Thesis Dissertation

DEVELOPMENT OF A CHATBOT FOR SOFTWARE LICENSE RECOMMENDATIONS

Giorgos Shittas

UNIVERSITY OF CYPRUS

DEPARTMENT OF COMPUTER SCIENCE

May 2024

UNIVERSITY OF CYPRUS

DEPARTMENT OF COMPUTER SCIENCE

Development of a Chatbot for Open-source Software License Recommendations

Giorgos Shittas

Supervisor
Dr. Georgia Kapitsaki

Thesis submitted in partial fulfilment of the requirements for the award of degree of Bachelor'sin Computer Science at University of Cyprus

May 2024

Acknowledgements

I would like to express my deepest gratitude to my supervisor Dr. Georgia Kapitsaki for her helpful supervision, encouragement and expert advice during the software licensing chatbot development process. Her active assistance and the beneficial criticism are instrumental in the process of developing this work and advancing in the sphere of software licensing. Her faith in my ability and her willingness to show her knowledge were crucial to my academic part and the accomplishment of this project.

To the professionals of software development also deserves a special thanks, to these very kind people who devoted a part of their life to answer my queries and provide me with some pieces of knowledge. Their perspectives and suggestions have enriched the level of the depth and accuracy of my study.

Special thanks also go to other colleagues who have been providing their help, both technical and in numerous discussions which have created a multitude of new concepts and approaches of my thesis.

I also have to thank my family, whose unwavering belief in my ability has always been a source of motivation. The sacrifices and inspiration they gave have not been in vain and have been the solid rock on which my determination stands.

Lastly, I should appreciate the function of the divine providence in my life. I appreciate the strength, knowledge, and opportunities that were given to me, which allowed me to attain my academic pursuits and complete this thesis.

I am not alone to do it, but everyone listed above is the result of teamwork and collaboration. Thank you very much everybody.

Abstract

This thesis is focused on creating, development, and evaluation process of the new software license chatbot that is supposed to help users in choosing the most suitable license for their needs. Due to the diversity of licensing options, users are often confused as to what license to use. The chatbot that this study presents uses a combination of natural language processing and machine learning algorithms for deciphering its users' inputs and gives them straightforward, succinct, and exact licensing recommendations.

The study starts with a detailed analysis of the current licensing routines and chatbot technologies, the failure of an easy-to-use and user-friendly way of software licensing process. The methodology focuses on the iterative design approach of the chatbot, development of its knowledge base and implementation of conversational user interface.

The body of the thesis describes the architecture of the chatbot and its decision processes which are similar to mimicking human-like behavior. Performance of the chatbot is tested strictly through the testing of users, and the result shows that there is a high level of precision and satisfaction.

Finally, the thesis dwells on the consequences of this research for the future of software licensing and possibilities of chatbots to make users experience different in complex decision-making situations. The project ends with remarks about the achievements, shortcomings, and directions for the further studies.

In summary, this thesis adds value to the field of software licensing by offering a user-friendly solution that makes the license selection process simple and hence improve user experience and compliance with licensing laws.

Table of Contents

Table of Figures

Chapter 1

Introduction

1.1 Background and Motivation

In the field of software development and specifically in open source software, the choice of the correct open source software license is a crucial decision. It controls the way open source software will be used, distributed, and modified and has serious consequences for the developers, enterprises, and end users. The software licensing landscape is a difficult one to navigate, with a wide variety of licenses (e.g. MIT, GPL licenses, LGPL licenses) and a set of permissions, constraints, and requirements. Such complexity usually causes misunderstanding and misinterpretation that lead to misuse of a software or its restriction of distribution and innovation.

This thesis is motivated by the challenges that both beginners and experts software engineers encounter while exploring the complex maze of open source software licenses while working on the creation of software. The void of a straightforward, simple to use, and user-friendly program to comprehend and select software licenses is an issue which has to be solved. The potential of a technology solution to fill this gap is vast, and chatbots are quite promising, as they are getting more advanced in NLP (Natural Language Processing) and machine learning and are widely used in the last period.

This research aims at utilizing the power of chatbot technology to simplify software licenses in general and give users an easy method of finding the license that is most appropriate for their software project. This thesis is aiming at assisting the software development community where the users are given the power to make decisions in alignment with their legal and commercial objectives and needs.

The belief that the suitable technology possesses the ability to convert intricate decision-making processes into easy and natural communication is the driver of this thesis. The building of a license chatbot software is not only academic but is a step towards a future where technology will enable vital information to be available in a manner that is accessible and actionable for all.

1.2 Problem Statement

Open source software licenses are crucial for the protection of intellectual property and the distribution of software, but most users, particularly those who are not legal experts or are novice license users, find it difficult to comprehend and select the right license for their software. The available tools and resources for choosing a license are typically either too technical or too scattered or too unfriendly for the user, which means that there is confusion and risks of non-compliance and consequently, legal and financial problems are created. The lack of user-friendly, interactive, and informative approach in software licensing does not allow users to make an intelligent choice as to how to fully benefit from a software license.

This thesis responds to the necessity for better accessible and efficient support of users in the software license selection process using a different approach via a chatbot implementation. The chatbot that is based on NLP and machine learning is being developed not only to simplify software licenses but also as a reliable and conversational tool for the user to understand the most appropriate type of licenses for their software projects.

1.3 Objectives of the research.

The goal of this research is to prototype, develop and evaluate a chatbot that aids users in understanding the nuances of open source software licensing and in choosing the best license for their software. The study is guided by the following specific objectives:

• Research of the current condition of software licensing, usual problems and obstacles, which obstacles users experience in understanding and selecting the proper license, are analyzed.

• Determine capabilities of the current chatbot technologies and natural language processing techniques that might be employed in construction of an interactive tool for software license selection.

• Design an interactive chatbot interface, which captures the visitors and communicates with them, knows what license they need and helps them make an informed decision.

• Create a knowledge base for the chatbot regarding all kinds of software license in detail.

• Using sophisticated NLP algorithms that allow the chatbot to comprehend user questions and provide the right answers.

• Do an in-depth evaluation on the chatbot on providing appropriate licensing recommendations and seamless user interaction using user testing.

• Make a significant contribution to software development community through the provision of a tool that enhances understanding and observance of software licensing principles.

Thus, the purpose of this study is to cover the process of software license selection, to accomplish these goals, familiarize the users with the different categories of licenses, and

discuss their characteristics, so they come up with reasonable and legal decisions at the software production stage.

1.4 Scope and limitations

This research is concentrated on the development of a chatbot that will aid the users to comprehend the listing of free software licenses and how to select one. The users of the chatbot are diverse and include individual developers and small as well as medium-sized companies that need help with software licensing. The study includes the following elements:

- A thorough analysis of various software license types, especially open-source.
- A chatbot that uses natural language processing to communicate with users and provide licensing recommendations.
- A user study on chatbot evaluation, accuracy, user satisfaction, and tool effectiveness.

However, the study also acknowledges certain limitations:

- The chatbot's knowledge base is wide but it may not cover all open source software licenses in existence and especially the more obscure or niche licenses.
- The chatbots recommendations are generic and may not address all legal peculiarities of each user, situation. As such, the chatbot does not provide legal advice but offers guides and recommendations.
- Nevertheless, usability testing of this research is limited to the users sampled and does not encompass feedbacks and experiences of the larger population.

- The artificial intelligence technology and algorithms of the chatbot are just a representative of the level of AI(Artificial Intelligence) and, as a consequence, cannot copy the refined understanding that a real human expert has.

It is realized that such limitations are inherent in the nature of the research and are accounted for in the interpretation of the findings. The study aims to create a solid foundation for future work, recognizing that both software licensing and chatbot technology domains are dynamic.

1.5 Structure of the thesis

The remaining of the text of the thesis has the following structure
Chapter 2 presents literature review, which is a detailed overview of the present knowledge. In this chapter, the details of software licensing are covered, the current solutions and tools are analyzed, and the different applications of chatbots are explored, conducting a gap analysis to discover the research opportunities that this current project is proposed to fulfil.

Chapter 3, named Chatbot Purpose and Significance, talks about the significant part of precise software licensing as well as that innovative role which chatbots can play in users' assistance. Also, this chapter forecasts what impact the created software license chatbot can have on the industry.

The research design and data collection and analysis methodologies are the focus of Chapter 4. It also explains the framework adopted for building the chatbot and the reasons of which certain natural language processing algorithms were preferred.

Chapter 5, Chatbot Design and Architecture, explores the inner workings of the chatbot. It lists the minimum system requirements required for effective operations, provides a workflow of the chatbot during interactions with the user, discusses the principles underlying the interface design and the backend architecture supporting the chatbot operations.

The Implementation process is discussed in Chapter 6 that details the development tools and technologies used to construct the chatbot. It also considers the development of knowledge base of the chatbot and the expected influence of use of the chatbot.

Chapter 7, Testing and Evaluation, provides a methodology utilized for testing the chatbot, metrics for evaluating its performance, and user acceptance testing process. This chapter ends with a detailed conversation on the findings that were obtained from these evaluations.

Chapter 8, the Discussion section, presents the key findings of the research and answers the question of how the chatbot performs and how it compares with existing solutions. This chapter also covers the limitations met during the project and beside this presents a possible way of further work.

Chapter 9, the Conclusion, summarizes the thesis' contributions, provides concluding remarks and suggests possible areas for future research.

The References include all academic and professional sources that have been referred to in the thesis.

Lastly, the Appendices, consist of supplementary information which includes user manuals and examples of chatbot dialogues which reinforce and supplement the primary thesis text.

Chapter 2

Literature Review

2.1 Overview of Software Licensing

The essence of software licensing is to protect the intellectual property rights of developers, as well as specify the rights and restrictions given to users. Software licenses range runs from strict proprietary licenses to more permissive open-source licenses, each tailored to individual goals and user freedoms.

Proprietary licenses, which are mainly used by commercial organizations, are made to have a high level of control of the software. These licenses usually limit the user's they have on developers,' businesses 'and end consumers 'lives. ability to copy, modify or redistribute the software and often require the payment of a license fee. These licenses are key to business models of software companies as the means of generating income and defending the product.

On the contrary, open-source licenses [4] promote a culture of partnership, a conducive atmosphere where software can be used, modified and distributed freely. With these licenses the user gets the source code and becomes empowered to adds enhancements and distribute the modification under the same license conditions. The open-source licenses that are the most dominant are the GNU General Public License (GPL), Apache License, and MIT License, each of which with its specifics and liberties.

Outside of these fundamental groups, the software licensing domain gets larger [3] with certain models like freeware for free use of the software without modification; shareware

which gives trial periods with an option to purchase; and public domain software that relinquishes all copyright and allows unrestricted use.

The entangled maze of software licenses is not only complex but also quite challenging, especially for people who are not legal specialists. The complexity is also enhanced by the fast pace of the software development lifecycle and the need to introduce new licensing models in order to cope up with new technologies and distribution models.

Clarity and understanding are paramount in software licensing since any form of misunderstanding may incur non-compliance that is subject to disputes and financial penalties. This complex background underscores the necessity for intelligent instruments that can clarify the licensing picture, allow users to make informed choices, and pick neither only a legally compliant license but also the one that suits their project's philosophy and goals.

The software licensing overview will eventually lead to the discussion of the development and refinement in this thesis of a chatbot tool that is supposed to simplify and streamline the complicated process related to the selection of software licenses for users of any kind.

2.2 A Comparative Study of Existing Solutions and Tools

The digital environment has plenty of ongoing projects and platforms developed to help users in dealing with the puzzlements of software licensing. Such websites as choosealicense.com [5] and tldrlegal.com [2] are a great value here being the sources of summaries, comparisons, and even more insights on the diversity of software licenses. The purpose of these platforms is to simplify the legal terminology to something more understandable and easier to comprehend; thus, facilitating informed decision making among the users without having to go through the complexities of legal language.

GitHub repositories are also a quality resource providing indications and community-driven discussions about software licensing. They are usually repositories with lists of

licenses with documentation and contribute to a common understanding in the software development community.

