Challenges in Chatbot Development: A Study of Stack Overflow Posts

Ahmad Abdellatif*, Diego Costa*, Khaled Badran*, Rabe Abdalkareem**, Emad Shihab*
*Data-driven Analysis of Software (DAS) Lab
Concordia University, Montreal, Canada
{a_bdella,d_damasc,k_badran,eshihab}@encs.concordia.ca
**Software Analysis and Intelligence Lab (SAIL)
Queen's University, Kingston, Canada
abdrabe@gmail.com

ABSTRACT

Chatbots are becoming increasingly popular due to their benefits in saving costs, time, and effort. This is due to the fact that they allow users to communicate and control different services easily through natural language. Chatbot development requires special expertise (e.g., machine learning and conversation design) that differ from the development of traditional software systems. At the same time, the challenges that chatbot developers face remain mostly unknown since most of the existing studies focus on proposing chatbots to perform particular tasks rather than their development.

Therefore, in this paper, we examine the Q&A website, Stack Overflow, to provide insights on the topics that chatbot developers are interested and the challenges they face. In particular, we leverage topic modeling to understand the topics that are being discussed by chatbot developers on Stack Overflow. Then, we examine the popularity and difficulty of those topics. Our results show that most of the chatbot developers are using Stack Overflow to ask about implementation guidelines. We determine 12 topics that developers discuss (e.g., Model Training) that fall into five main categories. Most of the posts belong to chatbot development, integration, and the natural language understanding (NLU) model categories. On the other hand, we find that developers consider the posts of building and integrating chatbots topics more helpful compared to other topics. Specifically, developers face challenges in the training of the chatbot's model. We believe that our study guides future research to propose techniques and tools to help the community at its early stages to overcome the most popular and difficult topics that practitioners face when developing chatbots.

ACM Reference Format:
Ahmad Abdellatif*, Diego Costa*, Khaled Badran*, Rabe Abdalkareem**, Emad Shihab*. 2020. Challenges in Chatbot Development: A Study of Stack Overflow Posts. In *17th International Conference on Mining Software Repositories (MSR '20), October 5–6, 2020, Seoul, Republic of Korea.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3379597.3387472

MSR '20, October 5–6, 2020, Seoul, Republic of Korea
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7517-7/20/05…$15.00
https://doi.org/10.1145/3379597.3387472

1 INTRODUCTION

More than 50 years after Weinzebaum introduced the first computer program to have a conversation with humans [68], chatbots have become the main conduit between humans and services [58]. Potentialized by the recent advances in artificial intelligence and natural language processing [32], chatbots are the primary interface in a variety of services, from smart homes [10, 65] and personal assistants [8, 28], to health care [18] and E-commerce[59]. Given how chatbots reduce the operational costs of services, the usage of chatbots will only increase - experts predict that 85% of users' interactions with services will be done through chatbots by 2021 [40].

Due to their importance and popularity, developing and maintaining chatbots is becoming more important. In addition, the development of chatbots requires expertise in specialized areas, such as machine-learning and natural language processing, which, distinguishes it from traditional software development [19]. While recently introduced chatbot frameworks (e.g., Microsoft Bot Framework [38]) have reduced the barrier to entry of creating chatbots, e.g., by providing the components for user interaction and natural language understanding platforms, little is known about the specific challenges that chatbot developers face when developing chatbots. Understanding such challenges is of paramount importance, helping the research community provide more effective tools for chatbot development, improving their quality, and ultimately increasing their adoption and usefulness among users.

In this paper, we provide the first attempt at understanding the challenges of chatbot development by investigating what chatbot developers are asking about on Stack Overflow. We study Stack Overflow since it is the most prominent code-centric Q&A website and used constantly by the development community to communicate their challenges and issues, provide solutions and foment discussions about all aspects in software development [2, 52]. Our investigation dives into the chatbot-related posts on Stack Overflow to pinpoint the major topics surrounding the discussions on chatbot development. We use well-known topic modeling techniques to group the posts into cohesive topics and apply a series of quantitative analyses, both through metrics and manual analysis. Specifically, our work investigates the following research questions:

- **RQ1: What topics are chatbot developers asking about?** We find that chatbot developers ask about 12 main topics that can be grouped into 5 main categories. The categories are related to chabot integration, development, natural language understanding (NLU), user interaction, and User Input.

The most popular questions include those related to chatbot creation, integration, and user interface.

- **RQ2: What types of questions are chatbot developers asking?** Chatbot developers use Stack Overflow primarily as a source of guidance for specific implementation routines, working examples, and troubleshooting. This shows a need for better documentation that provides real-scenarios and more information about the NLU models used by chatbots.
- **RQ3: Which topics are the most difficult to answer?** The most difficult topics are related to training the chatbot NLU models. On the other hand, posts related to traditional software development, e.g., chatbot development framework, are more frequently answered, albeit, we did not find any statistically significant correlation between the popularity and difficulty of the chatbot topics in our study.

In addition to the identified chatbot topics in Stack Overflow, we discuss the evolution of the chatbot topics on Stack Overflow and find that the chatbot-related discussions have increased substantially since 2016. The activity of some categories are linked to the releases of chatbot platforms. Also, we compare the chatbot topics to other mature SE fields (e.g., mobile and security) in terms of popularity and difficulty. Our results show that the chatbot community needs more effort to reach the maturity level of similar SE fields.

Our findings show that platform owners need to improve their current documentation and integration with popular third-parties. Moreover, we believe that our study guides future research to focus on the most popular and challenging chatbot topics.

Paper Organization. The rest of paper is organized as follows. Section 2 describes our methodology. Section 3 reports our empirical study results. Section 4 discusses our results and the implications of our findings. Section 5 presents the related work to our study. Section 6 discusses the threats to validity, and Section 7 concludes the paper.

2 METHODOLOGY

The main *goal* of our study is to examine what chatbot developers are asking about. To achieve this goal, we resort to analyze the developers' discussions on Stack Overflow as it provides a rich dataset and have been used by similar investigations in other domains, such as concurrency [6], cryptography APIs [41], and deep learning [30]. While providing structured data with questions, answers and their respective metadata (e.g., accepted answers), Stack Overflow does not contain any fine-grained topic information related to chatbots. Hence, we first need to identify the posts from Stack Overflow that are related to chatbots, group them according to their dominant topic, and then conduct our analysis. As Figure 1 shows, we perform the selection of chatbot related posts in a methodology of five-steps, which will be detailed further in this section.

Step 1: Download & extract Stack Overflow dump. We download the entire Stack Overflow dump (last updated 4 September 2019) [23], containing user questions, answers, and the metadata of the posts (e.g., view count, creation date) for the period between August 2008 and September 2019. The initial dataset contains approximately 18 million questions and 28 million answer posts.

Step 2: Identify chatbot tags. Stack Overflow holds posts on a myriad of different software development topics (e.g., Java, security, and blockchain). Posts are typically tagged by their authors with commonly used tags (e.g. chatbot, web) to improve the posts' visibility and chances of being answered [13]. To identify the most relevant chatbot-related tags, we follow the approach used by prior work [11, 54], and create a *tag set* using the following procedure. First, we retrieve all posts with the 'chatbot' tag, yielding a set of 2,116 posts. We refrain from adding any other tags in this inital step to reduce the chances of introducing noise, as this will be used to identify other chatbot-related tags. Second, we extract all the tags that co-exist with the 'chatbot' tag from the chatbot-tagged posts. Next, we use two heuristic metrics used in prior work to obtain a bigger set of chatbot-related tags [54, 67]. The first metric is the *tag relevance threshold (TRT)*, a measure of how related a specific tag is to the chatbot-tagged posts. This measure calculates the ratio of the chatbot-related posts (posts that include the 'chatbot' tag) for a specific tag compared to the total number of posts for that tag. Specifically, the TRT is measured using the equation $TRT_{tag} = \frac{No. of\ chatbot\ posts\ for\ the\ tag}{Total\ no. of\ posts\ for\ the\ tag}$. For example, 'rasa' is a tag with a TRT of 21.2%, which means that 21.2% of the posts tagged with 'rasa' are also tagged with 'chatbot'. By using the TRT we are able to eliminate the irrelevant tags from our set.

However, some tags that have a small number of posts (e.g., the 'botlibre' tag has only 3 posts) can have a high TRT of (33.3%) because a single one of their posts is chatbot-related, and this may introduce insignificant tags. Therefore, we use a second metric, the *tag significance threshold (TST)*, which is a measure of how prominent a specific tag is in the chatbot-tagged posts [54, 67]. This metric is measured by using the total number of the chatbot posts for that tag and the total number of the chatbot posts for the most popular tag ('chatbot' tag with 2,116 posts.) as follows $TST_{tag} = \frac{No. of\ chatbot\ postsforthetag}{No. of\ chatbot\ posts\ for\ the\ most\ popular\ tag}$. For example, the 'rasa' tag has a TST of 0.3% which means that the total number of the posts that are tagged with 'rasa' and 'chatbot' at the same time are equal to 0.3% of the total number of chatbot-related posts for the 'chatbot' tag.

