



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

IJRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 13, Issue 4, April 2024

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.423

 9940 572 462

 6381 907 438

 ijirset@gmail.com

 www.ijirset.com

Natural language Understanding Platform for Chat-bot

Prof.V.B.Deokate^{*1}, Renuka.V.Puri^{*2}, Mayur.S.Wabale^{*3}

Assistant Professor, Dept. of IT Engineering, SVPM's College of Engineering Malegaon(Bk), Baramati, India. ^{*1}

UG Students, Dept. of Information Technology Engineering, SVPM's College of Engineering, Malegaon(Bk), Baramati, India. ^{*2,3}

ABSTRACT: The fundamental technology that drives chat-bots and enables them to understand natural language input is called natural language understanding (NLU). Many NLU platforms have recently been made available as ready-built NLU components for chat-bots; however, selecting the best NLU for Software Constructing chat-bots remains an unsolved task. Therefore, in this research, we evaluate four of the most widely used NLUs: Google Dialog-flow, IBM Watson, Rosa, additionally Microsoft LUIS will clarify which NLU should be used in chat-bots based on software engineering. By allowing practitioners to ask inquiries about their software projects and interact with numerous services using natural language, chat-bots are expected to drastically change the discipline of software engineering. Specifically, we investigate the performance of the NLUs in intent categorization, confidence score stability, and entity extraction. We employ two data-sets that reflect two common actions performed by software engineering practitioners in order to evaluate the NLUs: a chat-bot for software repository inquiries and the behavior of posting development-related questions on Q&A websites (such as Stack Overflow). IBM Watson is the NLU that performs the best in all three categories (entity extraction, confidence ratings, and intents categorization), according to our research. However, the results for each individual component show that IBM Watson performs best in intents categorization with an F1-measure >84%, whereas Rosa leads in confidence scores with a median confidence score larger than 0.91. Furthermore, Our results show that all NLUs provide fairly robust confidence scores, except for Dialog-flow. In the two SE tasks, Microsoft LUIS and IBM Watson perform better for entity extraction than other NLUs. Software engineers can use our results as recommendations when choosing the neural network architecture (NLU) to use in their chat-bots.

KEYWORDS: Software Chat-bots, Natural Language Understanding Platforms, Empirical Software Engineering

I. INTRODUCTION

In the discipline of software engineering (SE), software chat-bots are becoming more and more popular because they allow users to interact with platforms using natural language, automate repetitive tasks to save time and effort, and more. Bot-related publications, conferences, seminars, and publications have multiplied, suggesting a discernible rise in interest. A recent study found that one out of every four open-source projects (OSS) on Git-hub uses software bots for diverse purposes. This is supported by the fact that bots streamline daily tasks for developers, such as updating dependencies, producing fixing patches, and deploying builds. Platforms for Natural Language Understanding, or NLU for short are the cornerstone of every chat bot. NLUs are necessary for the chat-bot to comprehend and react to user input. Using machine learning and natural language processing (NLP) techniques, the NLU takes unstructured user input (textual data) and extracts structured information (the user's query intent and related entities). Rather than building an NLU from scratch, which requires expertise in natural language processing, chat-bot developers utilize a select few widely used NLUs in their chat-bots. Owing to the multitude of widely-used NLUs, developers are required to select the most suitable NLU for their particular domain. This is a difficult task that has been thoroughly studied in previous research (especially because NLUs function differently in different environments). Due to As an illustration, Canonic & DE Russia showed how IBM Watson outperformed in the meteorological sector. Gregori evaluated NLUs using often asked questions from university students and found that Dialog-flow worked best. In actuality, there is

