Ticket Tracking Chatbot Based On Software Engineering

V.B.Deokate*, Ritu Jagtap**,Renuka Puri**, Mayur Wabale**

*(Assistant Professor, Information Technology, SVPM's College of Engineering Malegaon(bk), Baramati
Email:vbdeokate@engg.svpm.org.in)
**(UG Students, Information Technology, SVPM's College of Engineering Malegaon(bk), Baramati)
Email:jagtapritu16@gmail.com ,renukapuri740@gmail.com , mayurwabale5600@gmail.com

Abstract:

It is envisaged that chatbots would fundamentally alter the field of software engineering by enabling practitioners to communicate with various services through natural language and ask questions on their software projects. Natural Language Understanding (NLU) is the core technology that powers chatbots and allows them to comprehend natural language input. Lately, a lot of NLU platforms were offered as a ready-made NLU element for chatbots; yet, choosing the ideal NLU for Software The challenge of building chatbots is yet unsolved.Thus, we assess four of the most popular NLUs in this paper: IBM Watson, Google Dialogflow, Rasa, and Which NLU should be utilized in chatbots based on software engineering will be clarified by Microsoft LUIS. We specifically look into how well the NLUs perform in extracting entities, confidence score stability, and intent classification. In order to assess the NLUs, we make use of two datasets that represent two typical tasks carried out by practitioners of software engineering:a chatbot for software repository inquiries the activity of posting queries about development on Q&A sites (like Stack Overflow).Based on our research, IBM Watson is the NLU with the best performance across the three dimensions (entity extraction, confidence scores, and intents categorization). The results for each individual component, however, indicate that Rasa leads in confidence scores with a median confidence score greater than 0.91, while IBM Watson performs best in intents categorization with an F1-measure>84%. .. Additionally, our data demonstrate that all NLUs—aside from Dialogflow—generally offer reliable confidence scores. For entity extraction, Microsoft LUIS andIBM Watson outperform other NLUs in the two SE tasks. Our results provide guidance to software engineering practitioners whendeciding which NLU to use in their chatbots.

Keyword:- *Software Chatbots, Natural Language Understanding Platforms, Empirical Software Engineering*

I.INTRODUCTION

Software chatbots are becoming more and more common in the Software Engineering (SE) field because they let users communicate with platforms through natural language, save time and effort by automating repetitive processes, and more The number of publications, conferences, workshops, and publications pertaining to bots has increased indicating a noticeable increase in interest.Software bots are being used for various activities in one out of every four open-source projects (OSS) on GitHub, according to a recent study. The fact that bots make developers' daily tasks liked eploying builds, updating dependencies, and even generating repairing patches more efficient lends credence to this. Natural Language Understanding platforms, or NLU for short, are the foundation of all chatbots.The chatbot's comprehension and response to user input depend on NLUs . The NLU takes unstructured user input (textual data) and utilizes machine learning and natural language processing (NLP) techniques to extract structured information (the user's query intent and related entities). Because creating an NLU from scratch is quite difficult and needs knowledge in natural language processing, chatbot developers instead employ a small number of commonly used NLUs in their chatbots Due to the variety of commonly-used NLUs, developers must choose the most appropriate NLU for their specific domain. This is a challenging assignment that has been extensively examined in earlier studies (particularly since NLUs differ in how they operate in various settings). Because For example, within the meteorological domain, Canonico & De Russis demonstrated how IBM Watson performed better. Gregori assessed NLUs using frequently asked questions by college students and discovered Dialogflow was the most effective. Actually, there isn't a shortage. Stack Overflow debates around the ideal NLU to use in the deployment of chatbots as selecting an inappropriate platform for a specific topic has a significant impact the chatbot's user satisfaction .

