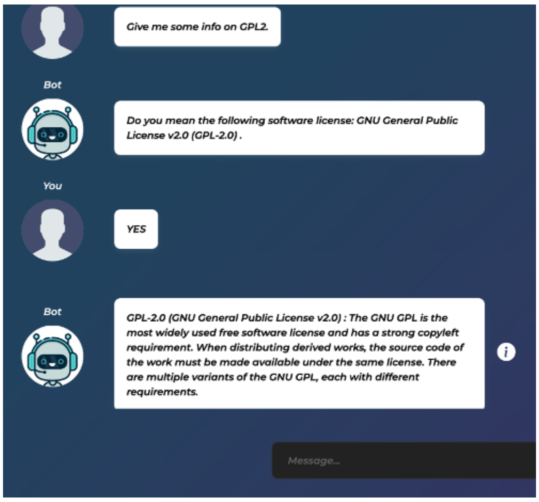

Figure 2: User asking *Licy* for a specific license.

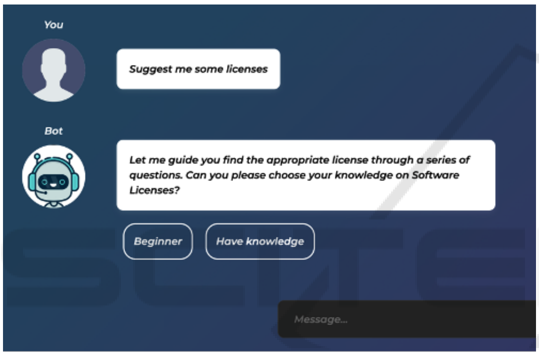

Figure 3: User asking *Licy* to recommend a license.

the chatbot as detailed in section 3.3 is also available in the repository (Licy, 2024).

4 EVALUATION OF *LICY*

4.1 Metrics Calculation

As a first evaluation step, we used the basic chatbot metrics introduced in a prior work (excluding the ones that require external tools), in order to get an overview of the general design of *Licy* (Cañizares et al., 2022). For this purpose, we have transformed the main configuration files of Rasa (*config.yml*, *nlu.yml*, *stories.yml* and *domain.yml*) to the CONGA (ChatbOt modelliNg lanGuAge) chatbot neutral notation (Pérez-Soler et al., 2020).

The results with the existing metrics are listed in Table 4. Comparing the values with the metrics of the chatbot dataset used in (Cañizares et al., 2022), it can be seen that *Licy* covers a wide range of options: there are 14 different entities (with the basic ones concerning license information and recommendations), a large number of training phrases and words per training phrase are used in intents (only *Dining-Out* chatbot has a higher number of TPI than *Licy* and only *googleChallenge* a higher number of WPTP among the 12 chatbot designs compared in (Cañizares et al., 2022)), and the characters per output phrase are also high (only *FAQ-RASA-NLU* in (Cañizares et al., 2022) has a higher number). The conversation length (CL) is relatively small, with the chatbot asking the user usually to confirm the license he/she is referring to, when asking for specific license information (as in the examples in section 3.3).

Table 4: Basic metrics calculation for *Licy* (Cañizares et al., 2022).

Name	Description	Value
INT	# intents	14
ENT	# user-defined entities	0
FLOW	# conversation entry points	12
PATH	# different conversation flow paths	12
Intent metrics		
TPI	# training phrases per intent	70.07
WPTP	# words per training phrase	5.43
PPTP	# parameters per training phrase	0.71
CPOP	# characters per output phrase	121.60
Flow metrics		
FACT	# actions per flow	1.33
FPATH	# conversation flow paths	1.00
CL	conversation length	3

4.2 User Evaluation Design

In order to evaluate the usefulness and the overall experience of *Licy*, a small scale user evaluation was performed. Questions introduced focused on assessing the visual appeal of the user interface, the user-friendliness of the chatbot, and the overall ease of use. The questionnaire consists of three main parts: 1) the first part asks users about their existing knowledge on OSS and OSS licenses, 2) the second part includes usability questions on the chatbot (e.g. how easy it was to use it), 3) the final part asks whether the chatbot was useful and their opinion on whether the information provided is accurate. Users were also asked to indicate any additional comments they might have, while access to a chatbot deployment was provided so that users could interact with the chatbot before answering the questionnaire.

In order to recruit participants, emails were sent to Computer Science students and researchers within University of Cyprus. Users were informed that no personal data would collected, and that any data used for research purposes would not lead to the participant in any way. In order to answer the questionnaire, participants gave their consent.

4.3 User Evaluation Results

18 individuals participated in the study, with all of them being Computer Science students or researchers. Regarding prior knowledge of OSS and OSS licenses (first part of questionnaire), most users have intermediate knowledge of OSS and beginner knowledge of OSS licenses (Figures 4 and 5). Concerning usability aspects, participants found *Licy*'s interface user friendly (61.1% found it very user friendly and 33.3% user friendly). They also found buttons and icons clear and easy to understand (94.4% of participants), with only one participant (5.6%) commenting that "*the close button for the about us screen was not clear.*"

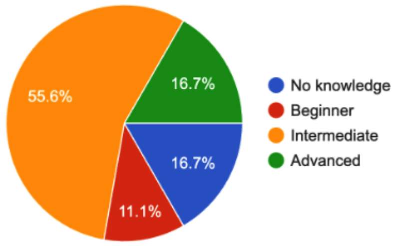

Figure 4: Participants knowledge on OSS.