much. The best NLU to employ when deploying chat-bots is a topic of discussion on Stack Overflow since choosing the wrong platform for a given topic can negatively affect the catboat's user experience. SE is a significant domain in which different NLUs' performance has not been examined. Software engineering is a specialist discipline with a very specific terminology used in a particular way. In the SE domain, for example, the term "ticket" designates a bug in a bug tracking system (such as Jira); yet, in other domains, it's associated with a film ticket (like Ticketmaster's bot) or a plane ticket. Furthermore, there is debate among SE chat-bot developers regarding whether NLU is more appropriate for the SE domain. For instance, Task-bot uses Microsoft Language Understanding Intelligent Service (LUIS) to help practitioners manage their tasks. MSR~Bot uses Google Dialog-flow NLU to reply to questions on the software repository. M Sabot assists practitioners in creating and managing micro-services by utilizing Rosa NLU. Because no research has examined which NLU performs best in the SE domain, chat-bot developers are unable to determine which NLU to use when developing SE-based chat-bots. Thus, in this research, we give the first evaluation of the effectiveness of popular NLUs to support SE tasks. We evaluate NLUs with questions related to two important SE tasks. Stack Overflow: Analyzing data from project repositories (e.g., "Which file in my repository is the most buggy?"), and Repository: Technical questions that developers frequently ask and get answers to through Q&A websites. For instance, "How can I turn an X Element object into a data-set or data table?" Using the two SE tasks, we evaluate four well-known NLUs: IBM Watson, Microsoft LUIS, Google Dialog-flow, Rosa, and Rosa. The accuracy with which NLUs categorize user intents; the level of confidence they demonstrate in accurately and incorrectly classifying queries (i.e., confidence score); and the accuracy with which NLUs extract entities from queries (i.e., accurately classify correct subjects)

- As far as we are aware, this is the first study to evaluate NLUs on two sample tasks from the software engineering domain (i.e., using data from Stack Overflow posts and software repositories).
- We evaluate the NLUs using different features (i.e., list and prediction features) for entity extraction.
- We study how the NLUs' intent classification performance is affected by the selection of different confidence score criteria.
- We provide a set of realistic recommendations to chat-bot operators to improve the performance of their NLU, based on our study and experience.
- We make our tagged data-set publicly available to encourage replication and future research in the field. The document's remaining sections are organized as follows: An overview of chat-bots and the associated concepts used in this study explained in Section. Section describes the case study setup that was used to evaluate the performance of the NLUs. We offer the evaluation findings in Section. Our findings are presented in this section along with some recommendations for enhancing the classification results.

II.METHODOLOGY

• Requirements Gathering:

Define the purpose and scope of the chat-bot (e.g., what types of tickets will it track, who will use it).

Gather requirements from stakeholders (e.g., users, support team) through interviews, surveys, or user stories.

•Design:

Design the conversation flow, including how the chat-bot will greet users, collect ticket information, and provide responses.

Define the back-end architecture, including how the chat-bot will interact with the ticket tracking system/database.

•Development:

Choose a platform or framework for building the chat-bot

Implement the catboat's conversation logic and integration with the ticket tracking system.

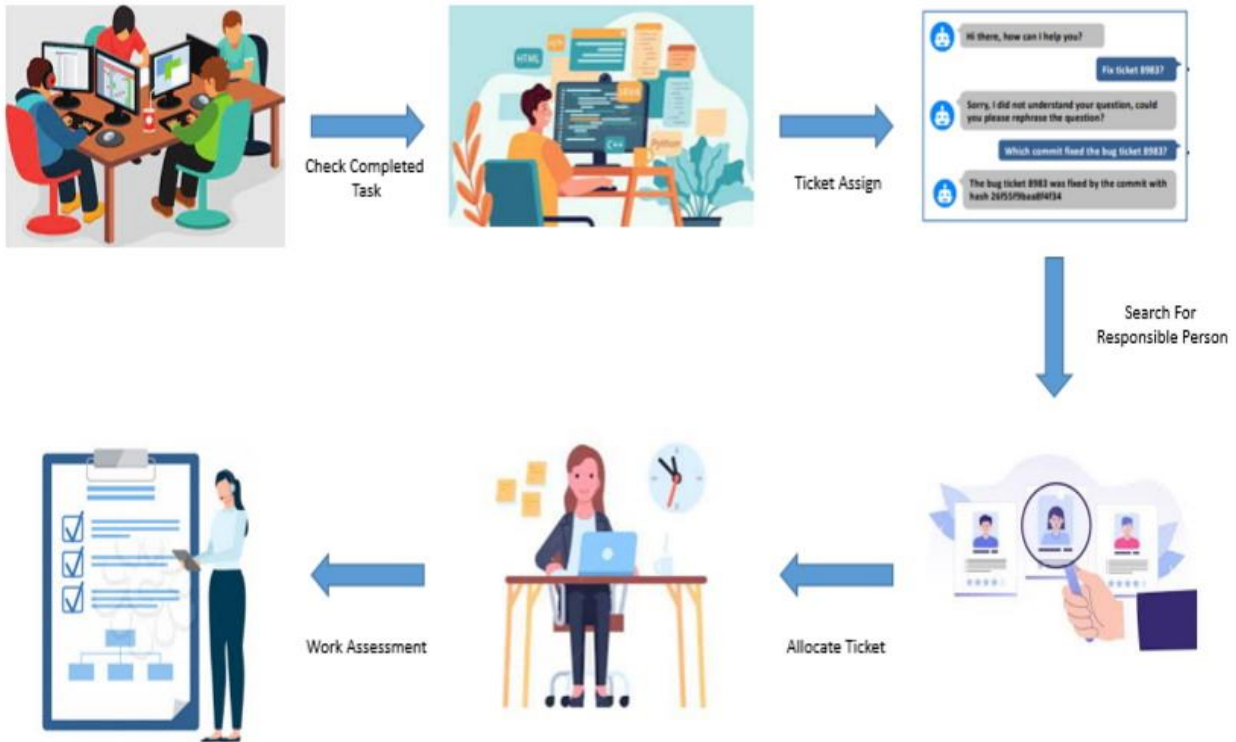
Test the catboat's functionality, including edge cases and error handling.