SE is an important domain where the performance of various NLUs has not been studied. Software engineering is a niche field with highly specialized vocabulary that is applied in a certain way. For instance, the term "ticket"

in the SE domain describes a bug in a bug tracking system (like Jira), but in other domains, it's associated with a movie ticket (like TicketMaster bot) or a plane ticket. Furthermore, there is disagreement among SE chatbot developers regarding whether NLU is most appropriate for the SE domain. For example, TaskBot assists practitioners in managing their duties by utilizing Microsoft Language Understanding Intelligent Service (LUIS). MSR~Bot responds to inquiries using Google Dialogflow NLU.about the repository of software. Utilizing Rasa NLU, MSABot helps practitioners create and manage microservices. Developers of chatbots cannot decide which NLU to utilize while creating SE-based chatbots because no study has looked at which NLU works best in the SE domain.

Therefore, we present the first study to evaluate popular NLUs' performance to assist SE tasks in this paper.We assess NLUs using queries associated with two significant SE tasks Repository: Examining project repository data (e.g., "Which file in my repository is the most buggy?"), and Stack Overflow: Technical queries that developers commonly pose and receive responses to via Q&A websites For example, "How can I turn an XElement object into a dataset or datatable?"We assess four popular NLUs using the two SE tasks: Microsoft LUIS, IBM Watson , Google Dialogflow , Rasa , and Rasa . The accuracy with which the NLUs classify the user's intents;The degree of confidence they exhibit in correctly classifying and misclassifying queries (i.e., confidence score); and The accuracy with which the NLUs identify the correct subjects from queries (i.e., entity extraction).

• To the best of our knowledge, this is the first study to assess NLUs on two sample tasks from the SE domain (i.e., data from software repositories and Stack Overflow posts).
• Using distinct features (i.e., list and prediction features) for entity extraction, we assess the NLUs.
• We investigate how choosing various confidence score criteria affects the NLUs' ability to classify intent.
• Based on our research and experience, we offer a series of doable suggestions to chatbot operators to enhance the functionality of their NLU.
• To facilitate replication and support upcoming studies in the area, we make our tagged dataset accessible to the general public .

The remainder of the document is structured as follows: A summary of chatbots and the related ideas utilized throughout this research are explained in Section . The case study arrangement used to assess the NLUs' performance is described in Section . In Section , we present the evaluation results.Section presents our results and offers some suggestions for improving the categorization outcomes.

II.BACKGROUND

We define the terms relevant to chatbots used throughout the paper in this part before delving into the evaluation of the NLUs. We also give a summary of the ways in which chatbots and NLUs collaborate to carry out specific tasks.

A. Definitions

Users and automated services are connected using software chatbots . Users ask the chatbot to carry out particular activities or to find out information using natural language. The NLU is then used internally by a chatbot to evaluate the user's request and take appropriate action.Extracting structured data from unstructured language input is an NLU's primary objective. Specifically, it takes user queries and extracts their intents and entities. Intents are the user's intention or purpose for the query, while entities are significant pieces of information. Consider a chatbot that responds to customer inquiries on software repositories, such as the MSRBot . "How many commits happened in the last month of the project?" is the question.Although each in its own way, the incorrect classification of intents and entities has a detrimental effect on the user experience.When an NLU incorrectly categorizes an intent, the chatbot is unable to comprehend the question at its core, which causes it to respond to a different query or carry out the incorrect action. On the other hand, miscclassifying entities results in the chatbot responding with incorrect information. For instance, there are three entities in the question "How to convert xml to json file in Java": "XML," "Json," and "Java."In the event that the NLU is unable to extract the "Java" item, the chatbot will no longer understand the question's context and may respond with a code example for converting XML to JSON using any other programming language, such as Python.Our goal is to examine how well the NLUs perform in terms of entity extractions, confidence scores, and intents classification. To guarantee that chatbots provide users with accurate and comprehensive responses, all three elements are essential.With a confidence level of 0.85, NLU retrieves the entity "ticket 8983" of type JiraTicket and categorizes the query's intent as GetFixingCommits. The chatbot then completes the required step by requesting information from the database in response to the inquiry, which reads, "The commit with hash 26f55f9baa8f4f34 fixed bug ticket 8983."

