Evaluation of the Existing Bots

Bhargavi Mohan

January 15, 2021

1 Introduction

The first phase towards assisting documentation process in a mechatronics environment is to assess the existing bots to check if there are any bots currently, that are capable of resolving the problem description as stated in the Thesis Proposal document[1]. My objective in this phase was to examine the functionality and features of the available bots in SLACK and MS TEAMS that exclusively handle topics like documentation, decision management and message tagging/labeling.

2 Existing bots and their features

In this section, I review 3 bots each in Slack and MS Teams to explain how they help the users in decision documentation process.

2.1 Bots in Slack

Slack has a good amount of bots that help in chat summarization. Summarized chats can be saved for future use hence promoting documentation process.

2.1.1 tilda—

A bot whose main feature is chat summarization. The main idea here is to group the incoming messages as questions, answers, ideas, information, action etc. The users are required to use the inbuilt commands to notify the bot what type of message are they sending in a group. For instance, the commands are **/∼addquestion**, **/∼addanswer** ,**/∼addinfo**, **/∼addanswer**. These commands are understood by the bot and it groups the user messages accordingly and gives a summary upon another command — **/∼currentsummary**. The summary contains the people involved the chat, the number of messages, date and the time.

The summaries are sent as messages by the bot in the same group and all the members in the group can be refer up to a certain time. These summaries may disappear later

---

[1] https://github.com/bhargavimohan/Masterthesis_bot2021/blob/main/ BhargaviMohan_Proposal_and_thesisapplication_signed.pdf

because slack offers to access messages only upto 10,000 recent messages and the older messages can be retrieved by upgrading the slack plan. Therefore, these summaries need to be screenshot explicitly to save them to the local system permanently.

2.1.2 summarize bot—

The feature of this bot is simplistic. It is only capable of summarizing any public we-blink with a group chat or individual chat. The bot can also summarize documents, images etc. The command used is **/summarize** [url]. Keyword extraction, key fragments list, desired summary size, downloadable results are its key features.

The reason to review this bot is because I wanted to understand the summarization technique used in this bot to actually see how it differs from the previous one.

2.1.3 decisionbot—

As the name suggests, this bot helps in quicker decision making in a group conversation. This is both decision management and decision documentation. A user can use the decision maker template by initiating the command— **/decision** and fill the details such as title, description, decision maker, due date and submit the form. A new channel with the title name gets created and the users are allowed to edit the details in the channel. Other members can also be invited to join the conversation. The "Make decision" option will also be available to the intended person. Once the right person submits the decision, the bot posts the same on the channel and the channel is then archived. However, these decisions are browsable even later in that particular channel. Additionally, there is another tab in the bot channel that shows a history of all decisions, a graphical representation of open and closed decisions over time.

2.2 Bots in MS Teams

MS teams does not have any summarizing bots, hence no bots that could help in documenting chats. Decision management bots are available and are discussed below.

2.2.1 we decide—

This bot is very similar to the bot discussed in 2.1.3. This bot provides a decision making form to a user who can fill the title, description, assigned date, due date and assign it to one of the members in the group. The responsible person either approves or rejects the decision. Additionally, there are text-boxes to add action items in case of approval and rejection. Files can also be attached for any required reference. This bot handles decision management and documentation.

2.2.2 perfony—

This is also a bot that helps decision making in a team. It has a complex UI and features. The decision can be created with the assignee details. This decision form can be created in the form of a folder and can create further sub-folders for related decision items.

There are 3 forms of views namely — folder view, kanbann view, Gnatt view. There are comment sections, attachment sections, bookmark option, download option, archive option, and also a status bar that can be dragged to indicate the state of the current decision topic. There are many more features that are not in interest for our requirement hence I do not list them here.

2.2.3 approve simple—

I considered this bot for the review because it claimed that it accelerated decision making in a corporate environment. Whilst using the bot, it was more useful for employees in the managerial roles and not much useful for decision making or documenting in group conversations. It is used as an approval tool for managers or directors in a 1:1 conversation with the bot. The user should type a command like My Approvals and the bot displays a queue of all the approvals in different systems like leaves, timesheet, budget etc. The manager can approve or reject stating the reason(optional).

2.3 Ideal bot features for thesis implementation

This thesis implementation involves prototyping a bot that should be **available in MS Teams** and should support **design decision management**. **Decision documentation** of any important decisions that are discussed in a group chat should also be taken care. There could be data that needs to be **tagged** as "Knowledge Transfer" material for later reference to junior employees which means **Knowledge management** should be possible. Furthermore, the bot should be able to offer a list of questions, decisions or actions and for this **message segregation** becomes necessary. For all these functionalities, **summarizing the group chat** on a timely basis seems to be an optimal remedy to execute the rest of the features highlighted above.

Ultimately, the users should be given an option to either save or discard the summary depending upon the significance of a group conversation.

2.4 Summary

In this section, all the above bots are reviewed against the preferred features that meet the requirements of the thesis topic.

Evaluation criteria	tilda	summarize bot	decisionbot	we decide	perfony	approve simple
Group chat summarization	yes	no	no	no	no	no
Message labeling/Tagging	no	no	no	no	no	no
Message type segregation	yes	no	no	no	no	no
Knowledge management	no	no	no	no	no	no
Decision management	yes	no	no	yes	yes	yes
Design decision documentation	partly yes	no	yes	yes	yes	no
Availability in MS Teams	no	no	no	yes	yes	yes

3 Summarization

The best possible solution to reinforce documentation of vital data in a group conversation seems to be summarizing daily conversations and saving the most prominent summaries. Summarizing conversations have multiple advantages namely—

- The long chat conversations are made precise and pertinent.