We consider a tag to be significant and relevant to the chatbot posts if its corresponding TRT and TST are above a certain threshold. The first three authors, with varying degrees of chatbot development experience, independently examined the tags with different TRT and TST thresholds. For each tag, we inspect a randomly selected sample of posts, to identify when the tags become less relevant and less specific to chatbots, to identify the most appropriate TRT and TST thresholds. This method has been used by several previous similar studies [11, 54] and has the goal of selecting tags relevant to chatbots without including too much noise in the dataset. Then, we discussed the chosen thresholds to reach a consensus on the optimal TRT and TST values. The first three authors independently evaluated the optimal TRT and TST thresholds that yield the best results and discussed their choices to reach a consensus. We find that tags with a TRT value higher than 11% and a TST value higher than 0.14% value yield an appropriate balance between the inclusion of more posts related to chatbots (i.e,. more representative dataset) and the filtering of posts that are unrelated to chatbots (i.e., less noise). It is important to note that our thresholds are in-line with the thresholds used by previous studies that adapted the same approach [6, 11, 72]. Finally, we use the selected TRT and TST thresholds to identify our tag set. Table

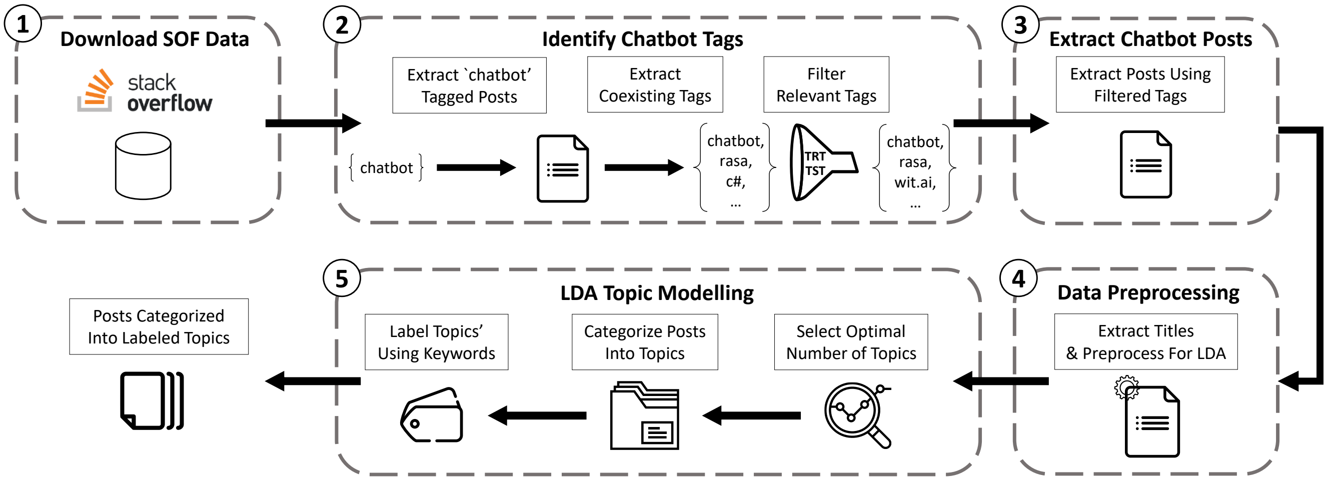
1
Download SOF Data
stack overflow
2
Identify Chatbot Tags
Extract `chatbot' Tagged Posts
Extract Coexisting Tags
Filter Relevant Tags
{ chatbot }
chatbot, rasa, c#, ...
TRT
TST
chatbot, rasa, wit.ai, ...
3
Extract Chatbot Posts
Extract Posts Using Filtered Tags
4
Data Preprocessing
Extract Titles & Preprocess For LDA
5
LDA Topic Modelling
Label Topics' Using Keywords
Categorize Posts Into Topics
Select Optimal Number of Topics
Posts Categorized Into Labeled Topics

Figure 1: Overview of the methodology of our study.

Table 1: The tag set used to identify the chatbot related posts. The TRT and TST are expressed in percentages.

Tag Name	TRT	TST	Tag Name	TRT	TST
chatbot	100	100	aws-lex	14.3	0.6
facebook-chatbot	42.1	6.2	sap-conversational-ai	50	0.5
amazon-lex	22.2	4.3	chatfuel	26.3	0.5
rasa-nlu	18.4	2.9	pandorabots	41.2	0.3
aiml	27.6	2.6	rasa	21.2	0.3
rasa-core	22.6	2.4	chatbase	18.2	0.3
wit.ai	13.1	1.9	chatscript	30.8	0.2
chatterbot	25.4	1.6	rivescript	28.6	0.2
api-ai	11.4	0.8	program-o	37.5	0.1
web-chat	13.6	0.8	botpress	33.3	0.1
gupshup	27.1	0.6	lita	25	0.1

1 shows the tags obtained in our tag set and their respective TRT and TST values.

Step 3: Extract chatbot posts. After obtaining the chatbot-related tag set, we use those tags (see Table 1) to extract the posts that will constitute our chatbot dataset throughout this study. We extract this corpus by querying all posts on Stack Overflow that are tagged with one of the tags in our tag set. This process yielded a dataset containing 3,890 chatbot posts and their respective metadata.

Step 4: Preprocessing chatbot posts. We filter out the irrelevant information before applying the topic modeling techniques. In this analysis, we focus only on the posts' titles, as opposed to their body contents, as the content in the posts' bodies can introduce noise to our analysis. This approach of using the posts' titles has been used in the prior investigations [54], as a post's title has been shown to be representative of the post body [20, 71]. After extracting the posts' titles, we prepare the data to be used in the topic modelling process. To do so, we leverage the Python NLTK [42] and Gensim [26] tools to perform the preprocessing steps on our dataset. First, we remove the stopwords, such as 'how', 'a' and 'can', using the NLTK stopwords corpus [43] as those words hinder the process of differentiating between topics. Next, we build a bigram model using Gensim since we notice that some words commonly appear together (e.g., 'Rasa NLU' and 'Bot Framework') and the topic modelling technique should consider them together. Moreover, we lemmatize the words to map them to their origin (e.g., 'development' is mapped to 'develop'). Those steps output a preprocessed dataset that is ready to be inputted to the topic modelling technique in our next step.

Step 5: Identify chabot topics. To identify the topics that are discussed by chatbot developers on Stack Overflow, we use the Latent Dirichlet Allocation (LDA) modeling technique [14], which has been widely used in Software Engineering studies [11, 54]. LDA groups the posts of our dataset into a set of topics based on the word frequencies and their co-occurrences in the posts. In particular, LDA assigns to each post a series of probabilities (one per topic) that indicate the chances of a post being related to a topic. The topic with the highest probability for a particular post (i.e., the post that contains more keywords of a particular topic) is considered to be the post's dominant topic. We use the Mallet implementation of LDA in our methodology [35].

The main challenge of using LDA is to identify the optimal number of topics K, that the LDA uses to group the posts. If the K value is too high, topics may become too specific to draw any relevant analysis. On the other hand, if K value is small, the yielded topics may be too generic, encompassing posts of many different aspects. To overcome this issue, we examine different K values ranging between 5 to 20 in steps of 1 and calculate the coherence metric value of the topics. The coherence metric measures the understandability of the topics resulting from the LDA using different confirmation measures, and has been shown to be highly correlated with human understandability [53]. Thus, the first two authors run the LDA with varying K values and then stored the resulting coherence score from each run. We find that K values in the range of 10 to 14 have very similar coherence scores (i.e., the difference is very small). To ensure that we select the best K value, the first two authors examined a randomly selected sample of 30 posts from each topic for K values from 10 to 14. Based on this examination, we find that

Table 2: The chatbot topics, categories, and their popularity.

Main Category	Topic	# Posts	Avg. Views	Avg. Favourites	Avg. Scores
Integration	API Calls	264	354.2	1.2	0.5
	Messenger Integration	**463**	638.0	1.4	0.7
	NLU Integration/Slots	388	406.0	1.1	0.8
Development	General Creation/Integration	250	**671.6**	**3.1**	0.6
	Development Frameworks	375	513.3	1.6	0.8
	Implementation Technologies	320	619.2	1.5	0.7
NLU	Intents & Entities	437	516.3	1.7	**1.0**
	Model Training	347	524.3	1.4	0.7
User Interaction	Chatbot Response	253	409.1	1.2	0.7
	Conversation	278	510.5	1.9	0.6
	User Interface	208	536.8	2.6	0.8
User Input	User Input	307	402.7	1.2	0.6

a K value of 12 (i.e., 12 topics) provides an optimal set of topics that balances the generalizability and the specificity (i.e., most cohere posts) of the resulting chatbot topics.