B Case Study Setup

We must choose the candidate NLUs we wish to investigate and the data corpus from the SE tasks to train and test those NLUs, as the primary objective of this paper is to assess the performance of various NLUs using SE tasks. This section describes the NLUs we chose, the SE tasks we utilized for the evaluation, and the design of our experiment.

I. Evaluated NLUs

Numerous popular NLUs are available that integrate with third-party apps with ease. In order to ensure the comprehensiveness of our research, we decided to analyze the output of four NLUs: IBM Watson, Dialogflow, Rasa, and LUIS. These NLUs are popular and frequently used by researchers and practitioners , and previous NLU comparative work in other fields has examined them. For these reasons, we have chosen them. Furthermore, the training process is facilitated by the ability to load the data into any chosen NLU by using their user interface or API calls. These NLUs are described in the sections that follow.

• IBM Watson Conversation: An NLU offered by the company [28]. IBM Watson comes with prebuilt models for many domains (like banking) and a visual dialog builder that makes it easier for nonprogrammers to create dialogs.

• Dialogflow: Google's neural learning unit More than 20 spoken languages are supported by Dialogflow, which may be linked with numerous chat services as Slack.

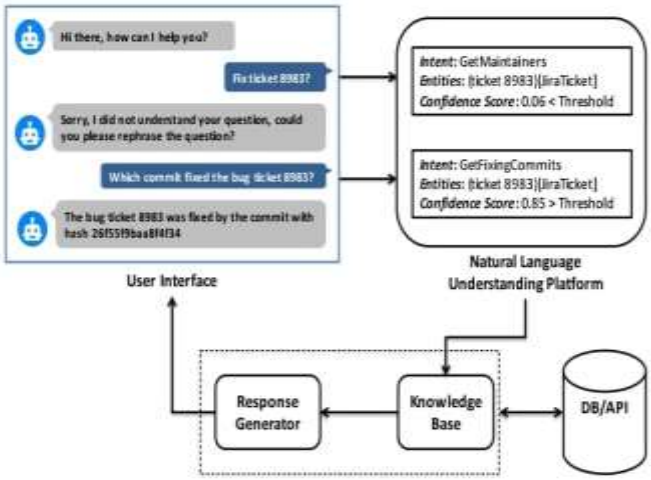
• Rasa: Owned by Rasa Technologies, the sole open-source NLU in our research. The NLU may be configured, deployed, and operated on local servers by developers thanks to Rasa. as a result, processing performance is increased while saving network time in contrast to cloud-based solutions. Rasa-nlu v0.14, the most recent version at the time the trial was conducted, is what we used for our evaluation.

• Microsoft's NLU cloud platform, Language Understanding Intelligent Service (LUIS). LUIS supports the following five programming languages: C#, Go, Java, Node.js, and Python. It also includes a number of prebuilt domains, like music and weather.

II. SE Tasks and Data Corpora

We use two representative data corpora, one for the Repository work and one for the Stack Overflow challenge, in order to assess the performance of the NLUs. Repository corpus, which is utilized for the Repository job and consists of queries practitioners ask chatbots to get information about the software repositories for their projects The Stack Overflow corpus, which is utilized in the Stack Overflow assignment, comprises a collection of posts from discussion threads on Stack Overflow. Ours two primary factors led to the selection of these two tasks: First of all, since developers are inquiring about problems they are having or seeking further details about their projects (such as changing a commit for a bug), both jobs mirror real-world scenarios. To understand the current state of the project repository, project managers frequently ask questions that are covered in the Repository job. As a result, our findings are more applicable to chatbot users in the SE domain because of both tasks. Second, employing two tasks in our analysis allows us to gain a deeper understanding of each NLU's performance in various SE sub-contexts