- The summaries would in turn have messages clustered into questions, answers, decisions and more and this provides a much easier look-up of the content.

- The saved summaries can serve as Knowledge Transfer material to juniors or new employees.

- Most of all, it is a good way to document all the crucial discussions in a group chat for future references.

The details of performing chat summarization will be discussed in the next phase of the thesis. However, a literature review of the popular summarization techniques is presented in the following section.

4 Literature survey on text summarization and segregation techniques

[1] Text Summarization Techniques: A Brief Survey — This paper reviews all major methods of automatic text summarizations. Various summarization processes, their pros and cons are surveyed. This paper defines "automatic summarization" and quotes that it is very challenging. The authors talk about different methodologies to calculate the weight of a sentence. Generally, there are two approaches to summarization namely extractive and abstractive summarization. The focus in this paper is more on extractive summarization due to the absence of complete abstractive summarization system. Three independent tasks in extractive summarization have been explained. The emphasis of context in summarization is discussed. Furthermore, graph-based and machine learning techniques are discussed to identify important sentence inclusion in a summary. Finally, Evaluation as in human evaluation and automatic evaluation methods are addressed.

[2] A Survey of Extractive and Abstractive Automatic Text Summarization Techniques — In this paper, bio-inspired methods for text summarization is the core of the paper. They are best for optimizing capabilities. Swarm based feature selection method is quoted to be 43% similar to manual summaries. Another method called Genetic Probabilistic based Summarization is used to extract the weights of features in a text, this means the selection of important sentences can be extracted to form a summary. Abstractive summarization has four main steps like text segmentation, construction of lexical chains, recognizing strong chains and finally derive substantial sentences. Many other scientific work like novel graph based summarization, SUMMARIST(equation

based summarization) has been quoted along with their key ideas. The authors conclude by saying that the abstractive summarization surpass extractive methods but more expensive to compute.

[3] Text Segregation On Asynchronous Group Chat — This paper presents the methodology for text segregation and the workaround for it is a pipeline and some heuristics with certain parameters. The authors say, once the long asynchronous message thread is well segregated, summarizing that is easier. Three main aspects to handle successful segregation is explained and they are message thread, time window an topic and the use of heuristics will manage them. A flowchart is presented to understand the thread disambiguation logic. The dataset, sample texts, the error rate and the results are all depicted. The segregated threads are presented in the form of a table.

[4] A Survey Automatic Text Summarization — This paper gives a comparative study of different text summarization techniques. The three phases to extractive summarization according to authors are pre-processing which involves techniques like Part of Speech tagging, stop word filtering, stemming, Feature calculation etc. The second phase is processing which involves frequency of words that helps giving a certain score to a sentence and the sentence with highest score is included in the summary and this is called query based summarization, an algorithm is also given to compute this. Several other techniques such as Hidden Markov Model, Neural Networks Based Text Summarization, Fuzzy Logic Based Text Summarization, Bayesian Classifier are all compared and examined. Additionally, two criterion — Precision and Recall evaluation terms are defined to rate the efficiency of a technique.

5 Conclusion

Based on the evaluation of the existing bots, there is no bot currently available in MS teams to handle the thesis requirement. There are decision management bots but these bots require the user to explicitly pick a decision template every time a decision has to be made. There is need for management and documentation within the asynchronous group chat and the quest is for such a capability within a bot. Among the examined bots, **"tilda"** that is available on slack seems to be the closest that has potential to tackle the prerequisites of this thesis topic. However, this bot has to be adapted to MS Teams with a few more modifications. Possible modification could be to automatically summarize and document without having to use explicit commands during a conversation. The usage of these commands currently is too much action for the users. It can make the users exhausted and not want to use the bot. On the other hand, automatic summarization is quite challenging based on my literature survey in section 4, it involves convoluted pre-processing and development. Hence, a more probable modification can be to make the usage of commands seamless and effortless .That way, the users are motivated to use the bot without spending too much time just on the commands.

My next set of tasks include figuring out if the automatic summarization is within the scope of this thesis implementation, working out on designing the bot with a more fluent

and smooth usage of commands. The results of this will be reported in my following document.

References

[1] Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D. Trippe, Juan B. Gutierrez, and Krys Kochut. Text Summarization Techniques: A Brief Survey. *arXiv:1707.02268 [cs]*, July 2017. arXiv: 1707.02268.

[2] V. Dalal and L. Malik. A Survey of Extractive and Abstractive Text Summarization Techniques. In *2013 6th International Conference on Emerging Trends in Engineering and Technology*, pages 109–110, December 2013. ISSN: 2157-0485.

[3] Abhishek Sinha, Midhush Manohar T.K., Srikumar Subramanian, and Bhaskarjyoti Das. Text Segregation On Asynchronous Group Chat. *Procedia Computer Science*, 171:1371–1380, 2020.

[4] Oguzhan Tas and Farzad Kiyani. A SURVEY AUTOMATIC TEXT SUMMARIZATION. *Innovation and Entrepreneurship*, page 10.