3 CASE STUDY RESULTS

In this section, we present the analysis of the chatbot posts and topics to answer our research questions.

3.1 RQ1: What topics are chatbot developers asking about?

Motivation: Chatbot development has some particularities that distinguish it from traditional software development [19]. For example, chatbot developers require specific expertise in natural language processing, machine learning, and conversation design, which are often unnecessary or overlooked in most conventional software development tasks. Hence, the challenges faced by chatbot developers are likely to differ from the challenges of traditional software development. Since developers use Q&A websites to communicate both problems and solutions, the goal of this research question is to dive into the invaluable data of Stack Overflow to identify the most common and pressing chatbot topics and the issues that are more frequently encountered by the chatbot community. Moreover, identifying the widely discussed chatbot topics is the initial step to highlight the topics that are gaining more traction and difficult to answer by the chatbot community.

Approach: We use the LDA as a method to identify the different topics that developers discuss on Stack Overflow as mentioned in Section 2. The first three authors (annotators) labelled the set of topics based on the posts overall theme. In particular, each of the annotators individually inspected the top 20 keywords and a random sample of at least 30 posts from each topic in order to label it with a title that best represents the posts of that topic. Then, the authors discuss each of the 12 topics' labels to reach a consensus about the titles of all topics. We observe that some topics that discuss similar aspects of the chatbot development process or are related to the same chatbot component can be further grouped into categories. For example, one topic with keywords related to 'response', 'webhook', and 'card' and another topic that has 'display', 'trigger', and 'prompt' keywords are related to chatbot user interaction. Therefore, we further categorize those topics to have a hierarchical view on the chatbot discussions on Stack Overflow. We also examine the most popular chatbot topics among developers. To investigate that, we use three different complementary measurements of popularity that have been adopted in prior work [6, 11, 12, 41]:

(1) **The average number of views (avg. views)** of the post from both registered and unregistered users. Our intuition here is that if a post is viewed by a large number of developers, then this post is popular among chatbot developers. Overall, this metric measures the interest of the community by telling us how often a post is visualized.
(2) **The average number of posts marked as favourite (avg. favourites)** by Stack Overflow users. This metric measures the issues and solutions that developers deemed to be helpful and having a high chance of recurring during the development of chatbots.
(3) **The average score (avg. scores)** of the posts. Stack Overflow allows it's members to up-vote posts that they consider to be interesting and useful. The votes are then aggregated as a score, which we use as a metric of perceived community value.

Results: Table 2 shows the 12 topic titles, which are grouped into 5 main categories. It also shows the number of posts that belong to each topic and the topics' popularity through our popularity metrics: views, favourites, and the scores received by developers on Stack Overflow. As seen from the table, the developers ask about different topics in chatbot development and the number of posts varies across the topics.

The 12 chatbot topics can be mainly grouped into five categories: 'Integration', 'Development', 'NLU', 'User Interaction', and 'User Input'. Next, we discuss those categories in more details.

Integration: This category contains three topics, namely Messenger Integration, NLU Integration/Slots, and API Calls. This category deals with the integration between chatbot platforms, APIs, and websites. About 28.6% of posts in our dataset belong to this category. We also see that the Messenger Integration topic has the highest number of posts in our dataset. In this topic, developers mainly ask about how to create and integrate chatbots to messenger applications. One of the reasons of the widespread of chatbots is the global adoption of messaging platforms (e.g., Slack) [32]. For example, Facebook reported that there are more than 300,000 active chatbots in 2018 that are deployed on its Messenger platform [15]. An example of posts under this topic is a developer asking on Stack Overflow “Facebook Chatbot (PHP webhook) sending multiple replies”[47]. As chatbots are used to integrate various services [32], chabot developers are more exposed to the challenges of multi-service and platform integration.

Development: The posts of this category are related to building chatbots using different development frameworks, asking about special configurations and features, and specific implementations using those frameworks. For example, a developer posted on Stack Overflow “How to start a conversation from Nodejs client to Microsoft bot”[44]. The posts of Development Frameworks, Implementation Technologies, and General Creation/Integration topics form this category. In our study, this category is the second largest, containing 24.3% of the posts in our dataset. This shows that developers tend to heavily rely on chatbot frameworks.

Natural Language Understanding (NLU): This category contains posts related to the definition of intents (the purpose/intention behind the user’s input) and entities (important pieces of information in the user’s input such as city names), handling and manipulating those intents and entities, customizing and configuring NLUs, and improving the performance of the NLU models. This category comprises 20.2% of the posts in our dataset. It has Intents & Entities and Model Training topics. Those topics are related to the chatbot capability of understanding the users’ input and replying accordingly, which has a direct impact on user satisfaction [3]. The post “How can I improve the accuracy of chatbot built using Rasa?” [46] is an example of posts from this category. Currently, large IT companies are investing to build NLUs (e.g., Microsoft developed LUIS platform [37]), which is an indicator of their importance and popularity. Moreover, NLU platforms nowadays are considered to be one of the critical components of chatbots [55]. Leveraging an NLU platform allows developers to focus on the core functionalities of their chatbots rather than having to analyze the user input and manage the conversation with the user.

User Interaction: This category contains posts about conversation design, generating reply messages to users, and designing the chatbot’s graphical user interface. For example, developers ask “How to resume or restart paused conversation in RASA?”[51] and “How to add custom choices displayed through Prompt options [...] using C#?”[48]. This category includes User Interface, Chatbot Response, and Conversation topics and forms 19% of the posts in our dataset. We believe that managing the conversation flow with the user is not an easy task since the chatbot users might deviate (i.e., change to other topic) from the designed conversation flow.

User Input: The posts of this category are related to checking/-validating and storing the user input, e.g., “How to store and retrieve the chat history of the dialogflow?”[50]. There is only one topic that is included in this category and it contains 7.9% of posts in our dataset. Having a single topic as a group indicates that parsing and storing chatbot users’ input is a more independent problem among the chatbot topics.

From our results, we observe that the categories cover the end-to-end development of chatbots. The User Interaction category covers the creation of the chatbot interface, while the User Input category covers the manipulation of the users’ input received through the User Interaction component. The NLU category includes posts about understating the users’ input and optimizing the NLU Model of the chatbot, the Development category covers the back-end development of the core functionalities of the chatbot, and finally, the Integration category covers the integration of all the chatbot components together (User interface, NLU, backend, etc.). This shows that developers are facing various challenges and seeking knowledge about each phase of the chatbot development process. Moreover, the topics within each category reflect specific concerns and issues within that category. For example, in the NLU category, developers are asking questions about defining/handling intents and entities, and improving the performance of the NLU model.

In the second part of our analysis, we investigate the popularity of the chatbot topics. We find that the most popular topics fall into the Development and NLU categories. Table 2 shows that the topic General Creation/Integration contains the most viewed and most favourited posts by chatbot developers. This topic contains posts with basic questions about chatbot creation and its high popularity can be explained by the introductory nature of the topic, that is, any newcomer will look for these posts to start developing their first chatbot. Another aspect of this topic’s popularity might be due the lack of proper chatbot introductory documentation and support for newcomers. The most viewed and favourited post in our dataset is “Any tutorials for developing chatbots?” with more than 71,565 views and 104 members marking it as a favorite post, evidences the lack of documentation concern. Interestingly, our findings suggest that the chatbot development community should give special attention to providing a more extensive and accessible documentation on how to develop chatbots from scratch. Intents & Entities is the topic with highest average of post score, the process of handling intents and entities is one of the most specialized aspects of chatbot development, which might explain why developers have a higher (relative) praise for posts from this particular topic.

Chatbot developers ask about every aspect and phase of the chatbot development process including Integration, NLU, Development, User Input, and User Interaction. The most popular topics in the chatbot dataset are related to General Creation/Integration.

3.2 RQ2: What types of questions are chatbot developers asking?

Motivation: After understanding the most interesting topics to chatbot developers, we set out to examine the types of posts that they ask in each chatbot category. Prior work [54] shows that developers ask different types (i.e., how, why, what) of questions to address distinct challenges, hence, this analysis will help us identify the nature of the challenges encountered during chatbot development.

Approach: To achieve that, we follow a similar approach used by prior work to identify the types of the posts on Stack Overflow [54, 63]. In particular, we randomly sample posts from each of the five main chatbot categories with a confidence level of 95% and a confidence interval of 5%. Our random sample size for each

Table 3: Chatbot posts types on Stack Overflow.