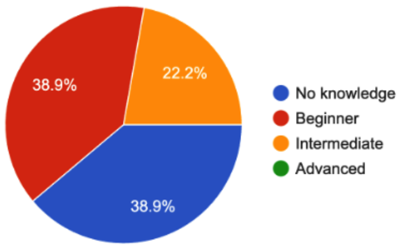

Figure 5: Participants knowledge on OSS licenses.

The *about us* screen is part of the additional information provided in the chatbot, as in the case of text with URLs pointing to additional information on the OSS licenses supported. All users also indicated that it was easy or very easy to use the chatbot and find the information they were looking for using *Licy*. Most users (88.9%) found that the chatbot understood their questions clearly and two users indicated that this happened only sometimes (11.1%).

On the use of the chatbot (third part of the questionnaire), not all users found it helpful and informative: 33.4% did not (Figure 6). Nevertheless, most said they thought the chatbot recommended the appropriate license(s) for them (82.4% replied positively). Even though the number of participants is small, we run statistical tests in order to examine tendencies in the data that larger datasets should verify. We run a Kruskal-Wallis H rank-based non-parametric test, using 1) whether the users found the chatbot helpful and informative and 2) whether they found the chatbot recommended the appropriate license as the dependent variables, and the participants prior knowledge on a) OSS and b) OSS licenses as independent variables (Vargha and Delaney, 1998). No statistically significant results were found, indicating that the prior knowledge of users on OSS and OSS licenses does not affect their experience with *Licy*. The same conclusion was drawn using as dependent variable 1) whether it was easy to use the chatbot, and 2) whether the user can easily find the information he/she is looking for.

In the participants' additional comments, where they were also asked to indicated additional features they would find useful, some useful additional features emerged, including the following: "*Provide licenses that fit on a more general idea. Like I am creating an application for logistics company what licenses should I check/need.*", "*Information such as in which cases it is useful to use a specific license or real-world problems that require this license would be helpful.*", "*Be capable to understand natural language better.*", "*I would find it useful if the chatbot responded to the question "What can I ask you and how"*." The last suggestion is already available in *Licy*, but the user might not have used a relevant phrase the chatbot can understand to answer this specific question (for training purposes, 18 alternatives were provided to *Licy*).

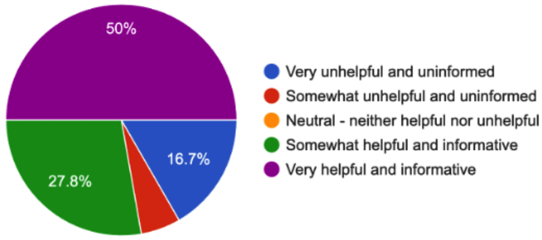

Figure 6: Participants' opinion on whether chatbot is helpful and informative.

5 DISCUSSION

For the chatbot implementation, a large number of licenses was taken into consideration (58 licenses including content licenses). Thus, the chatbot has a good coverage on licenses users might be interested in, and can guide license recommendations among all supported licenses. Concerning the users comments on additional features and better understanding of the user's intent, indeed the chatbot focuses on specific aspects – license details and license recommendations based on the employed license model – so it is expected that the users would prefer to have more options as the user evaluation showed. For novice users, it is more preferable to provide examples of licenses for specific cases, without potentially going into the details of the license terms. Nevertheless, the current implementation of *Licy* can be useful primarily for educational purposes, and can act as a guide for future and junior software engineers. Its integration in Computer Science courses or educational material for junior software engineers could be considered as a useful implication of the current work, and we argue that it may be more appropriate in this setting, considering that at the current stage we have not performed an evaluation with software engineers from the industry (Ait Baha et al., 2024). We also argue that the chatbot can be used as a starting point for creating relevant conversational assistants on OSS licensing.

Limitations. The chatbot's knowledge base is wide but it does not cover all Open Source Software licenses available today (as aforementioned, SPDX lists 634 OSS licenses). Moreover, the chatbot's license recommendations are generic and rely on the text of licenses and the specific terms as captured in the modeling of *choosealicense*, so they may not address all legal concerns a user may have. As such, the chatbot cannot be used to provide legal advice but offers guidance and recommendations that can be later verified via the help of a legal expert. User evaluation was limited to the users sampled that included students and researchers and does not include feedback and experiences from a wider population. Participants were mainly young individuals and thus, users with limited experience. The participation of senior researchers or developers from the industry may have led to different conclusions.