A.System Architecture



III. Performance Evaluation of NLUs

We train and assess IBM Watson, Dialogflow, Rasa, and LUIS using the corpora from the Repository and Stack Overflow challenges. We describe in the following how we train and test the NLUs for each task, taking into account the unique characteristics of each task.We employ the same training set from the Repository corpus, which consists of 10 intents with their queries and entities with their lists of synonyms, to assess the NLUs on the Repository task. The first step in configuring the NLUs is to set them up to use the list feature for all entities. This means that the NLU won't try to extract any entities that aren't in the training set.This aligns thematically with the nature of the Repository job, which involves a chatbot providing software repository-related replies to inquiries. In this case, the chatbot cannot extract any information for the user from an entity that does not exist in the repository (such as an incorrect Jira ticket number). Next, we define the entity types that are present in the repository corpus—namely, CommitHash and JiraTicket—using the NLUs API. FileName, and make use of DateTime, the fourth built-in entity type.

C. Case Study Results

This section compares how well the NLUs performed on the Repository and Stack Overflow tasks in terms of intent categorization, confidence score, and entity extraction.We use the corpus from both of the SE tasks to train and test each NLU in order to assess its performance in intentions categorization. We only use the highest-scoring intent as the categorized intent when testing the NLUs for two reasons. The first step is to mimic real-world scenarios in which chatbots employ the intent with the greatest score since it has the highest chance of being accurate .Second, because Dialogflow only provides a single intent and matching confidence score for a single query, it is important to make sure that all NLUs are evaluated consistently.

I. NLUs Confidence Scores

As was previously mentioned in Section the NLU assigns a confidence score to each intent classification it does. The NLU's intent classification can be trusted to a certain extent based on the confidence score.NLUs should typically offer high confidence scores for accurately classified intents. The NLU's users can rely on these confidence scores, for instance, if a query asks, "What is the number of commits between 1-July 2020 to 1-August 2020?" and the NLU assigns the question to the Count Commits by Date intent with a high confidence score Conversely, it is also true that one would lose faith in the confidence scores generated by NLUs if they gave high values to intents that were incorrectly classified.

II. Entity Extraction

Chatbots must accurately extract the pertinent entities in order to provide users with accurate answers to their queries. Only when the extracted entity's type and value precisely match the expected entity's type and value for that particular query in the oracle do we consider it to be correct. Our criteria are based on the observation that extracting entities with just partially accurate attributes leads to the chatbot responding to the user's query erroneously.

D.Results And Discussion

In order to better understand the NLUs' confidence score sensitivities and assess their capacity to extract unique entities, we go into the evaluation outcomes in this section. In conclusion, we offer a series of practical suggestions to researchers and chatbot developers in order to improve intent categorization and entity extraction performance.

A.Unique Entities

unique entities had an impact on the NLUs' performance while extracting entities from the Stack Overflow task. Unique entities, as their name implies, only appear once in the dataset used for the Stack Overflow task; hence, the NLUs must forecast their occurrences without any prior training. Because the NLUs have been trained on every entity in the Repository job, it is significant to note that there are no unique entities when analyzing the NLUs using the list feature. We look into the Stack Overflow task results and analyze the NLU performance on queries that contain only unique entities in order to have a better understanding of the NLUs' capacity to extract unique entities. Therefore, our analysis did not include any queries that contained non-unique entities.

B. Recommendations

Drawing from our research findings and expertise, we offer a series of practical suggestions aimed at assisting chatbot operators in enhancing the effectiveness of their chosen neural language unit.to enhance the NLUs' entity extraction and intent categorization capabilities. Despite the fact that our findings are based on SE tasks, some of the recommendations can be applied to any domain to enhance NLU performance. In the following, we go into further depth about the recommendations.

E.Threats To Validity

This section addresses the risks to our study's internal, construct, external, verifiability, conclusion, and repeatability.

I.Construct Validity

takes into account how theory and observation relate to one another in the event that the variables being measured fail to capture the true components. We use the MSRBot corpus, which was developed to assess the MSRBot, to assess the performance of the NLUs in the repository job. There could be certain restrictions on the MSRBot dataset, such as questions (intents) that may not be as well-liked in actual environments. But we contend that the questions that MSRBot answers were developed using a semi-structured procedure that gathered the most typical queries made by software professionals from earlier research. However, participants in the MSRBot evaluation were allowed to formulate their own queries for the chatbot. Lastly, the participants were not given access to the collection of questions that were used to train the MSRBot.