Main Categories	% How	% Why	% What	% Other
Integration	66.4	22.7	10.8	0.0
Development	57.9	23.4	18.3	0.4
NLU	54.3	29.5	15.9	0.4
User Interaction	66.8	22.5	10.3	0.4
User Input	68.4	14.6	14.6	2.3
Chatbots (all)	61.8	25.4	11.7	1.2

category yields a total of 1241 posts: 286 Integration posts, 273 Development posts, 258 NLU posts, 253 User Interaction posts, and 171 User Input posts. Overall, the annotators achieve substantial agreement (kappa=0.62) on the 1241 classified posts. Our level of agreement is higher than the agreement reached in similar studies [54]. For the cases that all annotators failed to agree on, the annotators revisit the questions together and discussed them to reach an agreement. Then, the first three authors individually examine the sample posts' titles and bodies and label each post using one of following types that were used by prior work [54]:

- **How**: Used for posts that ask about a method or technique to implement something [54]. Posts with this type differ from the 'why' posts as in here the developer has a particular goal in mind, and asks for the steps to achieve this goal (e.g., "how to get user name in Microsoft bot framework in C# using V4?").
- **Why**: Posts where the developer asks about the reason, cause, or purpose of something [54]. Posts of 'why' type are often related to troubleshooting where the developer expects an explanation of a particular (and unexpected) behavior (e.g., "why is Wordpress blocking the js livechat window?").
- **What**: Posts where the developer is asking for a particular information [54]. Often, the user wants to clarify a doubt that is useful to make more informed decisions (e.g., "what are "implicit triggers" in a Google Action package?").
- **Other**: We assign this type to posts that do not fall under any of the above types (e.g., "chatbot conversation objects, your approach?").

To measure the quality of our classification of the random sample, we use Cohen's Kappa [36] to measure the level of inter-agreement among the annotators.

Results: Table 3 shows the percentage of the posts types for each chatbot category. We see that more than half of the posts (61.8%) are of 'how' type, followed by 'why' (25.4%) and 'what' (11.7%). This shows that the developers are looking for more working examples, debugging, and information. The User Interaction category has the most 'how' posts (66.8%), showing a need for more sources of guidance to design and manage the conversation flow between the user and chatbot. The NLU category has the most 'why' posts (29.5%), suggesting the need for discussion forums and better documentation on how the NLU models work, especially given that most NLUs are closed source. The Development category has the most 'what' posts (18.3%), suggesting that providing general information about the supported features of the chatbot frameworks is appreciated by the community.

Table 4: The difficulty per topic.

Topic	Posts w/o Accepted (%)	Median Time (h)
General Creation/Integration	72.0	8.2
Intents & Entities	71.4	19.5
User Interface	70.7	7.0
Model Training	70.2	22.4
Messenger Integration	70.0	22.6
User Input	66.8	9.3
NLU Integration/Slots	66.5	12.8
Conversation	65.5	6.9
Chatbot Response	65.2	11.3
Implementation Technologies	64.7	15.5
API Calls	63.7	16.2
Development Frameworks	63.7	15.6

> ***Chatbot developers mainly (61.8%) look for implementation guidance by posting how posts, followed by why (25.4%) and what (11.7%). Developers are concerned about the how aspect of the User Interaction category, whereas most the highest share of why posts are from the NLU category, and what posts from the Development category.***

3.3 RQ3: Which topics are the most difficult to answer?

Motivation: Given that we know the popular topics and their types of posts. Now, we want to investigate the difficulty of answering posts in each topic. Finding whether some topics are harder to answer than others will help us identify the topics that need more attention from the community. Also, it allows us to highlight the topics where there is a need for better tools/frameworks to support developers at addressing chatbot development challenges.

Approach: We measure the difficulty of each topic by applying two metrics that have been used in prior work [11, 54, 72]:

(1) **The percentage of posts of a topic without accepted answers (% w/o accepted answers)**. For each chatbot topic, we measure the percentage of posts that have no accepted answers. While many answers can be issued in a post, the post's author has the sole authority to mark an answer as accepted if it satisfies and solves the original post's question. Therefore, topics with less accepted answers are considered more difficult [11, 54].

(2) **The median time in hours for an answer to be accepted (Median Time to Answer (Hrs.))**. We measure the median time in hours for posts to receive an accepted answer. This metric considers the creation time of the accepted answer and not the time at which the answer is marked as accepted. The longer it takes for a post to be properly answered (receive an accepted answer), the harder the post is[11, 54].

Our dataset includes some posts that did not have sufficient time to receive an answer. In our dataset of chatbot-related posts, questions take a median of 14.8 hours to be answered, hence, we remove from this analysis posts that were created less than 14.8 hours before the data collection date (September 4, 2019).

Table 5: Correlation of topics popularity and difficulty.

Correlation Coeff. / p-value	Avg. Views	Avg. Score	Avg. Favourite
% w/o Accepted Answers	0.524/0.084	0.147/0.651	0.419/0.176
Median Time to Answer (Hrs.)	0.105/0.749	0.223/0.485	−0.335/0.287

Results: Table 4 shows the percentage of accepted answers and median time (in hours) to receive an accepted answer for each of the identified topics in Section 3.1. The topics in Table 4 are ordered based on the percentage of accepted answers they received. The most popular topic General Creation/Integration is also the one with the largest share of posts without accepted answers. The posts in this topic, however, take a median time of only 8.2 hours to receive an accepted answer, which is the third fastest median time in our topics. To understand the reason behind the high percentage of posts with no accepted answers (72%), we examine the posts of this topic. We find that the posts without an accepted answer are given low scores (on average 0.17) from developers on Stack Overflow. This might be due to unclear or ill-formed questions, which effectively reduces the chances of getting an accepted answer.

If we analyze the median time to answer a topic, we see a higher variation among the topics. Messenger Integration, Intents & Entities, and Model Training are the most difficult topics based on their time to receive accepted answers. Interestingly, Intents & Entities, and Model Training are related to the NLU category which discusses how to load and train NLU models, and identify and handle intents and entities. The results show that the topics related to the NLU are harder to answer by the Stack Overflow community. This may be due to the black box implementation of most popular NLUs, which prevents chatbot developers from fully understanding and solving NLU related issues.

On the other hand, posts that are related to Development Frameworks have the highest percentage of accepted answers and a median time to answer in-line with the overall chatbot topics (15.6 hours). This topic includes posts on how to implement chatbot routines using a certain technology (e.g., “How to send location from Facebook messenger platform?”) or comparing of different platforms (e.g., “Comparison between Luis.ai vs Api.ai vs Wit.ai?”). These are also tasks that are more closely related to traditional software development, which could explain why the Stack Overflow respondents tend to answer this topic faster and more frequently.

To have a full view of the chatbot-related posts, we want to examine if there is a statistically significant correlation between the difficulty and popularity. In particular, we use the Spearman Rank Correlation Coefficient [57] to verify the correlations between the three popularity metrics (avg. views, avg. favourites, and avg. scores) and the two difficulty metrics (% w/o accepted answers and median time to answer). We choose Spearman’s rank correlation since it does not have any assumption on the normality of the data distribution. As shown in Table 5, we do not find any statistically significant correlation between the popularity and difficulty metrics since all correlations have $p - value > 0.05$. In other words, the difficult topics are not necessary popular among developers, and vice versa.

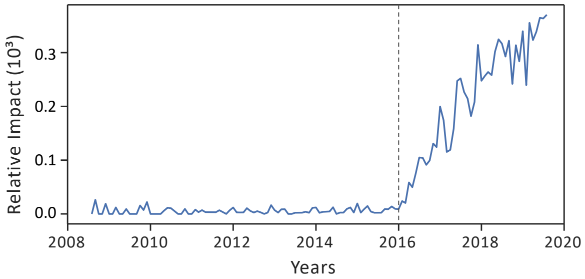

Figure 2: Relative growth of chatbot related posts over time.

Topics related to training chatbot models are the most difficult in chatbot development. While the most popular topic, General Creation/Integration, contains the largest share of unanswered posts. On the other hand, posts related to the Development Frameworks topic tend to be answered more frequently.

4 DISCUSSION & IMPLICATIONS

In this section, we discuss the chatbot topics evolution and compare our findings with the findings in prior work. Then, we delve into the data to identify the prevalent topics on different platforms and discuss the implications of our results.

4.1 Chatbot Topics Evolution

Chatbots are an emerging topic that is getting more attention from developers in different domains (e.g. security [22], software engineering [62]). To examine the evolution of a topic, we utilize two measures; the absolute growth, which measures the change in the total number of posts over time; and the relative growth, which represents the relative change in the total number of posts for a specific topic compared to the change in the total number of posts for the entire Stack Overflow dataset. To highlight the evolution of the chatbot topics, we examine the relative growth of all chatbot topics compared to Stack Overflow over time, from August 2008 to September 2019. Figure 2 shows the evolution of the chatbot in terms of relative growth compared to Stack Overflow. As seen from the Figure, the relative growth of the chatbot topics has an increasing trend that started in 2016. This increase in the last few years shows that chatbots are gaining more attention from the community over time.

To better understand the evolution of the different chatbot development activities, we measure the absolute growth of each of the five categories over time. We find that all of our categories are growing positively over time as shown in Figure 3. This means that the number of posts for every category is increasing overtime, which in turn indicates the increasing trend of the various chatbot development activities represented by the different categories.