Comparison of the software license chatbot developed in this thesis to the existing solutions showed many different benefits. Current resources as GitHub repositories and sites like choosealicense.com and creativecommons.org , as mentioned above, are a treasure trove of information for software licensing. But these platforms usually offer a static user experience, missing the interactive aspect, which is available with a chatbot interface.

Although GitHub repositories are loaded with content and community contributions, they primarily serve as information repositories. Users can view license templates and discussion but is mostly self-directed where one needs to browse, search and filters information. Likewise, sites such as choosealicense.com and creativecommons.org provide built-in advice and licensing suggestions, although tailored to the users individual circumstances.

These solutions also suffer from the fact that they are rather narrow-minded, concentrating on a small group of licenses, usually the most popular or widely recognized ones. Although this may be satisfactory for some users, the system does not cater to all licensing needs, especially users with special or unique requirements.

However, the software license chatbot developed in this study offers a dynamic, dialog interactive user interface which addresses users' licensing requirements in conversation. This interactive approach also humanizes the process and allows the chatbot to ask specific questions, clarify users' intentions, and give tailored recommendations, with which the user can interact.

Additionally, the chatbot provides more types of licenses to choose, which exceed the regular licenses that are usually found in a static form. The usage of larger knowledge base allows chatbot to handle wider user cases, from standard open-source licenses to more specialized or industry-specific licenses.

This paper also highlights several benefits of developing a Software License Chatbot over general AI models like ChatGPT which was tested multiple times for comparison reasons. These advantages are outlined as follows:

Specialized Knowledge:

- Deep Expertise: The software license chatbot is designed to have extensive knowledge about various software licenses. This includes a clear and thorough analysis of each license depending on its peculiarities and features specified in the official documents.

Accuracy and Reliability:

- Structured Responses: Details about each software license are provided in a structured manner and are sourced from the original documentation of the respective licenses, which reduces the probability of mistakes or misunderstandings.

Focused Assistance:

- Context-Specific Advice: It can comprehend particular scenarios and give recommendations which will be suitable for the particular context of the user.

Efficiency:

- Quick Responses: The chatbot is designed to focus on queries that are specific to software licenses and the responses are faster than a general-purpose AI.

- Targeted Queries: Users work with an easily manageable and very specific interface that does not contain any unrelated information.

Error Reduction:

- Reduced Missteps: The specificity of the software licenses lowers the chances of errors which could be made with more global AI models, which can cover various topics and may not contain the most accurate or detailed information about licenses.

While the specialized software license chatbot excels in the aforementioned areas, it does have certain limitations. If the user's intent does not fall within one of the predefined scenarios for which the chatbot is trained, it may have difficulty providing an appropriate response. In contrast, ChatGPT, as a large language model (LLM), is better equipped to interpret natural language and can address a wide variety of questions across different domains. This makes ChatGPT more versatile in handling diverse queries outside the specific scope of software licensing.

Summarizing, comparing with existing solutions shows the chatbot benefits in being interactive, personal, and holistic software license selection tool. The conversational nature and variety of licenses available makes the chatbot an improvement upon static approaches to software licensing help.

2.3 Chatbots and Their Usage

Chatbots are becoming popular in different sectors and changing the way consumers and businesses interact with technology. They are used in various areas, which includes customer service and support with health care, finance, and education. Through simulating human-like conversations, chatbots improve user experience, deliver instant responses, and automate functions that once required a human touch.

Regarding customer service, chatbots are employed to respond to queries, solve problems, and lead users through troubleshooting, sometimes available all the time, which leads to greatly improved customer satisfaction and operational effectiveness. Within healthcare,

chatbots help with appointment schedules, patient education, and even initial diagnostics, which in turn simplifies processes and offers support to medical workers.

The financial sector uses chatbots for a range of services, such as personal banking assistance, transaction queries and financial advice, which makes banking more accessible and individual. Being part of the education realm, chatbots act as virtual tutors and learning assistants, providing students interactive learning environments and on-demand support.

The degrees of use of chatbots are even widened in the Large Language Models (LLMs) [12] integration, which represents advanced AI systems that are able to understand, and replicate text written by humans. LLMs have huge information processing capabilities and can hold much more contextual and nuanced conversations with users. The models have dramatically enhanced the natural language understanding capacities of chatbots making interaction much more sophisticated and meaningful.

The chatbot applications of LLMs involve the capability to understand complex topics and provide explanations or instructions which is quite significant. It is especially useful to such fields as legal support or technical help.

Chapter 3

Chatbot Purpose and Significance

3.1 Importance of Software Licensing

Proper software licensing is a very important part of software development and distribution process with wide range of implications for all involved parties. This module focuses on the need for proper comprehension and application of software licensing with the threats associated with its omission.

Accurate licensing is crucial for software creators and developers to preserve their intellectual property rights. It enables them to specify the ways that their work can be used, changed, and distributed, and to generate income through licensing contracts. The appropriate type of license guarantees that the developers will have their due credit and compensation and it also provides a legal platform in case of infringement to enforce their rights. [4]

In terms of businesses and organizations, proper software licensing is very important in order to comply with legal requirements. Unlawful use of software resulting from lack of license or violation of licensing terms could cause legal conflicts, monetary compensations and impairing institutional image. It is also crucial for software supply chain integrity, since unauthorized software introduces security issues and other risks.

For end users, understanding and being compliant with software licenses, is a way of respecting the rights of software creators and preventing legal problems. It also makes users conscious of any limitations or duties which software they are using has, like

prohibitions to use it for commercial purposes or conditions to share modifications under the same license. Inappropriate or misinterpreted software licensing may result in non-volitional non-compliance, legal disputes, or financial losses. It also can kill the spirit of innovation, as well as collaboration in software community when the licenses are applicable in too much rigid ways, or not as per intended use of software.

The underlying significance of correct software licensing necessitates tools that should help the users mollify the complex nature of software licenses. An informative, precise and individual licensing chatbot is a very important tool in a compliance promotion, right protection, and the health of the development and distribution of the software ecosystem.

3.2 The purpose of Chatbots in User Assistance

Chatbots [6] have become a revolutionary power in user support, changing how people approach and get answers across different areas. Their function as a provider of user support becomes increasingly important as they offer an unparalleled mix of accessibility, speed and scalability that is hard to achieve in traditional support channels. This section discusses the multi-faceted role of chatbots in user assistance and their increasing impact in improving user experiences.

One of the basic functions that chatbots provide in user assistance is instant and 24/7 help. Chatbots are on-call constantly, unlike human-operated help desks that can be limited by business hours and staffing limitations. This continuity in service allows the clients to get what they need whenever they need without unbearable delays.

In this capability, chatbots are best suited for managing repetitive inquiries that are possibly overwhelming for human support staff. Through automation of answers to frequently asked questions, chatbots relieve humans to deal with intricate and detailed problems, and as a result, to improve overall efficiency and response quality.

Another important function of chatbots is their capability to provide individual support. Utilizing user data and advanced algorithms, chatbots are capable of personalizing their

interactions for the specific needs and preferences of each user. This customized approach improves user satisfaction and improves the chances of resolving problems successfully.

In addition, the chatbots act as an interactive learning tool guiding the users through troubleshooting steps, tutorials, or complex processes in a conversational manner. This interactive characteristic is most useful for tasks such as software licensing where fine details between options are very important.

Chatbots in the software licensing domain play a role in user education and awareness regarding different types of licenses, as well as the implications of a particular type of license, and in supporting the user in making informed decisions. Through the simplification of legal jargon and the provision of clear explanations, chatbots can take out the mystique associated with software licenses hence making the information available to a wider audience.

The Large Language Models (LLMs) [12] integration adds to the abilities of chatbots, making them more proficient in comprehending and producing human-like text with a higher level of precision. This improvement in natural language processing allows chatbots to have more intelligent conversations, to comprehend a context, and to give better help.

In short, chatbots entail a multi-faceted role of user assistance. They do not only redefine the support market but also become one of the most important players in the area of software licensing. The humankind is realizing the possibilities chatbot technology has in the field of user assistance with the progress of their technology.

3.3 Expected Impact of the Software License Chatbot

The implementation of a software license chatbot is expected to affect the approaches to software licensing on different levels of the software licensing process.

To begin with, the chatbot will notably improve user comprehension on the software licenses. The chatbot will make simple the everyday complex legal language of software

licensing agreements by providing easy to understand explanations and by answering questions in real time. This means that such improved understanding will enable the user to make better decisions and in effect, select the licenses that are best suited for their needs and that the software intends for use.

In addition, the software license chatbot is expected to simplify the license selection, process. It will reduce the work of users to read long documentation or access various websites in search of what they need. On the other hand, chatbot will function as a hub of information, providing tailored suggestions upon the user's request. This lean process will save users time and mental energy as they would not have to deal with the subtleties of different licenses.

Chapter 4

Methodology

4.1 Research Design

This thesis research design is thoughtfully developed to include the overall software license chatbot development and user evaluation. This chapter describes the methodology that is used to provide an in-depth analysis of the legal framework regarding software licenses, the intricate nature of different license types, and the challenges in the determination of the right licenses by the users.

To start with, the study covers an analysis of the legal codes and rules that influence software licensing practices. This encompasses legal analysis of intellectual property law, copyright law, and certain enactments governing software usage and distribution. Knowledge of these law bases is one of the prerequisites for the chatbot to ensure that in addition to correct recommendations, the comply with the existing legal standards.

Simultaneously, the study analyses multiple types of software licenses, particularly, their specific features, benefits, and disadvantages. In this part, the classification of licenses into general categories is considered, for example, proprietary, open-source, and public domain, and then but further into subcategories with specific terms and conditions. The aim is to establish a solid knowledge base, which includes that of various licensing options as well, upon which the decision-making algorithms of the chatbot will be grounded.

This trilateral research model gives a thorough understanding of the factors affecting software licensing decisions and common issues that users face. The integrated legal research, license classification, and user difficulty analysis in the research design allow creating modern chatbots, which are aware of the practical software licensing issues. The outcome will be a user-friendly instrument, which will simplify the process of choosing a license and eliminate all the problems that are associated with it.

4.2 Data Collection and Analysis

The following are some important data gathered during the research about licenses and its special characteristics.

Public Domain: Public domain software is, in fact, free for everybody to use. It is an open source and can be integrated with other programs without any restrictions. However, organizations should be careful as these modifications may not be fully aligned with very high quality and security standards which are required for enterprise use. In addition, one must be aware of licenses that look like public domain but do not declare it openly since this may lead to legal ambiguities.

Lesser General Public License (LGPL): This license gives developers the liberty to incorporate open-source libraries into their applications, and to choose any license model for their own code. It is very convenient in that the open-source libraries remain free and accessible while the software that makes use of them could be distributed under different terms.

Permissive Licenses: Some of the licenses contained in this category also do not impose very tight restrictions on how the software can be used, modified or distributed. Usually, they require that certain conditions are met when the software is redistributed such as preservation of the original license notices, copyrights, and trademarks. Nevertheless, Apache License, BSD (Berkeley Source Distribution) and MIT licenses are perfect examples of permissive licenses that are specific in their own way.

Copyleft Licenses: Such licenses guarantee that any distributed version or modification of the licensed code or any project that uses it will come under the same licensing terms. Consequently, a situation is realized when the software is freely manipulable and redistributed but all derived distributions are obliged to comply with the terms of the original license. This means that any new product that employs the code licensed under copyleft should also be copyleft-licensed.

Proprietary Licenses: Proprietary licenses are the strictest form of software licensing and prohibit unauthorized copying, modifications, distribution of the software. These licenses have been designed to ensure level of the protection of the IP rights of the software owners, whereby it is certain that the software is used as specified by the owner.

Unlicensed software is all-pervasive across many regions and industries, and as such, it poses a continuous and multilayered problem because it betrays the integrity of intellectual property rights and fair competition. This problem not only slows down innovation and economic growth, but it also endangers cybersecurity and data integrity, thus threatening both individuals and organizations with severe risks and liabilities. Addressing this multi-faceted issue entails integrated approaches, interdisciplinary approaches, and cooperation among governments, businesses, and civil society to implement control, promote awareness, establish ethical practices, and develop the culture of intellectual property rights.