•Maintenance and Updates:

Regularly update the chat bot to improve its functionality and address any issues that arise.

Monitor user feedback and analytic to identify areas for improvements

III. MODELING AND ANALYSIS



ARCHITECTURE

1. User Interaction:-

Users can create a new ticket by providing a description of the issue.

Users can view the status of their tickets and any updates made by employees.

2. Employee Interaction:-

Employees can view a list of tickets assigned to them.

Employees can update the status of a ticket (e.g., in progress, resolved).

3. Admin Interaction:-

Admins can view a list of all tickets.

Admins can assign tickets to developers and testers.

4. Developer Interaction:-

Developers can view a list of tickets assigned to them.

Developers can update the status of a ticket to indicate that they are working on it or that it is resolved.

5. Tester Interaction:-

Testers can view a list of tickets assigned to them.

Testers can update the status of a ticket to indicate that they have verified the fix or that further action is needed.



IV. RESULTS AND DISCUSSION

Improved Ticket Management:-

The chat-bot can automate ticket creation, assignment, and status updates, reducing the manual effort required by employees.

By integrating with the ticket tracking system, the chat-bot ensures that tickets are managed consistently and efficiently.

Enhanced User Experience:-

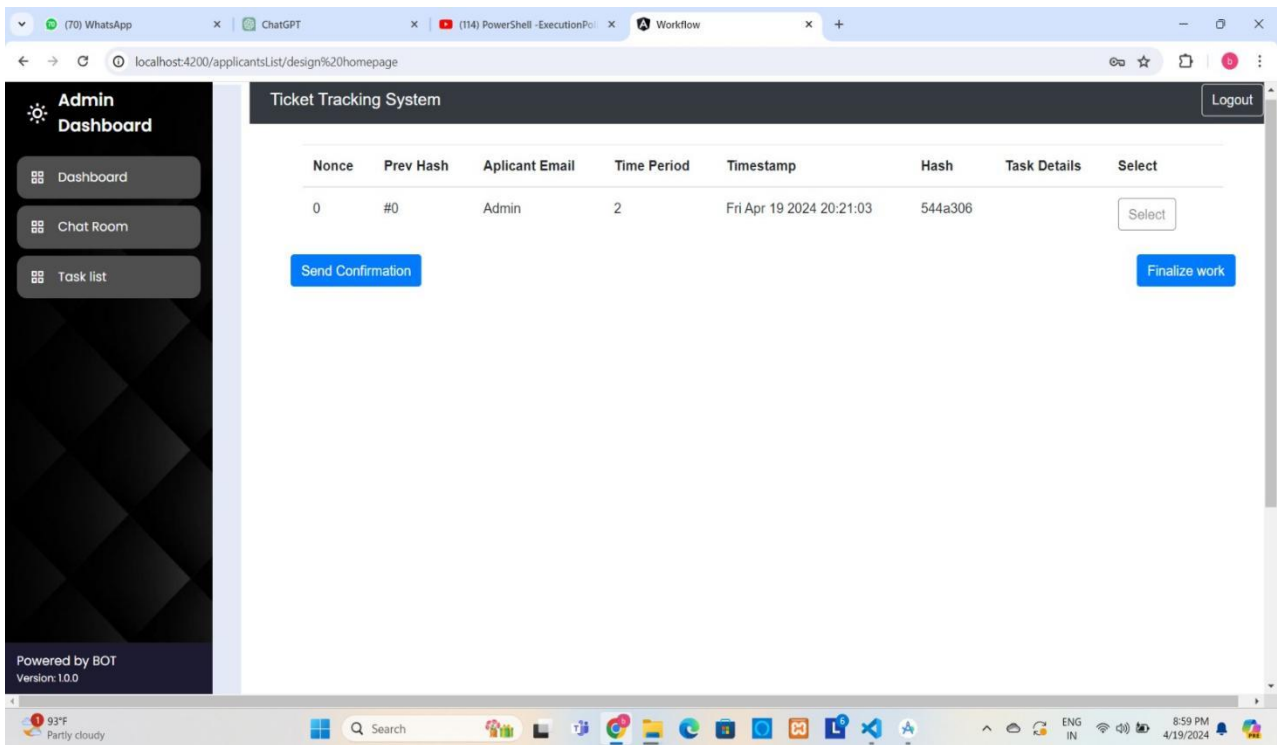
Users can easily create and track tickets using natural language conversations with the chatbot, making the process more intuitive and user-friendly.