We further investigate the reasons behind the sudden increases (i.e., hikes) in the number of posts during specific periods of time and find two interesting cases as shown in Figure 3. The first case is related to the Integration category which has the highest spike (46 posts) on June 2017. We find that most of the discussions during this spike are related to the integration of the Amazon Lex platform [7]

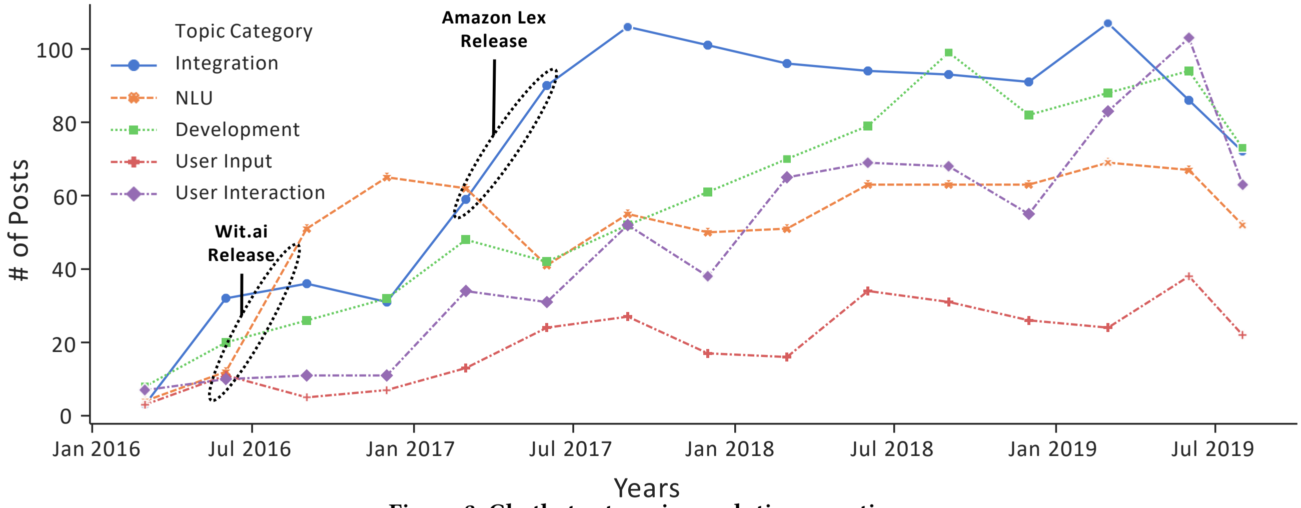

Figure 3: Chatbot categories evolution over time.

Table 6: Comparison of popularity and difficulty between different fields

Metrics	Chatbot	Mobile	Security	Big Data
# of Posts	3,890	1,604,483	94,541	125,671
Avg. ViewCount	512.4	2,300	2,461.1	1,560.4
Avg. FavoriteCount	1.6	2.8	3.8	1.9
Avg. Score	0.7	2.1	2.7	1.4
Avg. AnswerCount	1.0	1.5	1.6	1.1
% w/o Answers	67.7	52	48.2	60.3
Med. TimeToAnswer (Hrs.)	14.8	0.7	0.9	3.3

that was released in April 2017 [9]. The second sudden increase can be observed in the NLU category during November 2016. Posts of that spike are asking about the intents and entities in the Wit.ai platform [24], which was released in April 2016 [60].

Although we show the results of the chatbot categories' evolution over time, we share the evolution results of each of the topics in a publicly available online dataset [56]. In general, we can see a trend of chatbot development activities gaining traction among developers. Our findings also show that the chatbot community tends to pick up the new platforms as shown in the cases of Amazon Lex and Wit.ai.

4.2 Chatbot Compared to Other SE Fields

In the previous sections, we find that chatbot discussions only started to become more active in 2016. As a new and emerging field, we set out to investigate how the topics of chatbot compares against discussions of more consolidated Software Engineering (SE) fields such as mobile, big data and security (topics that were similarly studied in the past). To answer this question, we examine the difficulty and popularity of the chatbot topics and compare it against other disciplines, by including data from similar studies on Stack Overflow, focused on the topics of mobile apps [54], security [72], and big data [11]. Those studies were conducted in a different time frame, therefore, we use their reported keywords to construct an updated dataset and calculate the popularity and difficulty metrics for each of those fields.

Table 6 shows the results of the popularity and difficulty metrics among the four fields. From the sheer number of posts, the chatbot topic is, by a few orders of magnitude, smaller than mobile, security and big data. Second, the chatbot posts are consideranly more difficult compared to the other fields, which is also a consequence of having a small and niche crowd. There is a big gap in the time to receive an accepted answer for the chatbot-related posts compared to other topics. Most mobile and security posts are answered in less than an hour, while most chatbot posts take at least 14 hours. This corroborates with the emerging nature of the chatbot topic and indicates that much needs to be done to put the chatbot development community on pair with other mature fields such as mobile and security.

4.3 Implications

The results of our study can help chatbot community at better focusing their efforts on the most pressing issues in chatbot development. In the following, we describe how our results can be used to better guide practitioners, researchers and educators at improving the practice and learning of chatbots development.

To help identify the most pressing issues, we present in Figure 4 a bubble plot that positions the topics in terms of their popularity and difficulty. The size of the bubble represents the number of posts for a particular topic and we visually split the figure into four quadrants to show the relative importance and difficulty of the topics. We use the average number of views as a proxy for popularity and the percentage of posts without accepted answers as a proxy for difficulty.

Implication for Practitioners. As shown in Figure 4, albeit being the most popular topic, beginner questions on how to build Chatbots (General Creation/Integration) remain largely unanswered. The development community should use this finding to devise better tutorials and documentation aiming at reducing the entry-barrier for developing chatbots.

Our findings can help chatbot developers better prioritize their work by taking into account the areas of the most difficult topics in chatbot development. Topics related to NLU, such as Model Training and Intents & Entities, are among the topics with the highest share of posts without accepted answers. Software managers can take that into account by assigninig more resources (development time) to tasks that involve training NLU models, especially given that NLU has the highest share of troubleshooting posts (Section 3.2), indicating that developers experience issues more frequently with this kind of tasks.

The evidence of the difficulty of NLU related topics can be used to motivate better and more intuitive NLU frameworks. Practitioners can improve the current documentation of the NLU frameworks and companies that develop and publish NLU platforms should focus on improving the expressiveness of their current framework APIs. For instance, some platforms (e.g., Google DialogFlow [27] and Microsoft LUIS [37]) offer graphical interface for training the NLU model, in an attempt to extend the model training to users less familiar to software programming [21, 39].

Figure 4 also shows that Messenger Integration is the largest topic in our dataset. In fact, Integration is the category with the highest number of posts in Stack Overflow. Chatbots are expected to communicate between multiple services and integrate with messengers to make use of already existing Social Networks platforms (e.g., Facebook). Practitioners should invest more resources into facilitating integration of their platforms and tools with other services. For instance, Dialogflow offers developers a one-click integration feature to some of the most popular chatting platforms, such as Slack, Twitter and Skype [29]. As chatbot developers find integration a pressing issue, providing straightforward approaches to integration would allow developers to focus on the core chatbot functionalities, reducing the time and effort overhead of developing multi-service chabots.

Implication for Researchers. Our findings confirm that chatbot developers discuss topics such as Conversation, NLU Integration/Slots, and Chatbot Response, that differ chatbot development from traditional software development. As shown in Figure 4, NLU related topics are notoriously difficult and research can be put into some of the problems faced by chatbot developers at training their NLU models. One such problem is the acquisition of a high-quality dataset, frequently asked by developers in Stack Overflow [45, 49]. A high-quality dataset that represents well the intents and entities supported by the chatbot is paramount for the chatbot performance. New comprehensive datasets and approaches that focus on generating labelled data can help alleviate this challenge faced by developers. Another problem is related to methods for extracting intents and entities, which has received some attention by the research community [25, 71, 73, 74, 76], but remains a challenging problem in chatbot development.

Implication for Educators. Educators can use our topics and categories as a roadmap to design their chatbot-related courses. The category development also has a high number of discussions looking for the most appropriate framework and best practices ('what' posts), hence, educators can introduce their audience to the several existing chatbot development frameworks and discuss best practices and standards to be followed during the chatbot development phase. As mentioned before, special attention should be given to the NLU topics, which has shown to be difficult (Figure 4). In particular, since NLU has the highest share of 'why' posts, this indicates that chatbot developers are in need of theoretical explanations of NLU machine-learning algorithms and models.