6 CONCLUSIONS

In this work, we have introduced *Licy*, a chatbot assistant for OSS licensing. We have described the design and implementation process of *Licy* that currently covers 58 OSS licenses. The chatbot has been evaluated with chatbot design metrics and via a small user population, showing promising results for the future. As future work, we intend to utilize LLMs to handle cases, where the chatbot is not able to reply appropriately or does not understand the text provided by the user. They have not been integrated in the current state, as we wanted to focus initially on providing dedicated training data specific to OSS licenses and their modeling. We also aim to evaluate the chatbot with developers from the industry and expand the cases addressed to include additional licenses and multi-licensing schemes.

REFERENCES

Ait Baha, T., El Hajji, M., Es-Saady, Y., and Fadili, H. (2024). The impact of educational chatbot on student learning experience. *Education and Information Technologies*, 29(8):10153–10176.

Cañizares, P. C., Pérez-Soler, S., Guerra, E., and De Lara, J. (2022). Automating the measurement of heterogeneous chatbot designs. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, pages 1491–1498.

Clarizia, F., Colace, F., Lombardi, M., Pascale, F., and Santaniello, D. (2018). Chatbot: An education support system for student. In *Cyberspace Safety and Security: 10th International Symposium, CSS 2018, Amalfi, Italy, October 29–31, 2018, Proceedings 10*, pages 291–302. Springer.

Dam, S. K., Hong, C. S., Qiao, Y., and Zhang, C. (2024). A complete survey on llm-based ai chatbots. *arXiv preprint arXiv:2406.16937*.

Franciscatto, M. H., Fabro, M. D. D., Trois, C., Cabot, J., and Gonçalves, L. A. O. (2022). Querying multidimensional big data through a chatbot system. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, pages 381–384.

Gobeille, R. (2008). The fossology project. In *Proceedings of the 2008 international working conference on Mining software repositories*, pages 47–50.

Kapitsaki, G., Paphitou, A., and Achilleos, A. (2022). Towards open source software licenses compatibility check. In *Proceedings of the 26th Pan-Hellenic Conference on Informatics*, pages 96–101.

Kapitsaki, G. M. and Charalambous, G. (2016). Find your open source license now! In *2016 23rd Asia-Pacific Software Engineering Conference (APSEC)*, pages 1–8. IEEE.

Kapitsaki, G. M. and Charalambous, G. (2019). Modeling and recommending open source licenses with findosslicense. *IEEE Transactions on Software Engineering*, 47(5):919–935.

Kapitsaki, G. M. and Kramer, F. (2014). Open source license violation check for spdx files. In *Software Reuse for Dynamic Systems in the Cloud and Beyond: 14th International Conference on Software Reuse, ICSR 2015, Miami, FL, USA, January 4-6, 2015. Proceedings 14*, pages 90–105. Springer.

Laurent, A. M. S. (2004). *Understanding open source and free software licensing: guide to navigating licensing issues in existing & new software*. " O'Reilly Media, Inc.".

Lebeuf, C., Storey, M.-A., and Zagalsky, A. (2017). Software bots. *IEEE Software*, 35(1):18–23.

Licy (2024). https://github.com/CS-UCY-SEIT-lab/Licy_License_Chatbot_V2.

Lin, K.-J., Lin, Y.-H., and Ko, T.-M. (2009). Examining open source software licenses through the creative commons licensing model. In *Software Applications: Concepts, Methodologies, Tools, and Applications*, pages 2978–2990. IGI Global.

Lokman, A. S. and Ameedeen, M. A. (2019). Modern chatbot systems: A technical review. In *Proceedings of the Future Technologies Conference (FTC) 2018: Volume 2*, pages 1012–1023. Springer.

Moore, R. J. and Arar, R. (2018). Conversational ux design: an introduction. *Studies in conversational UX design*, pages 1–16.

Pérez-Soler, S., Guerra, E., and De Lara, J. (2020). Model-driven chatbot development. In *International Conference on Conceptual Modeling*, pages 207–222. Springer.

Reddy, H. R. (2009). Jacobsen v. katzer: the federal circuit weighs in on the enforceability of free and open source software licenses. *Berkeley Tech. LJ*, 24:299.

Sassi, S. B. (2024). Oslife-disc: Open source licenses discovering, selecting and comparing. *SoftwareX*, 27:101761.

Shinde, N. V., Akhade, A., Bagad, P., Bhavsar, H., Wagh, S., and Kamble, A. (2021). Healthcare chatbot system using artificial intelligence. In *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 1–8. IEEE.

Stewart, K., Odence, P., and Rockett, E. (2010). Software package data exchange (spdx) specification. *IFOSS L. Rev.*, 2:191.

Vargha, A. and Delaney, H. D. (1998). The kruskal-wallis test and stochastic homogeneity. *Journal of Educational and behavioral Statistics*, 23(2):170–192.

Xu, A., Liu, Z., Guo, Y., Sinha, V., and Akkiraju, R. (2017). A new chatbot for customer service on social media. In *Proceedings of the 2017 CHI conference on human factors in computing systems*, pages 3506–3510.

Xu, W., Wu, X., He, R., and Zhou, M. (2023). Licenserec: Knowledge based open source license recommendation for oss projects. In *2023 IEEE/ACM 45th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, pages 180–183. IEEE.