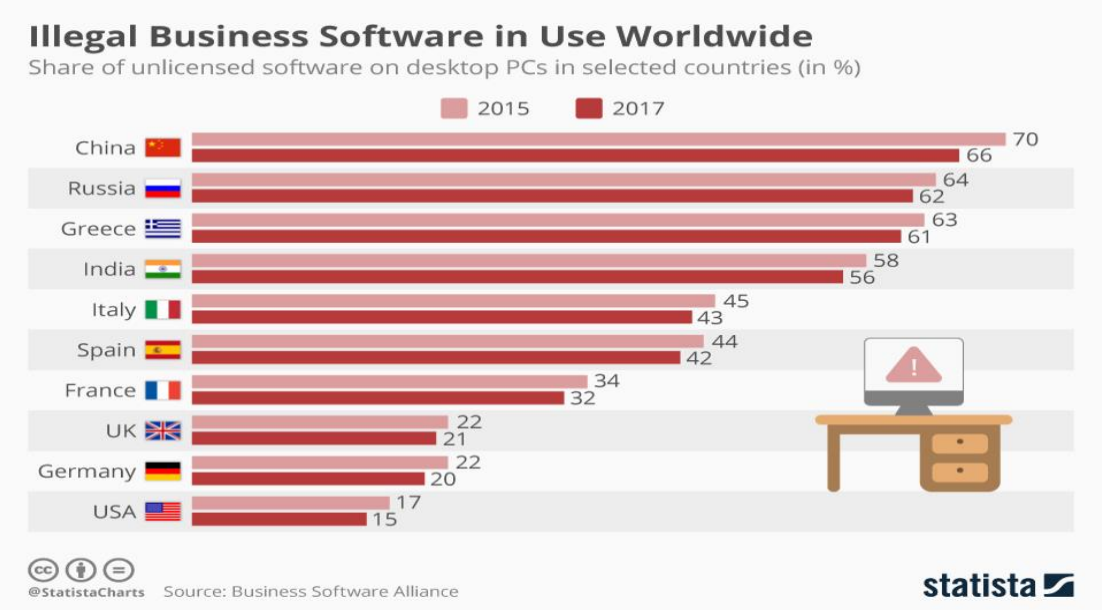


Figure 4.1: Illegal business software in use worldwide [9]

The diagram displays the percentage of new illegal software that is installed on desktop personal computers (PCs) in some of the countries in 2015 and 2017. Statistics reveal that an important share of business desktop PCs in these countries use pirated business software, 15%-70% rates. This highlights a big problem of software piracy and non-respect of the licensing policy.

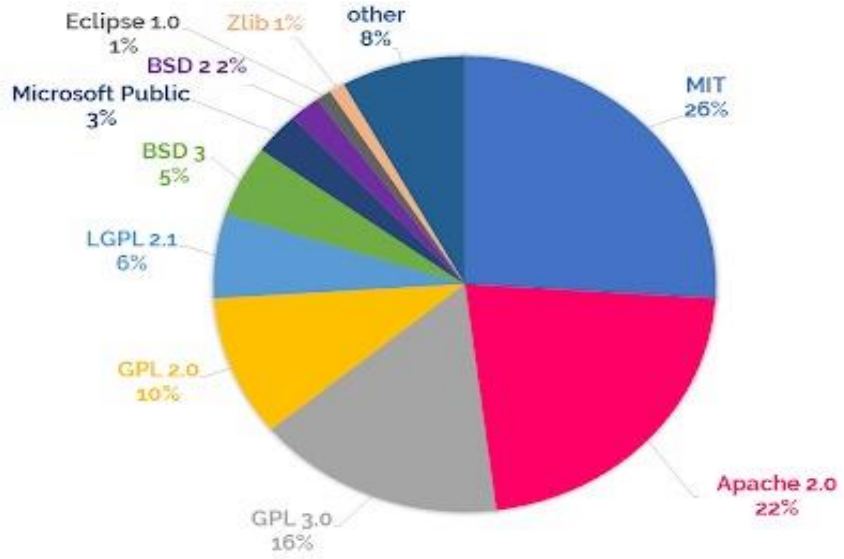


Figure 4.2: Most popular software licenses [10]

The image above offers the distribution of use the most popular software licenses [10] and from the picture, it is clear that there is a wide range of licenses that are liked by the software community. The most used is the MIT License (26%), Apache 2.0 License and GPL 3.0 (16%). The rest of the distribution includes other licenses like GPL 2.0, LGPL 2.1, BSD 3, Microsoft Public, and Eclipse 1.0 among several others.

Different license usage variety is a result of the different needs and preferences of software developers and organizations. Each license has its own set of terms and conditions, such as the open nature of the MIT and Apache licenses, which permit a wide degree of freedom in how the software can be used and modified, to the more restrictive GPL licenses, which embody copyleft principles, ensuring that derivatives remain open source.

Considering this variety, the research for this thesis was supposed to create an exhaustive set of software conditions, used by different companies, with various terms and conditions. This elaborate collection aims at easing the challenge for the users to find and choose the most suitable license for their software projects. The research makes it possible for one to make more informed choices, by offering a wider range of licenses which users can compare in order to select the one that most accurately corresponds to the goals of their project and legal requirements.

4.3 Chatbot Development Framework

A chatbot development requires choosing a framework that offers tools and features required to create, test, and distribute the bot. Python as a programming language has several frameworks, each with its own features and advantages. ChatterBot, spaCy, and Dialogflow SDK for Python are some of the popular Python frameworks for chatbot development [15].

After thoroughly reviewing the documentations of these options, examining the advantages and disadvantages of each, and assessing their suitability for the project requirements, it was determined that the Rasa framework is the most fitting choice for the development of the software license chatbot. Rasa is an open-source conversational AI

framework designed for developing chatbots requiring a high level of customization and control.

The advantages of using the Rasa framework [7] for this project are manifold:

Open-Source Nature: Rasa's open source nature enables complete transparency in the development process. It allows the possibility of browsing and modifying the code as required so that the chatbot may be adapted to the exact needs of software licenses selection.

Customization and Extensibility: Rasa provides a wide range of customization opportunities allowing for development of a chatbot suitable for particular software licensing conversations which can be complex and hard to process. It can be expanded with custom actions, integrations, and responses according to the individual requirements of the project.

Machine Learning-Based: Rasa uses machine learning to enhance the chatbot's ability to understand and later improve its responses. This functionality is essential for the correct understanding of all the various forms in which users can ask the same question about software licenses.

Contextual Conversations: The framework enables the control over conversation context, that is needed to preserve the flow of dialogue and properly follow user's preferences during the interaction.

Scalability: Rasa is built to be scalable with the app, with the number of users ranging from a few to thousands without a significant reduction in performance. This is necessary to make sure that the chatbot is able to remain responsive as it scales its user base.

Community and Support: Rasa has an active and powerful community, which offers a lot of resources, documentation, and support. This community is a great resource for help, best practices sharing, and keeping oneself updated on the latest chatbot technology progress.

Through the adoption of the Rasa framework, software license chatbot development is poised to realize these gains, leading to a strong, intelligent, and easy-to-use tool that can expertly steer users through the intricacies of software licensing.

Chapter 5

Chatbot design and architecture

5.1 System Requirements

For the best performance and effectiveness, the software license chatbot should be run on the machine supported parallel processing with Advanced Vector Extensions (AVX) instructions. [7] These instructions, the AVX instructions, give a set of the CPU instructions, the ability to execute multiple operations in one instruction hence greatly improving the power and speed of the computations particularly, numerical computing applications.

The technology that forms the basis of the chatbot, which comprises the TensorFlow [16] open-source library for machine learning, itself will get huge advantages with AVX support. TensorFlow has been developed to use such hardware features to speed mathematical calculations and these are necessary for the training of the machine learning models used by the chatbot.

The process of training machine learning models is resource-consuming, as it requires large data sets to be processed and complex calculations to be performed. Due to the use of a system with AVX instructions, the chatbot training phase is sped up, which allows for faster iterations and a more seamless development. This becomes critical as the chatbot's algorithms are fine-tuned and its knowledge base is updated to enhance precision and efficiency.

5.2 Chatbot Workflow

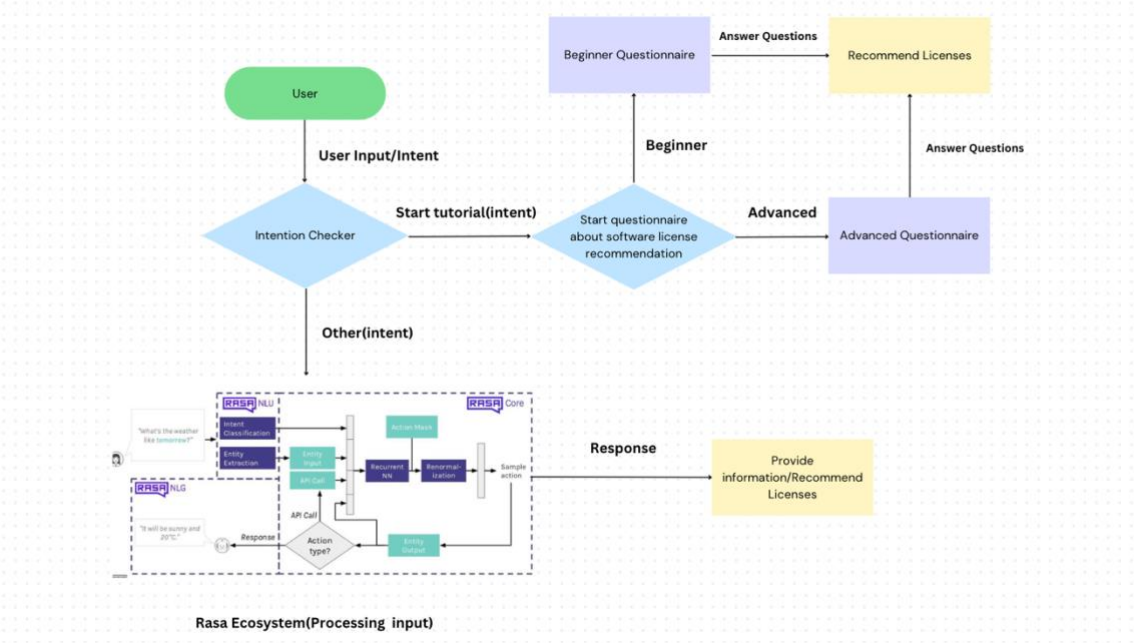


Figure 5.1: How the chatbot interacts with users and processes their queries. [8]

Here's a detailed explanation of the workflow:

1. User Interaction: The process starts when a user gives an input or expressing an intent.

2. Intention Checker: Intention checker is used by the chatbot to analyze the user's input and decide upon the actual intention. This is an important stage when chatbot determines what the users intend to achieve with their query.

3. Tutorial Intent: If the intention checker detects that the user's intention is to be guided through a series of questions for a license recommendation then the chatbot action is splitted into two types of questionnaires:

- Beginner Questionnaire: If the user is identified as a novice, the chatbot triggers a series of questions associated with beginner level questionnaire aimed to diagnose and improve his or her basic knowledge of software licensing.

- Advanced Questionnaire: On the contrary, if the user is assessed as knowledgeable, the chatbot asks an advanced questionnaire that contains more difficult issues about software licensing.

4. Other Intents: Non-tutorial intent oriented requests e.g. for particular licenses, permissions, restrictions and the like, are processed by the Rasa system.

5. Rasa Ecosystem: In the Rasa ecosystem, the user's input has several stages: [6]

- Intent Classification and Entity Extraction: The component classifies the intent of the user's message and extracts relevant entities.

- Dialogue Management: Rasa core steps in as the dialogue manager to predict the next action according to the classified intent and extracted entities.

- Custom Actions: In case the user's request needs specific information or a particular operation, custom actions of Rasa are called. They are predefined actions that the chatbot can carry out to get data, process information, or do a task.

6. Response: Rasa ecosystem processes the input and the chatbot produces a reply to the user. This may include providing license information, explaining permissions and restrictions, or suggesting licenses which suit the user's requirements.

7. Recommendation or Information Provision: The chatbot serves information depending on the user's initial question and the following communication or recommends appropriate software licenses. The last step is aimed at meeting the user's demand and aiding them in taking their decision concerning software licensing.