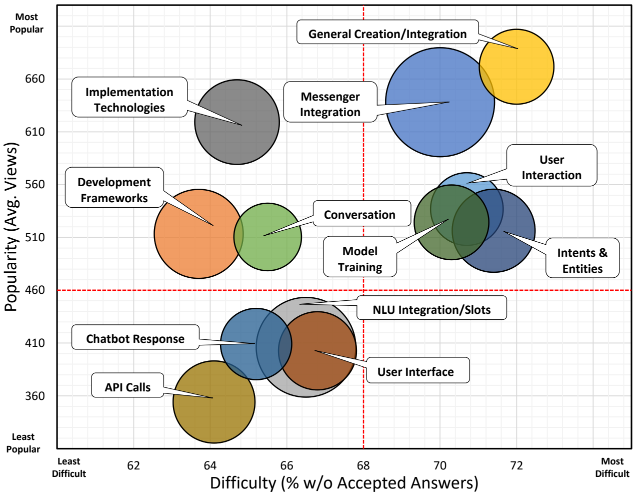

Figure 4: Chatbot topics' popularity vs. difficulty

There are many aspects that practitioners, researchers, and educators can take into consideration when deciding where to focus their efforts. Nevertheless, we believe that our findings and implications can help improve this decision-making process.

5 RELATED WORK

In this section, we present the studies related to the chatbots in SE domain and discuss the work that leverages and analyze Stack Overflow data to have more insights from developers perspectives.

Software Chatbots. A number of studies have focused on implementing chatbots to help developers in their daily tasks [3, 16, 61, 64, 69, 71]. For example, Bradley et al. [16] developed Devy to assist developers in their basic development tasks (e.g., commit a code). Abdellatif et al. [3] developed MSRBot that leverages repositories (i.e., Git and Jira) data to answer questions related to the software projects through natural language. Moreover, chatbots are used to assist customer service [70], answer student admission questions [5], and in the health care domain [17].

The rising of chatbots in academia and industry motivates us to examine the issues and challenges that facing chatbot developers in their implementations. We believe that our work provides an insights to the research community on the areas that require more investigation to allow developers focus on the core functionalities of the chabot and low the barrier to entry for the new practitioners to the chatbot domain.

Using Stack Overflow Data. There is a number of studies that use Stack Overflow data to study it's users commenting activities [75], the impact of code reuse from Stack Overflow on the mobile apps [1], and generates code comments for a code snippet [4]. The work closest to ours, is the work that applied LDA on Stack Overflow. Rosen and Shihab [54] summerized the mobile related questions on Stack Overflow, and the specific issues of the different mobile platforms. Similarly, Bagherzadeh and Khatchadourian [11] used topic modelling to extract the big data topics and big data developers interests from Stack Overflow. Wan et al. [67] use Stack Overflow to understand the challenges and needs amongst blockchain developers. Venkatesh et al. [66] examine the challenges that face client developers when using Web APIs using the Stack Overflow dump. Yang et al. [72] conduct a large scale study on Stack Overflow to identify the security-related questions asked by practitioners. Jin et al. [31] used Stack Overflow to investigate the issues that face developers when implementing or using Biometric APIs. Han et al. [30] conducted a large-scale study on Stack Overflow and Github using LDA to point out the topics discussed among developers about three deep learning frameworks (Tensorflow, PyTorch and Theano). Ahmed and Bagherzadeh [6] used LDA on Stack Overflow to identify the challenges and interests of concurrency developers.

To the best of our knowledge, there is no work that studied chatbot-related posts using Stack Overflow. We believe that our study complements prior work in Stack Overflow by analyzing chatbot-related posts. We extracted the chatbot topics and categorize them. Also, we examined the popularity, difficulty, and the growth of those topics compared to other studies. We believe that our work sheds the light for the research community on the areas that chatbot developers are interesting and challenging to the developers at an early stage of evolution of chatbots.

6 THREATS TO VALIDITY

Internal Validity: Internal validity concerns factors that could have influenced our results. We use tags from Stack Overflow to identify chatbot-related posts and it might be the case that some chatbot-related posts are mislabelled (i.e., missing tags or having incorrect tags) and therefore are omitted from our dataset. We mitigate this threat by examining all tags that coexist with the 'chatbot' tag and selecting a set of tags that are related to chatbots using the TST and TRT measures. Those measures have been used in prior work to have a better coverage of a certain topic's posts and limit the noise in the dataset [11, 54, 67, 72]. Moreover, we find that the TST and TRT thresholds that we obtain in our study are in-line with previous studies [6, 11, 72].

One potential threat is that we select $K = 12$ as the optimal number of topics for the LDA topic modelling technique. The number of topics (K) has a direct influence on the quality of the resulting topics from the LDA, and selecting an optimal number is known to be difficult. To alleviate this threat, we follow the approach used in similar studies to select the number of topics [30, 67]. Specifically, we experiment with different values of K and we examine the coherence of topics to select the optimal K value that balances the generalizability and relevance of the chatbot topics.

The labelling of posts types is another threat to the validity of our results, due to the subjectivity of the process. We mitigate this threat by performing three independent classifications and evaluating the interrater-agreement using the Cohen-Kappa test, that indicated substantial agreement among the annotators .

Construct Validity: Construct validity considers the relationship between theory and observation, in case the measured variables do not measure the actual factors. Labelling the resulting topics from the LDA might not reflect the posts associated with the topics. To minimize this threat, the first three authors individually examine the keywords and more than 30 posts randomly from each topic, then they discuss each topic's label to reach a consensus on the label that reflects the posts of that topic. We use different metrics to measure the popularity and difficulty of the chatbot topics which might be a threat to construct validity. These metrics have been used in similar studies [6, 11, 12, 41, 54, 72].

External Validity: Threats to external validity concern the generalization of our findings. Our study was focused on and collected data from posts on Stack Overflow, however, there are other forums that may host developers' discussions regarding chatbots. We believe that using Stack Overflow allows for the generalizability of our results as Stack Overflow is a very popular platform that hosts a large number of questions and answers from developers with a wide variety of domains and expertise. We also believe that this study can be improved by including discussions from different forums or surveying actual software developers about issues that they face when building chatbots.

The focus of this study is on chatbot which is considered to be a sub-category of software bots [33, 34]. Therefore, our observations and results cannot be generalized to other types of bots, such as agents. However, we believe that our observations are still relevant and contribute to the larger community (software bot). We encourage other researchers to conduct similar studies on other types of bots and compare the results from the different types to paint a full picture about bots in general.

7 CONCLUSION

In this paper, we analyze Stack Overflow posts to identify the most pressing issues facing chatbot development. We find that developers discuss 12 chatbot-related topics that fall under five main categories, namely Integration, Development, NLU, User Interaction, and User Input. Chatbot developers are highly interested in posts that are related to chatbot creation and integration into websites. On the other hand, training the NLU model of the chatbot proves to be challenging task for developers. We also find that chatbot practitioners show considerable interest in understanding the behavior of NLUs, while also seeking good recommendation regarding chatbot development platforms and best practices. We believe that our results are useful to the chatbot community as they guide future research to focus on the more pressing and difficult aspects of chatbot development. Moreover, our findings help platform owners to understand the issues faced by chatbot developers when using their platforms, and to overcome those challenges. Chatbot educators can take into consideration the discussed topics and categories and their perspective difficulty to better design their courses.

Our study opens the door for chatbot researches and practitioners to further understand the chatbot development challenges. Nevertheless, we plan in the future to examine developers' discussion from other forums to draw more accurate and generalizable conclusions. We also plan to investigate the developers discussions regarding bots in general, which would allow us to compare our results with with other bot types. Finally, we intend to investigate chatbot repositories and analyze the commits and bug reports to obtain further insights regarding the various issues faced by chatbot developers and their attempts to solve it.

REFERENCES

[1] Rabe Abdalkareem, Emad Shihab, and Juergen Rilling. 2017. On Code Reuse from StackOverflow. *Information Software Technology* 88, C (Aug. 2017), 148–158. https://doi.org/10.1016/j.infsof.2017.04.005

[2] R. Abdalkareem, E. Shihab, and J. Rilling. 2017. What Do Developers Use the Crowd For? A Study Using Stack Overflow. *IEEE Software* 34, 2 (Mar 2017), 53–60. https://doi.org/10.1109/MS.2017.31

[3] Ahmad Abdellatif, Khaled Badran, and Emad Shihab. 2020. MSRBot: Using Bots to Answer Questions from Software Repositories. *Empirical Software Engineering (EMSE)* (2020). https://doi.org/10.1007/s10664-019-09788-5

[4] E. Aghajani, G. Bavota, M. Linares-Vásquez, and M. Lanza. 2018. Automated Documentation of Android Apps. *IEEE Transactions on Software Engineering* (2018), 1–1. https://doi.org/10.1109/TSE.2018.2890652

[5] H. Agus Santoso, N. Anisa Sri Winarsih, E. Mulyanto, G. Wilujeng saraswati, S. Enggar Sukmana, S. Rustad, M. Syaifur Rohman, A. Nugraha, and F. Firdausillah. 2018. Dinus Intelligent Assistance (DINA) Chatbot for University Admission Services. In *2018 International Seminar on Application for Technology of Information and Communication*. IEEE Press, 417–423.