II.Verifiability Validity

relates to whether the study's findings can be verified . In this work, we used two common SE tasks the Repository and Stack Overflow tasks to compare the performance of several NLUs. Different tasks or NLU combinations may produce different results. As a preliminary step to benchmark the NLUs in the SE domain, we chose NLUs that have been used in previous work in order to lessen this threat. Additionally, we examined each task's characteristics in Section , explained our case study setup, and went into detail about our analysis and findings in Section . Lastly, we released the NLU replies , the training/testing dataset, and the scripts that were utilized .

II.External Validity

Relates to the application of our findings generally. There are other NLUs that are not included in our analysis, but we select four of the most widely used NLUs to assess their performance in the SE area. Since identifying the top-performing NLU in the SE domain is our main objective, in this studyWe only choose NLUs that are well-liked by practitioners and researchers, and whose user interfaces and/or API calls can be used for training.The fact that we test the NLUs on the Repository and Stack Overflow tasks may have an impact on our study; as a result, our findings might not apply to other tasks in the SE area. Nonetheless, we think they handle a lot of routine jobs in SE that chatbots may help with.Having stated that, we urge further researchers to carry out comparable investigations that take into account additional NLUs and SE tasks.

III. CONCLUSION

Because software chatbots may save development time and costs, they are gaining popularity in the SE community. Every chatbot is powered by a neural language unit (NLU), which makes it possible to comprehend user input. It can be difficult to choose the optimal NLU for a chatbot that works in the SE domain. In this work, we assess the effectiveness of four popular NLUs: Microsoft LUIS, IBM Watson, Google Dialogflow, and Rasa. We evaluate the NLUs on two distinct tasks derived from Stack Overflow and a repository in terms of intents classification, confidence score, and entity extraction.As a result, we urge academics to create strategies and tactics that improve the NLUs' performance on tasks with various attributes. Our study, we believe, helps chatbot practitioners choose the NLU that best suits the SE task that their chatbots are performing.our research lays the groundwork for more studies in this field.first, as demonstrated by our findings, NLUs typically perform well after receiving additional training examples. In order to improve the performance of the NLUs, we therefore intend to investigate several data set augmentation strategies. Furthermore, we think that more research is necessary to benchmark NLUs in the SE context by comparing various NLUs using a larger number of data sets. We support this endeavor by making our data set available to the public.

ACKNOWLEDGMENT

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.ijsred.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. Badran, and E. Shihab, "MSRBot: Using bots to answer questions from software repositories," Empirical Software Engineering (EMSE), p. To Appear, 2019.

[2] J. Dominic, J. Houser, I. Steinmacher, C. Ritter, and P. Rodeghero, "Conversational bot for newcomers onboarding to open source projects," in Proceedings of the 2nd International Workshop on Bots in Software Engineering, ser. BotSE '20. IEEE Press, 2020.

[3] M. Canonico and L. De Russis, "A comparison and critique of natural language understanding tools," CLOUD COMPUTING 2018, p. 120, 2018.

[4] E. Gregori, "Evaluation of modern tools for an omscs advisor chatbot," 2017.

[5] StackOverflow, "Natural language processing," https://stackoverflow.com/questions/4115526/ natural-language-processing, 03 2020, (Accessed on 03/09/2020).

[6] StackOverflow, "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] StackOverflow, "NLP - build chatbot for education purporse," https://stackoverflow.com/questions/ 52206324/build-chatbot-for-education-purporse, (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.

[9] J. Ask, M. Facemire, and A. Hogan, " The state of chatbots," Forrester.com report, vol. 20, 2016.

[10]IBM, "IBM Watson," https://www.ibm.com/watson, (Accessed on 11/20/2019).

[11] Google,"Dialogflow," https://dialogflow.com/, (Accessed on 02/05/2020).