The workflow properly routes user inquiries to suitable processing flows, thus, making sure that the users get a precise and relevant help, whether they are answering educational questions via questionnaires or getting particular information routed via the Rasa ecosystem.

5.3 User Interface Design

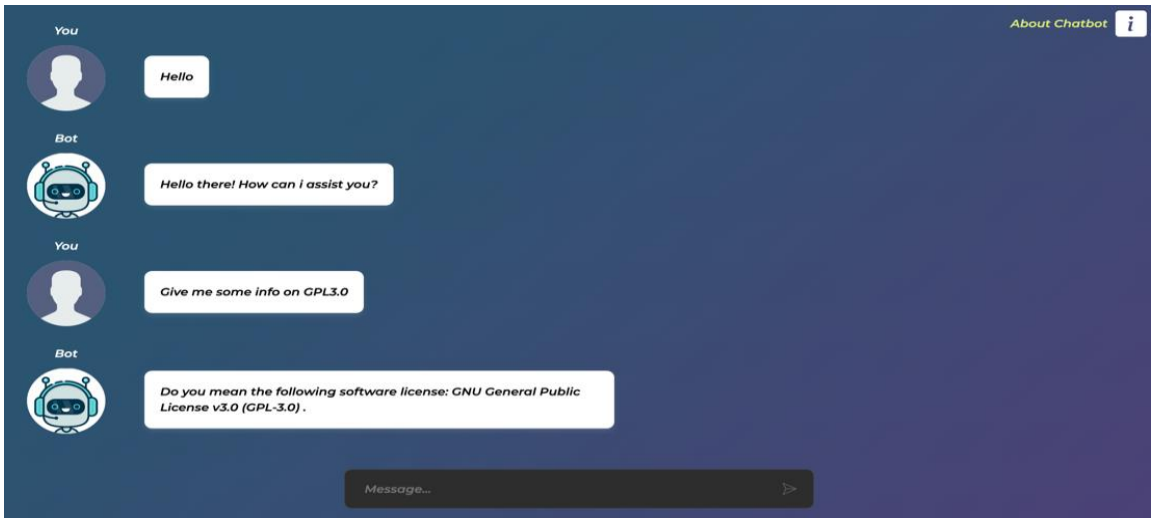


Figure 5.2: User interface (conversation)

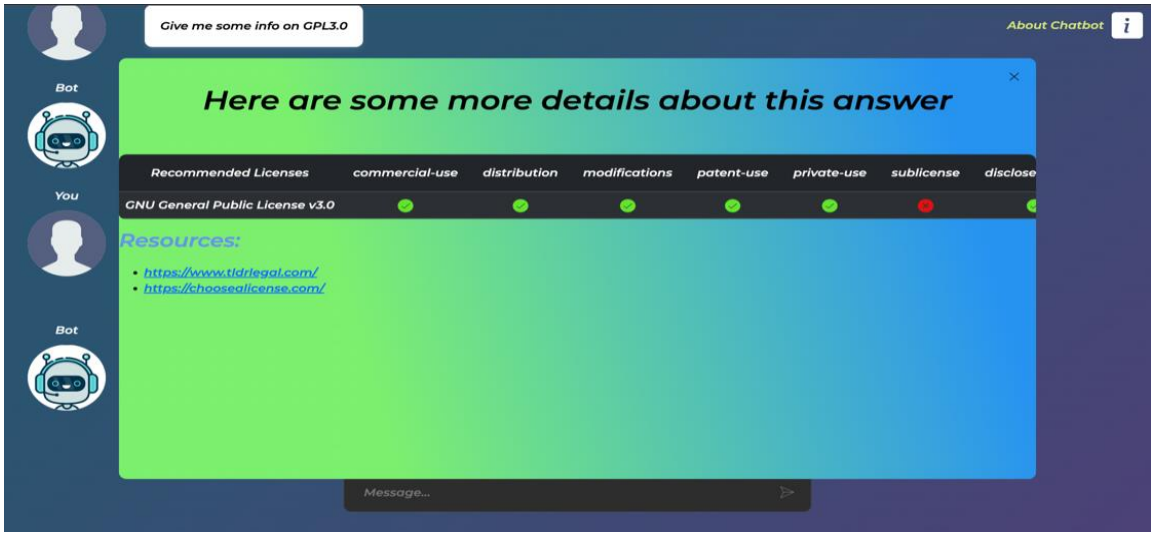


Figure 5.3: User interface (more details about an answer)

The User Interface (UI) design of the software license chatbot plays an important role as it directly influences the satiation of the user. A properly designed UI not only improves the convenience of use but also increases the user involvement and satisfaction. To this chatbot, the UI design is created to provide the environment that is friendly to the user, attractive and visually appealing. [1]

Simplicity: The chatbot interface is created to be simple, enabling even newbie users navigate with ease. The design is user-friendly, with a focus on simple points of entry for conversation and little clutter that can sidetrack or confuse users.

Color Scheme: The layout has a thoughtful color scheme that complements the interface, making it lively and friendly, but not too much. The selection of colors is meant to provide the user an enjoyable visual experience that can get them interacting with the chatbot. Color too helps to distinguish between types of interactions like questions, answers and commands.

Typography: Readability is paramount. The UI uses readable fonts with the correct size, thus, improving the general readability of chatbot's responses and instructions.

Interactive Elements: The interactive elements like buttons and sliders are integrated to give the user quick options to act and an unhindered conversation. The elements are made to be easily recognizable and available, thereby making the interaction with the chatbot more dynamic.

Responsive Design: Chabot UI is fully responsive, i.e. it works flawlessly on a variety of devices and screen sizes, starting from conventional desktops, and ending with most sophisticated smartphones or tablets. This guarantees that users get a coherent and operational experience, irrespective of their access point to the chatbot.

In general, the UI design of the software license chatbot is focused on the convenience and aesthetical appeal. The aim is to produce a pleasant and efficient exercise that does the hard work of understanding and choosing software licenses as simple and pleasurable as possible.

5.4 Backend Architecture

The backend architecture is the behind the scenes backbone of the software licensed chatbot, making sure that it is running well. It is developed with the Firebase for database purposes and Flask Python for backend services, providing a strong base for the chatbot's functional features. This part will describe the way that these technologies are used to implement a safe, scalable and responsive chatbot that can cope with complex demands of software license selection assistance.

Cloud Database - Firebase: The chatbot uses Firebase as the cloud database solution. The database in firebase is designed to allow for the effective synchronization of data between client and server applications. It houses user profiles, conversation logs, and the chatbot's software license database of dynamic knowledge. The use of Firebase makes sure that the data is controlled in a reliable, expandable, and effortlessly accessible way that allows real-time updates and an easy access of the data needed for the chatbot responses.

Backend Routing - Flask Python: [13] Flask, which is a lightweight web framework implemented in Python, takes care of the backend routing. With Flask, a chatbot is endowed with the required routing capabilities, enabling it to handle HTTP requests and responses efficiently. It acts as the link between the user activities on the frontend and the operations conducted in the backend. It is these characteristics of Flask that enable the chatbot to work with RESTful APIs used for communication with other services and the Firebase database.

Additional Flask Features: Apart from routing, Flask provides a variety of features that improve the backend functionality of the chatbot. These concerns session management and error handling.

Chapter 6

Implementation

6.1 Development Tools and Technology

The development of the software license chatbot involved a carefully selected suite of tools and technologies to ensure an efficient building process. This section enumerates the user interface development tools and technologies, as well as the backend functionalities development tools and technologies.

For the user interface, the development relied on the foundational web technologies: [14]

- HTML (HyperText Markup Language): Was the top part of the chatbot's frontend regarding the layout and the content to the user interface.

- CSS (Cascading Style Sheets): Delivered the styling and visual design parts that made the interface comfortable, bright and orange to look at. An attractive interface functionality of a chatbot was contributed by CSS.

- JavaScript: Add interactivity to the chatbot interface, which enables dynamic content updates, and responsive user interactions with no page reloads.

The choice of a frontend framework was omitted intentionally to maintain simplicity and control over the chatbot's interface with no overhead of the advanced frameworks.

In the back-end, Python was the language of choice because of its readability and wide support in the development community.

- Natural Language Understanding (NLU): Python NLU libraries [11] were used for input understanding, meaning extraction, and action decision. Such libraries allowed the chatbot to know the context and intent of user questions, which is crucial so that it would give accurate and relevant answers.

- Natural Language Processing (NLP): In addition to other NLP tools [11] , NLU was employed to parse and analyze the text, which enables the chatbot to conduct human-like dialogues.

The cooperation of these development tools and technologies resulted in a strong chatbot who provides the users a continuous journey from the frontend with to the backend. To develop a chatbot that would be convenient in its operation and be user-friendly and, at the same time, utilize the best characteristics from every tool and modern technology, a chatbot was designed.

6.2 Building the Knowledge Base

```
- 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)"
    - I want to know more about "[Unlicense](license_name)"
    - "[Artistic-2.0](license_name)"
```

Figure 6.1: Intention of the user asking for a specific license information.

```
- intent: ask_for_license_permission
  examples: |
    - Does "[CPL-1.0](license_name)" "[allow](choice)" "[private-use](permission)" ?
    - Does "[CC-BY-NC](license_name)" "[forbid](choice)" "[sublicense](permission)" ?
    - Does "[GPL-2.0](license_name)" "[permit](choice)" "[modifications](permission)" to the source code ?
    - "[EUPL-1.2](license_name)" "[deny](choice)" "[commercial-use](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) ?
```

Figure 6.2: Intention of the user asking for a permission of a specific license.

The knowledge base creation is a key part of the software license chatbot development. The main goal of creating this knowledge library was to cover a wide range of user queries. This wide coverage is vital for the chatbot performance to be able to understand the intention of the user inputs.

To reach this objective, a knowledge base was carefully built containing possible questions, statements, and commands that the users may address to the chatbot (See Figure6.1, Figure 6.2). This in-depth database is rich in content and is diverse in languages and phrasing used by various users. This variability being factored in, the chatbot will be able to identify and comprehend a wide array of user intents.

It is the knowledge base on which the NLU algorithms of Rasa act. The NLU component of Rasa queries the knowledge base as the users interact with the chatbot to associate user inputs with the most relevant intents. This is made possible by the machine learning models present in the framework which have been trained using the knowledge base on how to comprehend the subtleties of human language.

```yaml
- story: Who are you
  steps:
    - intent: who_are_you
    - action: utter_iamabot

- story: User thanks chatbot
  steps:
    - intent: thank
    - action: utter_pleasure

- story: Ask for software license definition
  steps:
    - intent: ask_for_license_definition
    - action: utter_license_definition

- story: User asks for license permissions and agrees
  steps:
    - intent: ask_for_license_permission
    - action: action_get_license
    - intent: affirm
    - action: action_check_permission
```

Figure 6.3: Some scenarios of user interaction with the chatbot inside the stories.yml

In the Rasa framework, stories (See Figure 6.3) are used to define example conversations that a user might have with the chatbot. These stories 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 learn how to manage dialogue by predicting the next action based on the current state of the conversation.

How Stories Function as Scenarios for User Interaction:

- Defining Interaction Patterns: Stories outline the flow of a conversation by specifying a sequence of user intents and corresponding chatbot actions. Each story represents a potential path that a conversation can take, capturing various user interactions and the bot's responses.

- Training the Dialogue Model: By providing these example conversations, the Rasa framework trains its dialogue model to understand and predict the appropriate actions to take during real user interactions. This enables the chatbot

to manage complex conversations dynamically and handle various scenarios effectively.

- Handling User Intents: Each step in a story includes an intent that represents the user's message or action. The chatbot uses these intents to understand what the user is asking or saying. The ‘action’ defines the chatbot's response to that intent, which can be a simple reply (utterance), a custom action, or any other predefined response.

```yaml
permission:
  type: text
  influence_conversation: false
  mappings:
    - type: from_entity
      entity: permission
      intent: ask_for_license_permission

allowed_word:
  type: list
  influence_conversation: false
  mappings:
    - type: from_entity
      entity: allowed_word
      intent: ask_for_license_suggestions

restricted_word:
  type: list
  influence_conversation: false
  mappings:
    - type: from_entity
      entity: restricted_word
      intent: ask_for_license_suggestions
```

Figure 6.4: Domain and Entities(domain.yml).