[6] Syed Ahmed and Mehdi Bagherzadeh. 2018. What Do Concurrency Developers Ask About? A Large-scale Study Using Stack Overflow. In *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM ’18)*. ACM, New York, NY, USA, Article 30, 10 pages. https://doi.org/10.1145/3239235.3239524

[7] Amazon. 2019. Amazon Lex - Build Conversation Bots. https://aws.amazon.com/lex/. (Dec 2019). (Accessed on 12/12/2019).

[8] Apple. 2020. Siri - Apple. https://www.apple.com/ca/siri/. (2020). (Accessed on 01/08/2020).

[9] Amazon AWS. 2019. Document History for Amazon Lex - Amazon Lex. https://docs.aws.amazon.com/lex/latest/dg/doc-history.html. (2019). (Accessed on 12/12/2019).

[10] C. J. Baby, F. A. Khan, and J. N. Swathi. 2017. Home automation using IoT and a chatbot using natural language processing. In *2017 Innovations in Power and Advanced Computing Technologies (i-PACT)*. IEEE Press, 1–6. https://doi.org/10.1109/IPACT.2017.8245185

[11] Mehdi Bagherzadeh and Raffi Khatchadourian. 2019. Going Big: A Large-Scale Study on What Big Data Developers Ask. In *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2019)*. Association for Computing Machinery, New York, NY, USA, 432–442. https://doi.org/10.1145/3338906.3338939

[12] Kartik Bajaj, Karthik Pattabiraman, and Ali Mesbah. 2014. Mining Questions Asked by Web Developers. In *Proceedings of the 11th Working Conference on Mining Software Repositories (MSR 2014)*. ACM, New York, NY, USA, 112–121. https://doi.org/10.1145/2597073.2597083

[13] Anton Barua, Stephen W. Thomas, and Ahmed E. Hassan. 2014. What are developers talking about? An analysis of topics and trends in Stack Overflow. *Empirical Software Engineering* 19, 3 (01 Jun 2014), 619–654.

[14] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *The Journal of Machine Learning Research* 3 (March 2003), 993–1022. http://dl.acm.org/citation.cfm?id=944919.944937

[15] Marion Boiteux. 2018. Messenger at F8 2018 - Messenger Developer Blog. https://blog.messengerdevelopers.com/messenger-at-f8-2018-44010dc9d2ea. (2018). (Accessed on 12/21/2019).

[16] Nick C. Bradley, Thomas Fritz, and Reid Holmes. 2018. Context-Aware Conversational Developer Assistants. In *Proceedings of the 40th International Conference on Software Engineering (ICSE ’18)*. Association for Computing Machinery, New York, NY, USA, 993–1003. https://doi.org/10.1145/3180155.3180238

[17] Gillian Cameron, David Cameron, Gavin Megaw, Raymond Bond, Maurice Mulvenna, Siobhan O’Neill, Cherie Armour, and Michael McTear. 2018. Best Practices for Designing Chatbots in Mental Healthcare: A Case Study on IHelpr. In *Proceedings of the 32nd International BCS Human Computer Interaction Conference (HCI ’18)*. BCS Learning & Development Ltd., Swindon, GBR, Article Article 129, 5 pages. https://doi.org/10.14236/ewic/HCI2018.129

[18] PACT Care. 2020. Florence - Your health assistant. https://www.florence.chat/. (2020). (Accessed on 01/08/2020).

[19] Chatbot application Life cycle 2019. Chatbot application Life cycle - Data Driven Investor - Medium. https://medium.com/datadriveninvestor/chatbot-application-life-cycle-8b2d083650a8. (June 2019). (Accessed on 12/16/2019).

[20] G. Chen, C. Chen, Z. Xing, and B. Xu. 2016. Learning a dual-language vector space for domain-specific cross-lingual question retrieval. In *2016 31st IEEE/ACM International Conference on Automated Software Engineering (ASE)*. 744–755.

[21] Google Dialogflow. 2020. Build an agent. https://cloud.google.com/dialogflow/docs/quick/build-agent. (2020). (Accessed on 01/16/2020).

[22] Saurabh "Dutta, Ger Joyce, and Jay" Brewer. "2018". "Utilizing Chatbots to Increase the Efficacy of Information Security Practitioners". In *"Advances in Human Factors in Cybersecurity"*, Denise" "Nicholson (Ed.). "Springer International Publishing", "237–243".

[23] Stack Exchange. 2019. Stack Exchange Data Dump. https://archive.org/details/stackexchange. (2019). (Sept. 2019).

[24] Facebook. 2019. Wit.ai. https://wit.ai/. (2019). (Accessed on 12/12/2019).

[25] J. Gao, J. Chen, S. Zhang, X. He, and S. Lin. 2019. Recognizing Biomedical Named Entities by Integrating Domain Contextual Relevance Measurement and Active Learning. In *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*. 1495–1499. https://doi.org/10.1109/ITNEC.2019.8728991

[26] Gensim. 2019. gensim: Topic modelling for humans. https://radimrehurek.com/gensim/. (2019). (Accessed on 12/03/2019).

[27] Google. 2020. Dialogflow. https://dialogflow.com/. (2020). (Accessed on 01/16/2020).

[28] Google. 2020. Google Assistant, your own personal Google. https://assistant.google.com/. (2020). (Accessed on 01/08/2020).

[29] Google. 2020. Integrations-Dialogflow Documentation. https://cloud.google.com/dialogflow/docs/integrations/. (2020). (Accessed on 01/16/2020).

[30] Junxiao Han, Emad Shihab, Zhiyuan Wan, Shuiguang Den, and Xin Xia. 2019. What do Programmers Discuss about Deep Learning Frameworks. *Empirical Software Engineering (EMSE)* (2019), To Appear.

[31] Z. Jin, K. Y. Chee, and X. Xia. 2019. What Do Developers Discuss about Biometric APIs?. In *2019 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. 348–352. https://doi.org/10.1109/ICSME.2019.00053

[32] C. Lebeuf, M. Storey, and A. Zagalsky. 2018. Software Bots. *IEEE Software* 35, 1 (January 2018), 18–23. https://doi.org/10.1109/MS.2017.4541027

[33] C. Lebeuf, A. Zagalsky, M. Foucault, and M. Storey. 2019. Defining and Classifying Software Bots: A Faceted Taxonomy. In *2019 IEEE/ACM 1st International Workshop on Bots in Software Engineering (BotSE)*. 1–6. https://doi.org/10.1109/BotSE.2019.00008

[34] Carlene R Lebeuf. 2018. *A taxonomy of software bots: towards a deeper understanding of software bot characteristics*. Ph.D. Dissertation.

[35] Andrew Kachites McCallum. 2002. MALLET: A Machine Learning for Language Toolkit. http://mallet.cs.umass.edu/. (2002). (Accessed on 12/03/2019).

[36] Mary McHugh. 2012. Interrater reliability: The kappa statistic. *Biochemia medica* 22 (10 2012), 276–82. https://doi.org/10.11613/BM.2012.031

[37] Microsoft. 2019. LUIS (Language Understanding) - Cognitive Services - Microsoft Azure. https://www.luis.ai/home. (2019). (Accessed on 12/12/2019).

[38] Microsoft. 2020. Microsoft Bot Framework. https://dev.botframework.com/. (2020). (Accessed on 01/16/2020).

[39] Microsoft. 2020. Quickstart: Create a new app in the LUIS portal - Azure Cognitive Services | Microsoft Docs. https://docs.microsoft.com/en-us/azure/cognitive-services/luis/get-started-portal-build-app. (2020). (Accessed on 01/16/2020).

[40] Milja Milenkovic. 2019. The Future Is Now - 37 Fascinating Chatbot Statistics. https://www.smallbizgenius.net/by-the-numbers/chatbot-statistics/. (Oct. 2019). (Accessed on 12/18/2019).

[41] Sarah Nadi, Stefan Krüger, Mira Mezini, and Eric Bodden. 2016. Jumping Through Hoops: Why Do Java Developers Struggle with Cryptography APIs?. In *Proceedings of the 38th International Conference on Software Engineering (ICSE ’16)*. ACM, New York, NY, USA, 935–946. https://doi.org/10.1145/2884781.2884790

[42] Natural Language Toolkit (NLTK). 2019. Natural Language Toolkit - NLTK 3.4.5 documentation. https://www.nltk.org/. (2019). (Accessed on 12/12/2019).

[43] Natural Language Toolkit (NLTK). 2019. NLTK’s list of english stopwords. https://gist.github.com/sebleier/554280. (2019). (Accessed on 12/23/2019).

[44] Stack Overflow. 2017. node.js - How to start a conversation from nodejs client to microsoft bot - Stack Overflow. https://stackoverflow.com/questions/46183295/how-to-start-a-conversation-from-nodejs-client-to-microsoft-bot. (2017). (Accessed on 12/20/2019).