The chat-bot can provide quick responses and updates, improving user satisfaction.

Increased Efficiency:-

The chat-bot can handle multiple ticket-related tasks simultaneously, reducing the time and effort needed to manage tickets.

Employees can focus on more strategic tasks, knowing that the chat-bot is handling routine ticket management tasks.



V. CONCLUSION

The ticket tracking chat-bot based on software engineering principles offers a streamlined and efficient way for companies to manage and resolve customer issues and internal tasks. By leveraging natural language processing and automation, the chat-bot simplifies the process of ticket creation, assignment, and resolution, leading to improved customer satisfaction and employee productivity. Software chat-bots are becoming more and more popular in the SE community as a potential cost and time-saver in development. Neural language units (NLUs) are the brains behind any chat-bot, enabling it to interpret user input. Selecting the best NLU for a chat-bot operating in the SE domain can be challenging. In this paper, we evaluate four widely used NLUs: IBM Watson, Google Dialog-flow, Microsoft LUIS,



and Rosa. We assess the NLU's in terms of intents classification, confidence score, and entity extraction on two different jobs taken from Stack Overflow and a repository. Therefore, we strongly encourage researchers to develop methods and approaches that enhance the NLU's performance on tasks involving different qualities.

ACKNOWLEDGEMENTS

We take this opportunity to thank our project guide Prof.V.B.Deokate and Head of the Department Prof.Dr.Gawade J.S. and Honourable Principal Prof.Dr.Mukane S.M. for their valuable guidance and for providing all the necessary facilities, which were indispensable in the completion of this project report. We are also thankful to all the staff members of the Department of Information Technology of SVPM's College of Engineering, Malegaon(Bk) for their valuable time, support, comments, suggestions Available at WWW.ijirset.com and persuasion. We would also like to thank the institute for providing the required facilities, Internet access and important books.

REFERENCES

- [1] A. Abdellatif, K. Bad-ran, and E. Shihab, "MSRBot: Using bots to answer questions from software repositories," *Empirical Software Engineering (EMSE)*, p. To Appear, 2019.
- [2] J. Dominic, J. House, I. Steinmacher, C. Ritter, and P. Herod, "Conversational bot for newcomers on boarding to open source projects," in *Proceedings of the 2nd International Workshop on Bots in Software Engineering*, ser. Bots E '20. IEEE Press, 2020.
- [3] M. Canonic o and L. DE Russia, "A comparison and critique of natural language understanding tools," *CLOUD COMPUTING* 2018, p. 120, 2018.
- [4] E. Gregory, "Evaluation of modern tools for an oms-cs ad visor chat-bot," 2017.
- [5] Stack Overflow, "Natural language processing," <https://stackoverflow.com/questions/4115526/natural-language-processing>, 03 2020, (Accessed on 03/09/2020).
- [6] Stack Overflow, "Artificial intelligence - comparison between Luis.AI vs api.AI vs wit.AI?" <https://stackoverflow.com/questions/37215188/comparison-between-Luis-ai-vs-api-ai-vs-wit-ai>, (Accessed on 04/11/2020).
- [7] Stack Overflow, "NLP - build chat-bot for education purpose," <https://stackoverflow.com/questions/52206324/build-chat-bot-for-education-purpose>, (Accessed on 11/25/2019).
- [8] C. Lebeuf, M. Storey, and A. Zagalsky, "Software bots," *IEEE Software*, vol. 35, no. 1, pp. 18–23, January/February 2018.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details