The domain.yml file (See figure 6.4) in the Rasa framework is a central configuration file that defines the chatbot's universe. It specifies important elements such as intents, entities, slots, responses, actions, and more. Essentially, this file outlines the scope of what the chatbot can understand and how it should respond to different user inputs.

Entities in the domain.yml file represent pieces of information that the chatbot extracts from user inputs. They are used to capture specific details or variables that are relevant to the conversation. For example, in a software licensing chatbot, entities might include permissions, allowed words, and restricted words, as shown in the provided image.

Entities serve several key purposes:

- Extracting Relevant Information: Entities help the chatbot identify and extract specific pieces of information from the user's messages. This information is crucial for understanding the user's intent and providing accurate responses.

- Storing and Managing Data: Entities can be mapped to slots, which store the extracted values and keep track of them throughout the conversation. This allows the chatbot to maintain context and manage data efficiently.

- Driving Custom Actions: Extracted entities are often used in custom actions to perform specific tasks or retrieve additional information. For instance, a custom action might use an extracted entity to look up license details or check permissions.

```python
class LicenseInfoProvider(Action):

    def name(self) -> Text:
        return "action_get_license_info"

    def run(
        self,
        dispatcher: CollectingDispatcher,
        tracker: Tracker,
        domain: Dict[Text, Any],
    ) -> List[Dict[Text, Any]]:
        read_licenses_info("./licensesJSON/")
        print("Confirmed License: ", tracker.get_slot("confirmed_license_name"))
        index = titles.index(tracker.get_slot("confirmed_license_name"))
        license_id = [licenses[index].id]
        license_permissions, license_titles = getLicenseInfo(license_id)
        print("License id: ", license_id)
        json_mess = {
            "key": "license_info",
            "license_id": license_id,
            "license_titles": license_titles,
            "license_permissions": license_permissions,
        }
        dispatcher.utter_message(
            text=f"{licenses[index].id} ({licenses[index].title}) : \n {licenses[index].description}",
            json_message=json_mess,
        )

        return []
```

Figure 6.5: Custom action that provides info about a license with JSON.

In the Rasa framework, custom actions are Python functions that allow the chatbot to perform specific tasks beyond the predefined responses. They enable the chatbot to interact with external systems, fetch data from databases, or execute complex logic to provide dynamic and contextually relevant responses to users. Custom actions are crucial for extending the capabilities of the chatbot, allowing it to handle more sophisticated queries and provide more detailed and accurate information. In the above exaple (See Figure 6.5) we use the entity (name of a license) extracted from the user query in order to retrieve some information about this specific license from some JSON files.

In the next part, we will investigate the sequence of events made possible by Rasa framework that use the knowledge base to receive the data and fulfil the user requests. All the mentioned actions are critical for the chatbot to deliver correct and contextually relevant answers, which allows for a smooth and informative user interaction.

6.3 Integrating Natural Language Processing

Inclusion of Natural Language Processing (NLP) into the software license chatbot is an important part of making the chatbot able to comprehend and react properly to users' inputs. The Rasa framework, on the other hand, is specialized to the chatbot's NLP functionalities and works in a series of defined steps within its ecosystem. Here's a step-by-step explanation of what happens in the Rasa ecosystem [8] when a user provides an input:

1. User Input: The interaction starts when a user is typing a message to the chatbot. This is the raw text that the user types into the chat interface.

2. Input Preprocessing: The message of the user is sent to Rasa where the text is preprocessed. This might include text normalizing procedures, such as converting to small letter cases, removal of punctuation, as well as correction of mistakes.

3. Intent Recognition: The preprocessed text is further fed to the Rasa NLU (Natural Language Understanding) component. The NLU utilizes machine learning models that have been trained to analyze the text and predict the user's intent. It identifies what the user is trying to communicate with their message.

4. Entity Extraction: In addition to intent recognition, the NLU module also does entity extraction. Entities are important details and names mentioned in the user's message, like particular names of software licenses, types, or other relevant information.

5. Dialogue Management: When we identify the intent and entities, this data is relayed to Rasa Core that acts as the dialogue manager. The dialogue manager determines what response should be made to the user's input by considering the context of the conversation and the training data of the chatbot.

6. Action Selection: Considering the current form of the dialogue and the anticipated intention, Rasa Core chooses an action to be performed. Actions can be to return a

response to the user, to ask the knowledge base or invoke a custom action which does a particular job.

7. Response Generation: When an action implies sending a message back to the user, Rasa forms a reply by the 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 from the user's message.

8. Custom Actions: For more advanced queries or operations, custom actions can be triggered by Rasa. These are built-in functions that are designed to accomplish tasks such as fetching data from a database, manipulating data, or communicating with external APIs.

9. Follow-Up Actions: The Rasa Core will evaluate if any additional actions are needed after performing an action. This can be done by asking the user to provide more details or revising the user's query.

10. Logging and Learning: In the final stage, the interaction is recorded, and this data can be utilized for more training and refinement of the NLU and dialogue management models. This ongoing learning cycle leads to improvement of the chatbot in terms of accuracy and performance over time.

The integration of NLP via the Rasa framework provides the software license chatbot with the ability to receive natural language inputs, comprehend user intents and have meaningful conversations. This coalescence is critical in developing a chatbot capable of helping users and delivering a natural, conversational feel.

Chapter 7

Testing and Evaluation

7.1 Testing and Methodology

For the testing and evaluation of the software license chatbot, the availability of the chatbot was ensured through the Virtual Private Network (VPN) of the University of Cyprus, allowing access to the chatbot via the following URL: http://10.16.90.34:5000/ . The system of this chatbot was created in such a way the testing and evaluation of the chatbot was in the university network environment, thus ensuring the safe and controlled access.

Furthermore, in conducting the user experience evaluation alongside the tests for chatbot functionality and performance, a Google Form questionnaire was employed. The questionnaire comprised inquiries aimed at assessing the visual appeal of the user interface, the user-friendliness of the chatbot, and the overall ease of use. The participants predominantly consisted of university student colleagues and some members of the academic staff. They were solicited to articulate their perspectives on the interactions with the chatbot, encompassing encountered challenges and suggestions for enhancement.

The user experience evaluation was carried out to obtain the valuable information on the usability and the efficiency of the chatbot in answering the software license related questions of the users. The feedback which was collected from the questionnaire responses was analyzed to see the strengths and the areas for improvement in the chatbot's design and functionality. This testing process was an excellent source of data for the

chatbot's performance evaluation and for the enhancement of its user interface which in turn, made the user experience better.

7.2 Results and Discussion

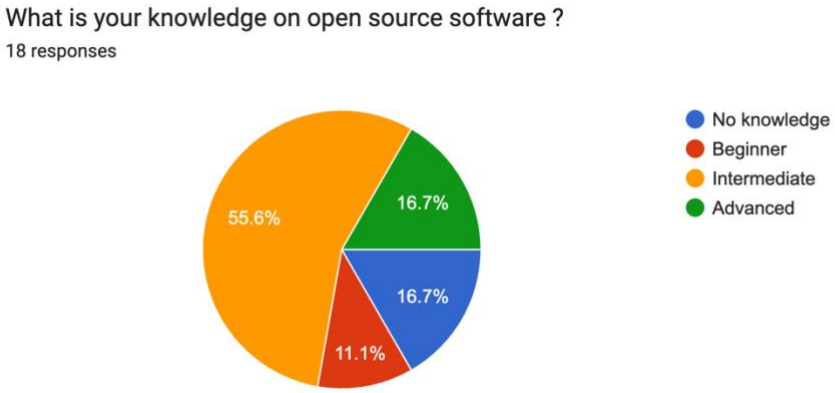


Figure 7.1: Knowledge on open-source software chart.

The open-source software license helper chatbot interface, which was the subject of the given feedback, showed that the majority of the users were at the level of intermediate and advanced in terms of user experience. This allocation implies that a big fraction of users was at least somewhat familiar with the software licensing and technology concepts.

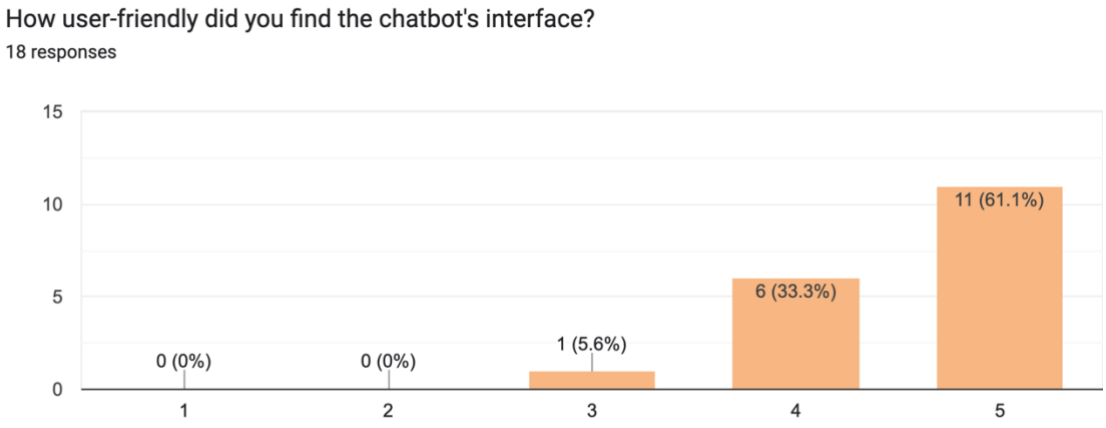


Figure 7.2: Results on chatbot user-friendliness.

Regarding the user satisfaction, the responses revealed that most of the users found the chatbot interface to be visually appealing. The great visual appeal and the choice of the color scheme were most probably the factors that contributed to the positive user experience and the active participation in the chatbot.

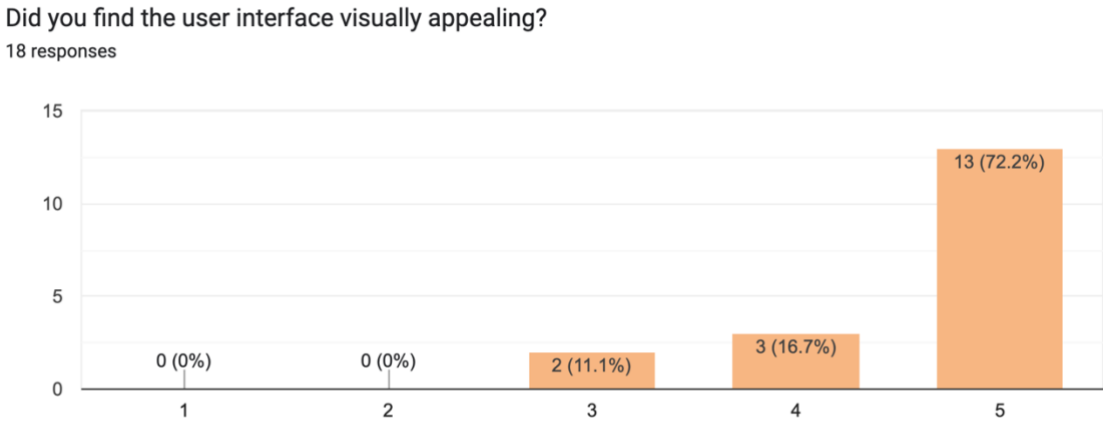


Figure 7.3: Results on user-interface visual appeal.