[45] Stack Overflow. 2017. python - Dataset to train MITIE ner model - Stack Overflow. https://stackoverflow.com/questions/46602495/dataset-to-train-mitie-ner-model. (2017). (Accessed on 01/13/2020).

[46] Stack Overflow. 2017. Rasa nlu parse request giving wrong intent result - Stack Overflow. https://stackoverflow.com/questions/46466222/rasa-nlu-parse-request-giving-wrong-intent-result. (2017). (Accessed on 12/20/2019).

[47] Stack Overflow. 2018. Facebook Chat bot (PHP webhook) sending multiple replies - Stack Overflow. https://stackoverflow.com/questions/36609549/facebook-chat-bot-php-webhook-sending-multiple-replies. (2018). (Accessed on 12/20/2019).

[48] Stack Overflow. 2019. botframework - How to add custom choices displayed through Prompt options inside Cards & trigger actions on choice click in BOT V4 using c#? - Stack Overflow. (2019). https://stackoverflow.com/questions/56280689/how-to-add-custom-choices-displayed-through-prompt-options-inside-cards-trigge(Accessed on 01/16/2020).

[49] Stack Overflow. 2019. nlp - Is there a dataset that provides shopping conversations? - Stack Overflow. https://stackoverflow.com/questions/55324833/is-there-a-dataset-that-provides-shopping-conversations. (2019). (Accessed on 01/13/2020).

[50] Stack Overflow. 2019. node.js - How to store and retrieve the chat history of the dialogflow? - Stack Overflow. https://stackoverflow.com/questions/49665510/how-to-store-and-retrieve-the-chat-history-of-the-dialogflow. (2019). (Accessed on 12/21/2019).

[51] Stack Overflow. 2019. python - How to resume or restart paused conversation in RASA - Stack Overflow. https://stackoverflow.com/questions/57365685/how-to-resume-or-restart-paused-conversation-in-rasa. (2019). (Accessed on 12/20/2019).

[52] Stack Overflow. 2019. Stack Overflow Developer Survey 2019. https://insights.stackoverflow.com/survey/2019. (2019). (Accessed on 01/09/2020).

[53] Michael Röder, Andreas Both, and Alexander Hinneburg. 2015. Exploring the Space of Topic Coherence Measures. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining (WSDM '15)*. ACM, New York, NY, USA, 399–408. https://doi.org/10.1145/2684822.2685324

[54] Christoffer Rosen and Emad Shihab. 2016. What Are Mobile Developers Asking About? A Large Scale Study Using Stack Overflow. *Empirical Software Engineering* 21, 3 (June 2016), 1192–1223. https://doi.org/10.1007/s10664-015-9379-3

[55] B. Rychalska, H. Glabska, and A. Wroblewska. 2018. Multi-Intent Hierarchical Natural Language Understanding for Chatbots. In *2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS)*. 256–259. https://doi.org/10.1109/SNAMS.2018.8554770

[56] A. Abdellatif, D. Costa, K. Badran, R. Abdalkareem, E. Shihab. 2020. Dataset. https://zenodo.org/record/3610714. (2020). (Accessed on 01/16/2020).

[57] Spearman. 2008. *Spearman Rank Correlation Coefficient*. Springer New York, New York, NY, 502–505. https://doi.org/10.1007/978-0-387-32833-1_379

[58] Margaret-Anne Storey and Alexey Zagalsky. 2016. Disrupting Developer Productivity One Bot at a Time. In *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE 2016)*. ACM, New York, NY, USA, 928–931. https://doi.org/10.1145/2950290.2983989

[59] Sumo. 2020. 5 Ecommerce Chatbots (Plus How To Build Your Own In 15 Minutes). https://sumo.com/stories/ecommerce-chatbot-marketing. (2020). (Accessed on 01/08/2020).

[60] TechCrunch. 2017. Wit.ai is shutting down Bot Engine as Facebook rolls NLP into its updated Messenger Platform. (2017). https://techcrunch.com/2017/07/27/wit-ai-is-shutting-down-bot-engine-as-facebook-rolls-nlp-into-its-updated-messenger-platform (Accessed on 12/12/2019).

[61] Y. Tian, F. Thung, A. Sharma, and D. Lo. 2017. APIBot: Question answering bot for API documentation. In *2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE)*. 153–158. https://doi.org/10.1109/ASE.2017.8115628

[62] Carlos Toxtli, Andrés Monroy-Hernández, and Justin Cranshaw. 2018. Understanding Chatbot-mediated Task Management. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 58, 6 pages. https://doi.org/10.1145/3173574.3173632

[63] Christoph Treude, Ohad Barzilay, and Margaret-Anne Storey. 2011. How Do Programmers Ask and Answer Questions on the Web? (NIER Track). In *Proceedings of the 33rd International Conference on Software Engineering (ICSE '11)*. ACM, New York, NY, USA, 804–807. https://doi.org/10.1145/1985793.1985907

[64] Simon Urli, Zhongxing Yu, Lionel Seinturier, and Martin Monperrus. 2018. How to Design a Program Repair Bot? Insights from the Repairnator Project. In *Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP '18)*. ACM, New York, NY, USA, 95–104. https://doi.org/10.1145/3183519.3183540

[65] Stefano Valtolina, Barbara Rita Barricelli, and Serena Di Gaetano. 2020. Communicability of traditional interfaces VS chatbots in healthcare and smart home domains. *Behaviour & Information Technology* 39, 1 (2020), 108–132. https://doi.org/10.1080/0144929X.2019.1637025 arXiv:https://doi.org/10.1080/0144929X.2019.1637025

[66] P. K. Venkatesh, S. Wang, F. Zhang, Y. Zou, and A. E. Hassan. 2016. What Do Client Developers Concern When Using Web APIs? An Empirical Study on Developer Forums and Stack Overflow. In *2016 IEEE International Conference on Web Services (ICWS)*. 131–138. https://doi.org/10.1109/ICWS.2016.25

[67] Z. Wan, X. Xia, and A. E. Hassan. 2019. What is Discussed about Blockchain? A Case Study on the Use of Balanced LDA and the Reference Architecture of a Domain to Capture Online Discussions about Blockchain platforms across the Stack Exchange Communities. *IEEE Transactions on Software Engineering* (2019), 1–1. https://doi.org/10.1109/TSE.2019.2921343

[68] Joseph Weizenbaum. 1966. ELIZA-A Computer Program for the Study of Natural Language Communication between Man and Machine. *Commun. ACM* 9, 1 (Jan. 1966), 36–45. https://doi.org/10.1145/365153.365168

[69] Marvin Wyrich and Justus Bogner. 2019. Towards an Autonomous Bot for Automatic Source Code Refactoring. In *Proceedings of the 1st International Workshop on Bots in Software Engineering (BotSE '19)*. IEEE Press, Piscataway, NJ, USA, 24–28. https://doi.org/10.1109/BotSE.2019.00015

[70] Anbang Xu, Zhe Liu, Yufan Guo, Vibha Sinha, and Rama Akkiraju. 2017. A New Chatbot for Customer Service on Social Media. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. Association for Computing Machinery, New York, NY, USA, 3506–3510. https://doi.org/10.1145/3025453.3025496

[71] Bowen Xu, Zhenchang Xing, Xin Xia, and David Lo. 2017. AnswerBot: Automated Generation of Answer Summary to Developers' Technical Questions. In *Proceedings of the 32Nd IEEE/ACM International Conference on Automated Software Engineering (ASE 2017)*. IEEE Press, Piscataway, NJ, USA, 706–716. http://dl.acm.org/citation.cfm?id=3155562.3155650

[72] Xin-Li Yang, David Lo, Xin Xia, Zhi-Yuan Wan, and Jian-Ling" Sun. 2016. What Security Questions Do Developers Ask? A Large-Scale Study of Stack Overflow Posts. *Journal of Computer Science and Technology* 31, 5 (01 Sep 2016), 910–924.

[73] D. Ye, Z. Xing, C. Y. Foo, Z. Q. Ang, J. Li, and N. Kapre. 2016. Software-Specific Named Entity Recognition in Software Engineering Social Content. In *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, Vol. 1. 90–101. https://doi.org/10.1109/SANER.2016.10

[74] Shayan Zamanirad, Boualem Benatallah, Moshe Chai Barukh, Fabio Casati, and Carlos Rodriguez. 2017. Programming Bots by Synthesizing Natural Language Expressions into API Invocations. In *Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE 2017)*. IEEE Press, 832–837.

[75] H. Zhang, S. Wang, T. Chen, and A. E. Hassan. 2019. Reading Answers on Stack Overflow: Not Enough! *IEEE Transactions on Software Engineering* (2019), 1–1. https://doi.org/10.1109/TSE.2019.2954319

[76] Qi Zhang, Jinlan Fu, Xiaoyu Liu, and Xuanjing Huang. 2018. Adaptive Co-attention Network for Named Entity Recognition in Tweets. (2018). https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16432