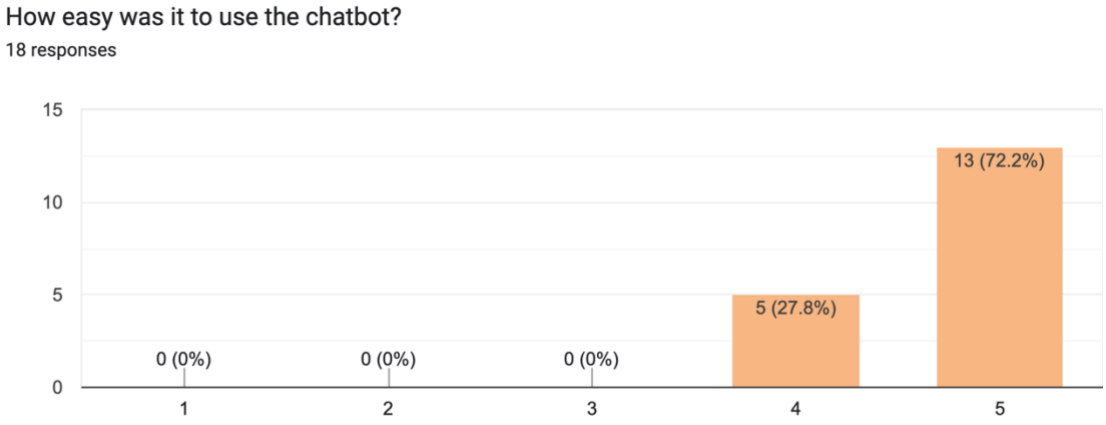


Figure 7.4: Results on user-experience with the chatbot.

The user friendliness was also good as 89% of users said that they could easily understand the interface. This proves that the chatbot's design and layout were good in guiding the users through the software licensing process even for different levels of experience.

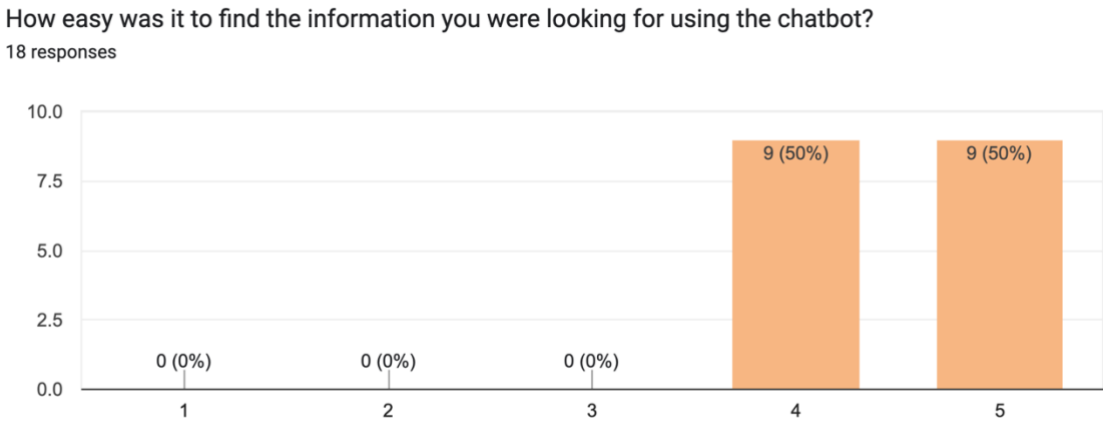


Figure 7.5: Results on how easy it was to find the information needed.

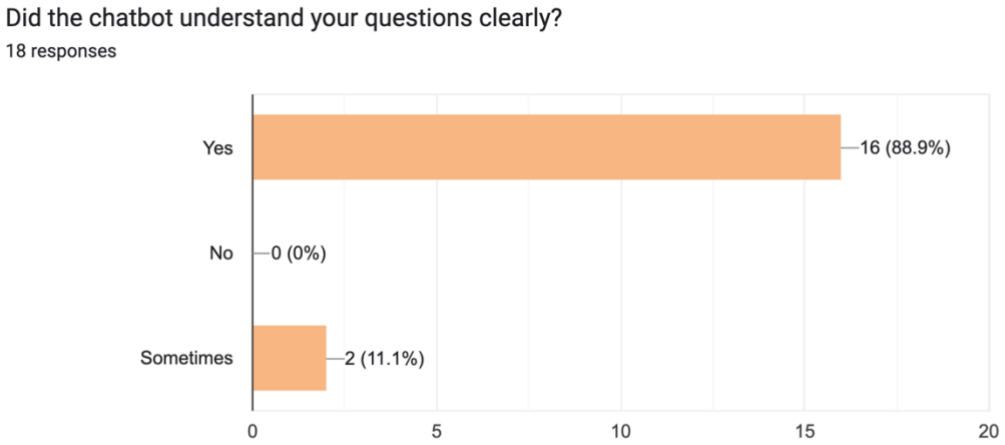


Figure 7.6: Results on user-query understandability by chatbot.

Regarding finding the right information almost all the users agreed that the chatbot accurately understood their questions. The depth of the understanding is the main characteristic of a chatbot that is designed to assist users who are dealing with complex matters like software licensing. The chatbot's ability to comprehend the user's questions correctly is the main reason for providing the relevant and helpful answers. The positive feedback on comprehension says that the chatbot was able to comprehend user inputs well and making the user satisfied.

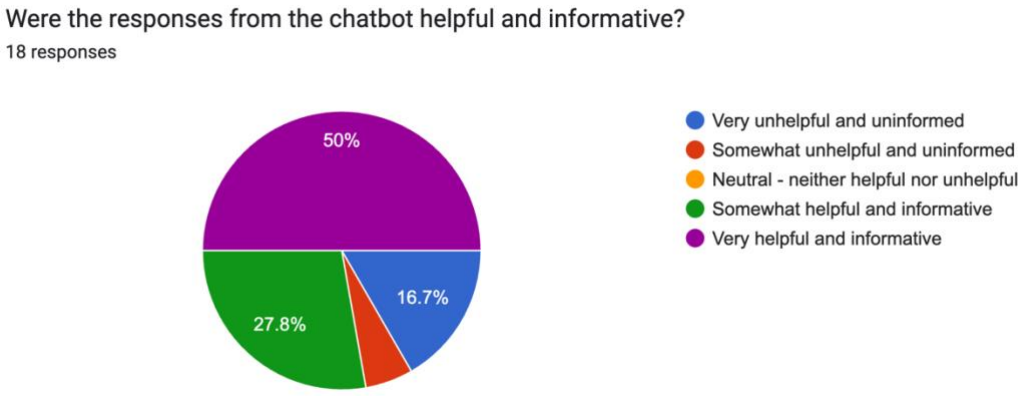


Figure 7.7: Results on how useful users found the information provided by chatbot.

As for the information and the helpfulness of the chatbot, 50% of users found it very helpful and 27. 8% of the respondents thought of them as slightly useful. This is the way in which the chatbot got the positive feedback which implies that the chatbot was effective in giving the useful and appropriate information to the users who were asking for assistance with the software licenses. The chatbot's informative responses are the key for the users to be led through the intricacies of the software licensing and to make the right decisions. Only 22% of the respondents suggest a potential for future improvement in the chatbot for the information to be more detailed and clear.

Chapter 8

Conclusion

8.1 Summary of Contributions

This thesis has successfully developed a chatbot system to assist users in understanding and selecting the most appropriate software licenses for their projects. The key contributions of this research work are as follows:

1. Conducted a comprehensive analysis of the software licensing landscape, including legal frameworks, license types, and challenges faced by users in navigating the complex licensing process.

2. Developed a robust knowledge base encompassing a wide range of software licenses, their features, permissions, and restrictions, enabling the chatbot to provide accurate and relevant recommendations.

3. Employed advanced natural language processing (NLP) techniques and machine learning algorithms, specifically leveraging the Rasa framework, to enable the chatbot to understand and interpret user queries in a human-like manner.

4. Designed and implemented a user-friendly, conversational interface that simplifies the software licensing experience, making it accessible to users with varying levels of expertise.

5. Performed rigorous testing and evaluation of the chatbot, demonstrating its accuracy, user satisfaction, and effectiveness in streamlining the license selection process.

6. Contributed to the advancement of chatbot technology, particularly in the domain of legal and compliance applications, by exploring the integration of customized actions and the iterative improvement of machine learning models.

7. Provided a scalable and adaptable solution that can accommodate future updates and expansions, ensuring its relevance as new licensing models emerge and user requirements evolve.

8.2 Limitations and Future Work

Even though the license chatbot software developed in this thesis is an important improvement in user support in software licensing, it is essential to recognize the limitations of the system and consider possible future improvements.

The chatbot's inability to comprehend the user's intention is one of the constraints established during the evaluation phase when the user used multiple syntax errors or wrong vocabulary usage. Even though the chatbot has natural language processing competence, human language with its complications and variations can sometimes bring about confusion. This is particularly the case when users enter queries with substantial deviations from standard language patterns or employ highly technical or niche terminology that was not considered in the chatbot's training data.

Another shortcoming is the depth of chatbot's knowledge. While it may give a solution to many questions that relate to software licenses there are topics that it cannot handle. Currently the chatbot is built to cater for questions that are within its programmed knowledge base and may fail to adequately respond to questions that go beyond this set scope. This limitation is a window of opportunity for the future works that aim at extending the chatbot abilities and to enable it to cover a wide range of topics about software licensing.

As for the diversity of licenses, the chatbot provides a wide spectrum; however, there is always something to improve. Additional licenses could be added in future versions of chatbot, specifically those that are less common or are emerging in the industry. However,

when the knowledge base is continuously updated to cover these rarer licenses, the chatbot assumes even more value for users with varied and specific licensing needs.

Future work on the chatbot could focus on several areas:

1. Enhanced Language Understanding: By using more advanced NLP techniques and machine learning algorithms, the chatbot could be given a better ability of processing and understanding user queries, even, when language irregularities are presented.

2. Expanded Knowledge Base: Keeping the knowledge base of a chatbot updated with additional licenses and licensing situations will make the latter a more relevant and complete tool.

3. Additional Features: For instance, the chatbot's capability to handle complex conversation or to be linked with external legal databases in order to get real-time information could be a game changer to its utility.

4. User Feedback Loop: The inclusion of a powerful user feedback system enables the users' chatbot to be identified of any possible drawbacks and to provide the required data to refine its reply and abilities.

5. Internationalization: The chatbot's language support extension for non-English speakers could expand its user base and make it available to the whole world.

In future work, a significant enhancement could be also achieved by integrating the specialized software license chatbot with a large language model (LLM). This integration would take advantage of the impressive NLU and NLP features that are evident in LLMs to facilitate improved intent identification and entity extraction.

These parameters can then be used to train the LLM when a clear pattern of intents and entities are defined in the chatbot. Due to the LLM's higher processing capabilities and experience in different database types, it can give more accurate and versatile

interpretations of the user's query. This means it can interpret and deconvolute user intents more effectively than the LLM when questions are posed in twisted or non-standard ways.

The integration would work as follows:

Intent Recognition: The LLM would be trained on several basic predefined intents including those that are specific to software licensing. This means that it would be able to understand the natural language better, and therefore be better placed to understand the user's intent no matter how the question is framed. This would help in minimizing the cases where the chatbot will not be able to understand or misunderstand what the user is searching for as the information he or she is searching for is not in the predefined scenarios.

- Entity Extraction: The LLM can easily filter out the entities that the user is interested in, be it specific software license names, the type of project, or legal terms. These entities are well picked by the NLP to give accurate responses that are expected by users of the system.

- Enhanced User Interaction: Thus, the improvement of the NLU and NLP of the LLM makes the chatbot capable of answering a greater number of questions and giving more detailed answers to them. This makes the interaction seamless and more productive, enhancing the user experience.

- Continuous Learning: The LLM can be updated with new data from time to time, which makes the knowledge of the machine grow with time. This dynamic learning capability will make sure that the chatbot is always up to date in the ever-evolving field of software licensing.

Through overcoming these constraints and concentrating on the suggested directions for future research, the software license chatbot may develop further and meet the needs of its customers, thereby turning into an even more effective instrument in the software licensing area.

8.3 Final Thoughts

The development of the software license chatbot represents a significant step forward in addressing the challenges associated with software licensing. By leveraging cutting-edge technologies, this research has successfully bridged the gap between complex legal frameworks and user-friendly interfaces, empowering users to make informed decisions regarding software licenses.

The chatbot's ability to engage in natural conversations, provide personalized recommendations, and simplify the licensing process has the potential to revolutionize the way software developers, organizations, and end-users approach software licensing. By promoting a better understanding of licensing terms and implications, the chatbot can contribute to increased compliance, reduced legal risks, and a more collaborative and innovative software ecosystem.

Furthermore, this research has demonstrated the versatility and potential of chatbot technology in tackling complex decision-making scenarios, paving the way for similar solutions in other domains where intricate legal or regulatory frameworks exist.

References

[1] J. Juviler, "9 Guidelines & Best Practices for Exceptional Web Design and Usability," 27 5 2021. [Online]. Available: https://blog.hubspot.com/blog/tabid/6307/bid/30557/6-guidelines-for-exceptional-website-design-and-usability.aspx.

[2] "TLDRLegal," FOSSA, [Online]. Available: https://www.tldrlegal.com/.

[3] M. Di Penta, M. D. German, Y.-G. Guéhéneuc and G. Antoniol, An Exploratory Study of the Evolution of Software Licensing, vol. 1, Proceedings of the 32nd ACM/IEEE
International Conference on Software Engineering, 2010, pp. 145-154.

[4] A. M. St. Laurent, "Understanding open source and free software licensing: guide to navigating licensing issues in existing & new software.," 2004. [Online]. Available: https://people.debian.org/~dktrkranz/legal/Understanding%20Open%20Source%20and%20Free%20Software%20Licensing.pdf.

[5] "Choose a license," Github, [Online]. Available: https://choosealicense.com/.

[6] A. Jiao, "An intelligent chatbot system based on entity extraction using RASA NLU and neural network," *Journal of Physics: Conference Series,* vol. 1487, no. 1, p. 012014, March 2020.

[7] "Rasa Documentations," Rasa, [Online]. Available: https://rasa.com/docs/rasa/.

[8] "Rasa framework core," indiaAI, [Online]. Available: https://indiaai.gov.in/article/rasa-chatbot-framework-nlu-core.

[9] D. Loesche, "Illegal Business Software in Use Worldwide," *Statista,* 7 June 2018.

[10] D. Crane, "The Developer's Guide: Open Source Software License Comparison," 13 June 2019.

[11] J. Thanaki, Python Natural Language Processing, Packt Publishing, 2017.

[12] IBM, "What are LLMs?," [Online]. Available: https://www.ibm.com/topics/large-language-models.

[13] Flask, "Flask Documentation," [Online]. Available: https://flask.palletsprojects.com/en/3.0.x/.

[14] "Frontend web technologies," [Online].

Available: https://www.studysmarter.co.uk/explanations/computer-science/computer-network/web-technologies/#:~:text=Client%2Dside%20technologies%20include%20HTML,and%20frameworks%2C%20such%20as%20Node..

[15] B. Community, "13 Best Open Source Chatbot Platforms to Use," 14 7 2022. [Online]. Available: https://botpress.com/blog/open-source-chatbots.

[16] Tensorflow "Essential documentation",03 02 2023. [Online]. Available: https://www.tensorflow.org/guide

Appendices

A. User Guide

What can the user ask the chatbot?

- You can ask the chatbot to give you some information(summary) about a license.

  - Example: Can you give me some info about GPL-2.0?

- You can ask the chatbot if a license allows a specific permission (The permissions you can ask are listed below)

  - Example: Does the software license EPL-2.0 permit commercial-use?

- You can ask the chatbot to suggest you a license through a series of questions. Different questions are listed based on your knowledge about software licenses.

  - Example: Can you suggest me some software licenses?

- You can ask the chatbot to suggest you some licenses that [allow], [restrict] and [offer] specific permissions. There is a recommended format for asking this question. For each keyword you can use a specific subset of permissions. Make sure to separate them with 'and'(See examples below)

1. allow: [commercial-use, modifications, distribution, private-use, sublicensing, patent-use, trademark-use]
2. demand: [include-copyright, disclose-source, document-changes, network-use-disclose, same-license]
3. offer: [liability, warranty]

- Example: Can you suggest me some licenses which accept distribution, sublicensing, commercial-use, modifications, private-use and does not demand document-changes and offer liability, warranty
- Example: Tell me some licenses that allow patent-use, trademark-use, private-use and demand same-license, disclose-source, network-use-disclose and give liability, warranty

- You can ask the chatbot what benefits or restrictions the license have

  - Example: What are the benefits of EUPL-1.2?

  - Example: What benefits does MPL-2.0 have?

- You can ask the chatbot about the definition of a software license

  - Example: What is a software license?

  - Example: Can you help me understand what a software license is?

The following table shows which software licenses are supported from the chatbot:

License Names:	License Ids:
GNU Affero General Public License v3.0	AGPL-3.0
Vim License	Vim
Creative Commons Attribution Share Alike	CC-SA

License Names:	License Ids:
The Unlicense	Unlicense
Artistic License 2.0	Artistic-2.0
BSD 3-Clause "New" or "Revised" License	BSD-3-Clause
EU DataGrid Software License	EUDatagrid
Eclipse Public License 1.0	EPL-1.0
European Union Public License 1.1	EUPL-1.1
LaTeX Project Public License v1.3c	LPPL-1.3c
GNU Lesser General Public License v2.1	LGPL-2.1
Creative Commons Attribution 4.0 International	CC-BY-4.0
BSD Zero Clause License	0BSD
Microsoft Public License	MS-PL
ISC License	ISC
GNU General Public License v3.0	GPL-3.0
Open Data Commons Open Database License v1.0	ODbL-1.0
MIT No Attribution	MIT-0
Creative Commons Attribution NonCommercial NoDerivs	CC-NC-ND
Academic Free License v3.0	AFL-3.0
Microsoft Reciprocal License	MS-RL
Eiffel Forum License v2.0	EFL-2.0
Common Development and Distribution License	CDDL-1.0
Creative Commons Zero v1.0 Universal	CC0-1.0
GNU Free Documentation License v1.3	GFDL-1.3
Mulan Permissive Software License, Version 2	MulanPSL-2.0
Cube License	CL
Apache License 2.0	Apache-2.0
Boost Software License 1.0	BSL-1.0
University of Illinois/NCSA Open Source License	NCSA

License Names:	License Ids:
Creative Commons Attribution Share Alike 4.0 International	CC-BY-SA-4.0
BSD 2-Clause "Simplified" License	BSD-2-Clause
GNU Lesser General Public License v3.0	LGPL-3.0
Creative Commons Attribution NoDerivs	CC-ND
Eclipse Public License 2.0	EPL-2.0
SIL Open Font License 1.1	OFL-1.1
Adaptive Public License 1.0	APL-1.0
Open Software License 3.0	OSL-3.0
Creative Commons Attribution-NoDerivatives 4.0 International	CC-BY-ND-4.0
CeCILL Free Software License Agreement v2.1	CECILL-2.1
Apple MIT License	AML
MIT License	MIT
CUA Office Public License v1.0	CUA-OPL-1.0
Mozilla Public License 2.0	MPL-2.0
"Do What The F*ck You Want To Public License"	WTFPL
Educational Community License v2.0	ECL-2.0
zlib License	Zlib
Universal Permissive License v1.0	UPL-1.0
European Union Public License 1.2	EUPL-1.2
Computer Associates Trusted Open Source License 1.1	CATOSL-1.1
Apple Public Source License	APSL-2.0
Erlang Public License v1.1	ErlPL-1.1
GNU General Public License v2.0	GPL-2.0
Creative Commons Attribution NonCommercial ShareAlike	CC-NC-SA
BSD 4-Clause "Original" or "Old" License	BSD-4-Clause
Common Public License 1.0	CPL-1.0
PostgreSQL License	PostgreSQL

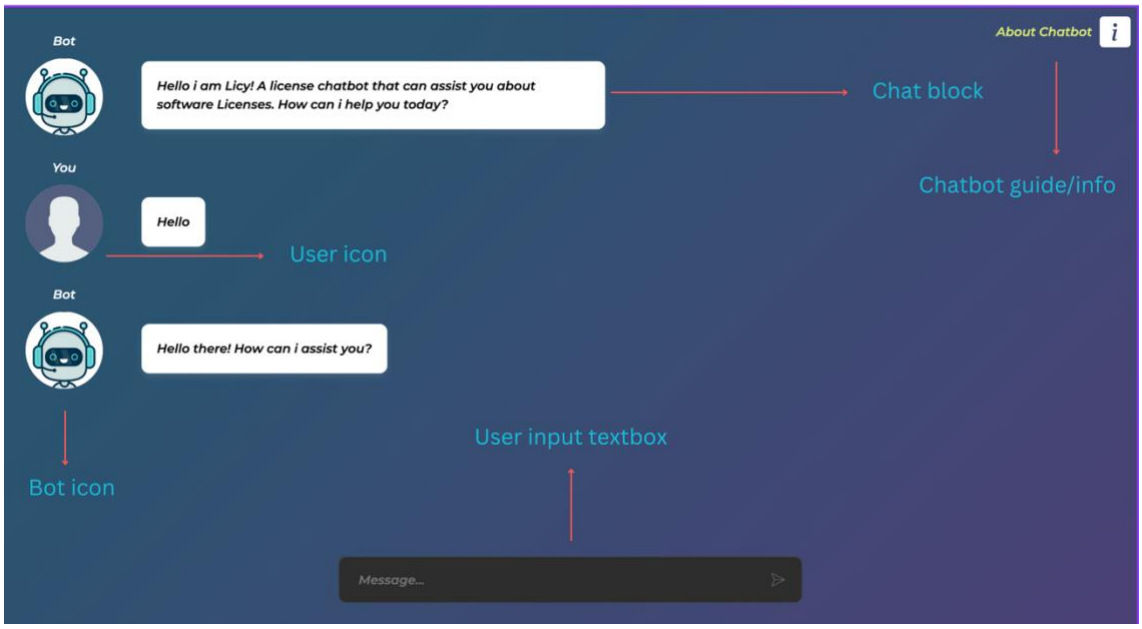
License Names:	License Ids:
Creative Commons Attribution NonCommercial	CC-BY-NC

The following table shows the permissions of the software licenses and its explanations.

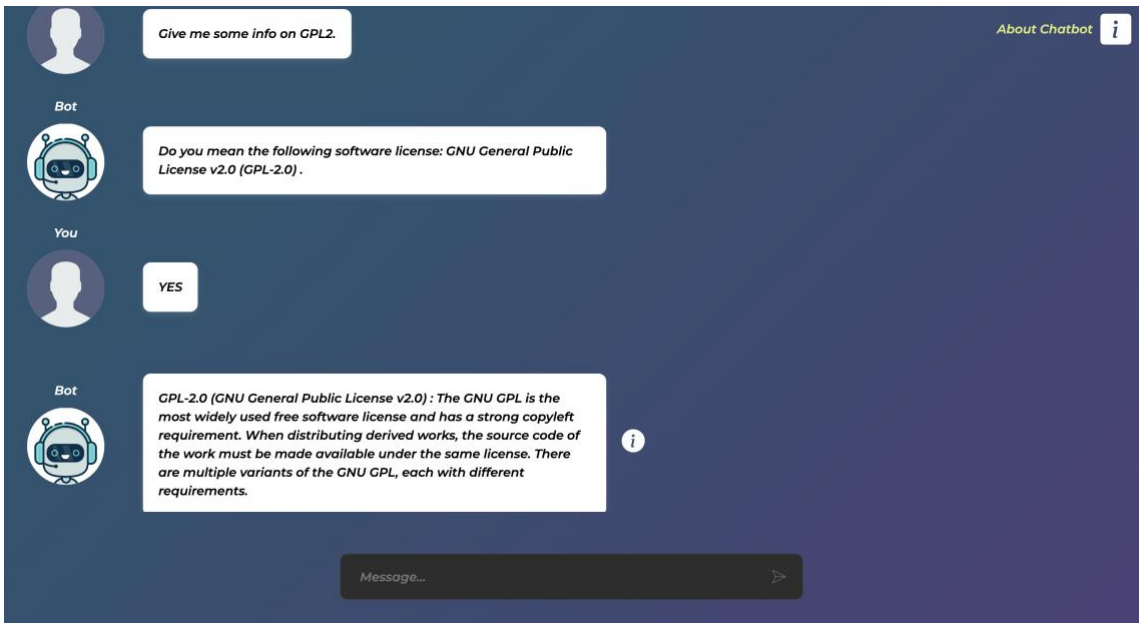
Permission	**Explanation:**
commercial-use	If the licensed material and derivatives may be used for commercial purposes.
modifications	If the licensed material may be modified.
distribution	If the licensed material may be distributed.
private-use	If licensed material may be used and modified in private.
sublicense	If others who get a license from you can grant different licenses than the one you used to others
patent-use	If this license provides an express grant of patent rights from contributors
include-copyright	If a copy of the license and copyright notice must be included with the licensed material in source form but is not required for binaries. Network use is distribution
disclose-source	If source code must be made available when the licensed material is distributed.
document-changes	If changes made to the licensed material must be documented.
network-use-disclose	If users who interact with the licensed material via network are given the right to receive a copy of the source code.
same-license	If modifications must be released under the same license when distributing the licensed material. In some cases, a similar or related license may be used, or this condition may not apply to works that use the licensed material as a library.
liability	If this license includes a limitation of liability.
warranty	If this license explicitly states that it does provide any warranty.

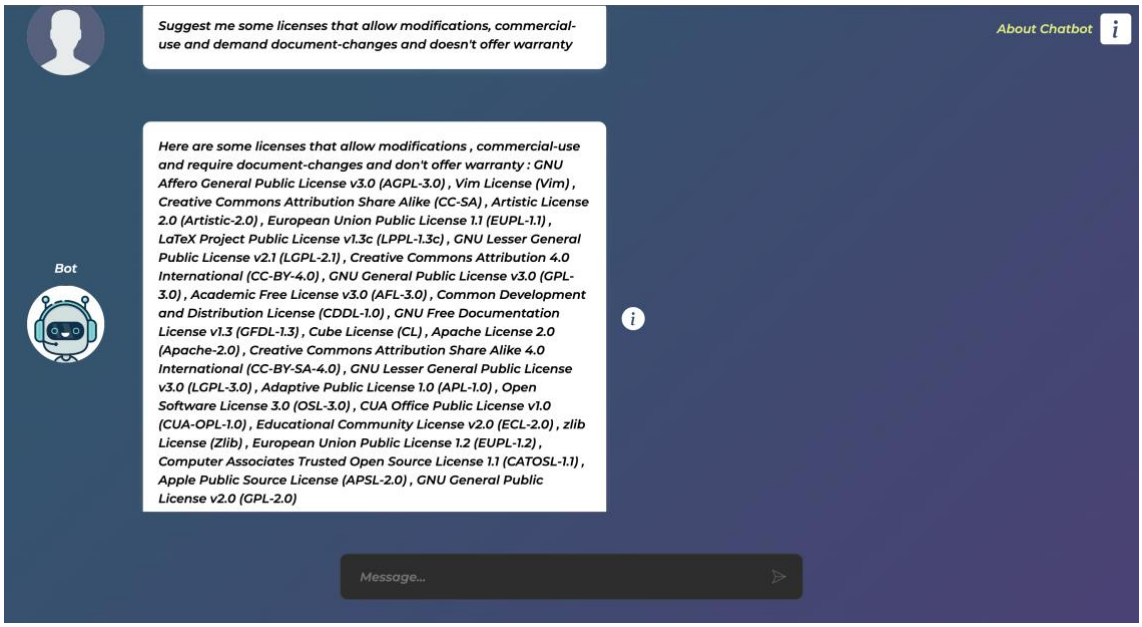
Permission	Explanation:
trademark-use	This license explicitly states that it does grant trademark rights, even though licenses without such a statement probably do not grant any implicit trademark rights.

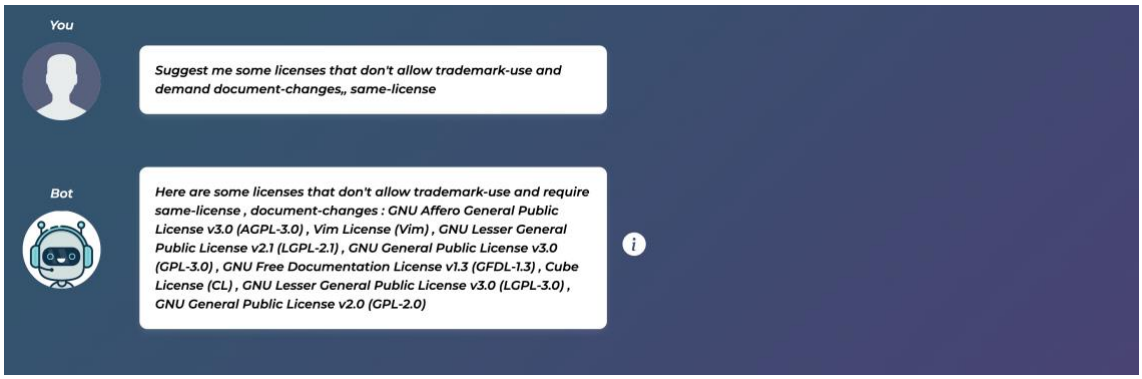
User interface components explained

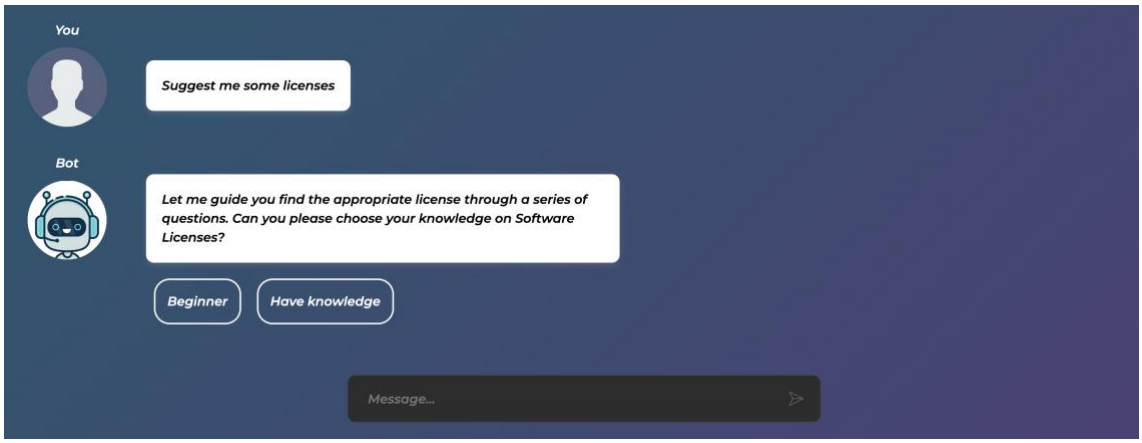


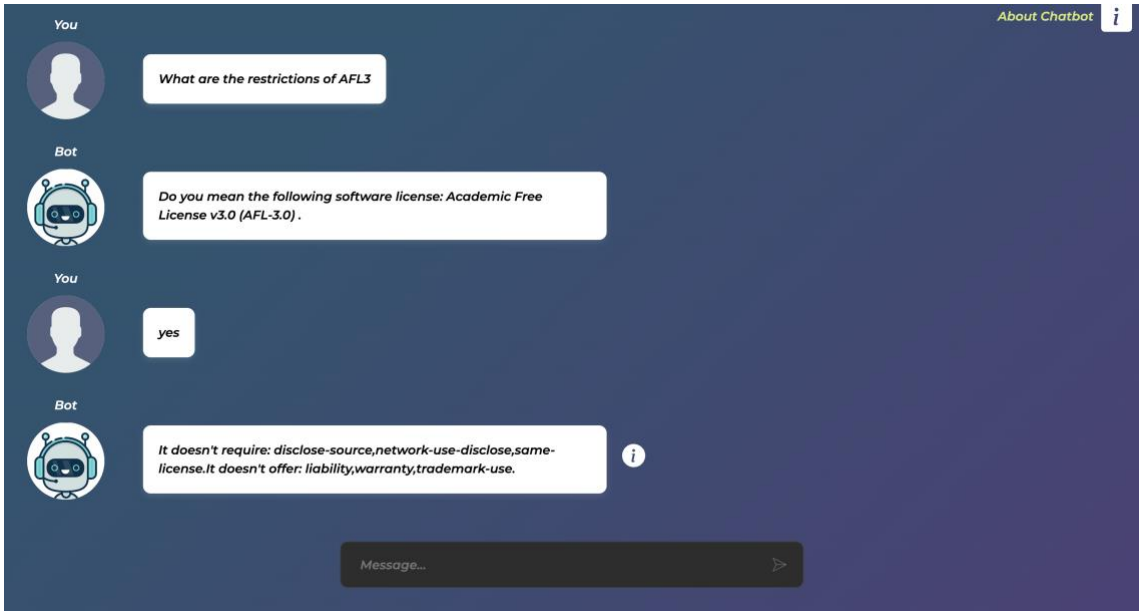
B. Chatbot dialogue examples

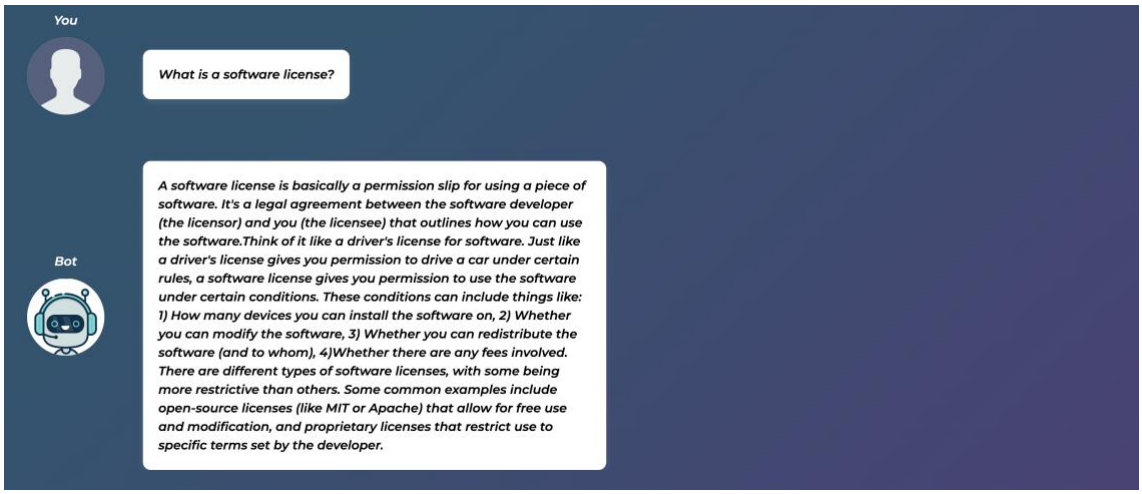





You
Suggest me some licenses that don't allow trademark-use and demand document-changes,, same-license
Bot
Here are some licenses that don't allow trademark-use and require same-license , document-changes : GNU Affero General Public License v3.0 (AGPL-3.0) , Vim License (Vim) , GNU Lesser General Public License v2.1 (LGPL-2.1) , GNU General Public License v3.0 (GPL-3.0) , GNU Free Documentation License v1.3 (GFDL-1.3) , Cube License (CL) , GNU Lesser General Public License v3.0 (LGPL-3.0) , GNU General Public License v2.0 (GPL-2.0)


You
Suggest me some licenses
Bot
Let me guide you find the appropriate license through a series of questions. Can you please choose your knowledge on Software Licenses?
Beginner
Have knowledge
Message...


About Chatbot
You
What are the restrictions of AFL3
Bot
Do you mean the following software license: Academic Free License v3.0 (AFL-3.0) .
You
yes
Bot
It doesn't require: disclose-source,network-use-disclose,same-license.It doesn't offer: liability,warranty,trademark-use.
Message...


You
What is a software license?
Bot
A software license is basically a permission slip for using a piece of software. It's a legal agreement between the software developer (the licensor) and you (the licensee) that outlines how you can use the software.Think of it like a driver's license for software. Just like a driver's license gives you permission to drive a car under certain rules, a software license gives you permission to use the software under certain conditions. These conditions can include things like: 1) How many devices you can install the software on, 2) Whether you can modify the software, 3) Whether you can redistribute the software (and to whom), 4)Whether there are any fees involved. There are different types of software licenses, with some being more restrictive than others. Some common examples include open-source licenses (like MIT or Apache) that allow for free use and modification, and proprietary licenses that restrict use to specific terms set by the developer.