SENG 435 Project Interim Report

Alison Goshulak V00806939 | Jake Rothwell V00813277 | Jonathan Grandfield V00823227

Introduction and Motivation

The goal of our project is to develop a Slack bot, named S.P.A.R.K.Y., that encourages collaboration between members of a group through gamification and pathos. To elaborate on this “appeal to emotion” method, the main objective of S.P.A.R.K.Y. is to encourage the members of the group to work together with a common goal of nurturing this seemingly living entity. Our target users are students working on a group project (with a team size of about five people) and using a Slack channel for communication and document sharing towards completing the project. Our main goals for building S.P.A.R.K.Y. are to motivate quieter members of a school project to be more engaged and to give the group an idea of the “health” of their channel, based on how often as well as what forms they communicate with each other.

Up until this point, we have reviewed some relevant research papers which have given us some insights into potential pitfalls with the gamification features of our bot. We have also looked at a few products with similar goals to ours that have given us ideas of possible solutions to information update scheduling and message intention processing. We have created a very basic version of the bot with a few key features, such as alerts to the channel when there has been no messages posted in *x* hours and state changes based on channel activity. Since the time of our project proposal, we have found a few more potential risks and ethical issues with the implementation and evaluation of S.P.A.R.K.Y. The remainder of the report discusses all of these subjects in detail.

Related Work

We have examined a couple research papers on gamification of collaborative work. The first paper, “Gamify Employee Collaboration - A Critical Review of Gamification Elements in Social Software,” provides a critical analysis of popular products that appeared in the authors’ market review. The authors argue that the analyzed products favor quantitative work over qualitative work, which ultimately discourages users by shifting their focus to intangible rewards and achievements instead of the intrinsic value of the work itself [1]. This bias also disrupts the flow (the balance between challenge and skill) of the users’ work by influencing them to pick smaller, less challenging tasks to get more work done faster [1]. This is a risk for S.P.A.R.K.Y. since our current plan for the bot involves rewarding the group with achievements based on the number of messages, reactions, links, etc. in the channel. This paper has therefore made us re-strategize to also (tentatively) add a reputation mechanic based on feedback on given answers to questions (this idea is discussed further in the Implementation section).

The second paper, “Using Gamification as a Collaboration Motivator for Software Development Teams: A Preliminary Framework” [2], did not give us many ideas on how to evaluate our bot,

but it did introduce us to the Behavior Platform by Badgeville, a gamification platform that appeared several times in the market review of the first paper [1]. The platform includes descriptions of game mechanics that the company considers effective, such as missions and reputation mechanics [3]. The Behavior Platform could help us build a common language for the gamification aspect of our bot and avoid potential pitfalls associated with the different mechanics: for example, the platform argues that desired behavior from the users must be reinforced [3], so we may want to think more about how we can reward good behavior quickly (i.e. leveling-up may be too delayed a response on its own).

We also looked at a few related Slack bots to compare to our own. The first two bots, Standuply and Standup Alice, are used to automate asynchronous stand-up meetings [4,5] and are similar to our own project since they facilitate communication between group members. What we were most interested in when reviewing the bots is how they schedule the meetings, specifically how much flexibility they give the users when setting up a stand-up team. Both bots allow the user to pick the time for the meetings, but are strict on the meetings occuring daily [4,5]; we may follow this idea in own project by allowing the user setting up S.P.A.R.K.Y. to pick what time the bot gives the channel a status update, but not how often.

The third bot we reviewed,Tettra, is used for building wikis to organize collective team knowledge [6]. This bot gave us an idea of how to solve one of our problems: we want our bot to link a question posted by a group member to a given answer, but we're not sure how to best manage this. If we create an object instance representing a posted question (like how Tettra creates a Page Request for a user asking for info from another team member [6]), we can link that question with a corresponding answer. An issue with this solution is that it might require users to include commands when asking or answering questions which may not feel natural. Alternatively, we could use Wit.ai, a Natural Language Processing (NLP) application [7], to detect questions and answers without requiring the users to explicitly identify questions and answers.

We have had some trouble finding relevant research papers that were not specifically about gamification, i.e. about encouraging communication in collaborative work more generally. In the future, we will look for more applications with similar goals to S.P.A.R.K.Y., such as improving team culture or providing channel or team analytics, which may be easier to find. We will also look for more resources on gamification to further inform our approach to this aspect of the bot.

Approach

Our approach to the project has generally consisted of meeting up for brainstorming sessions with a goal in mind of honing in on the features we want S.P.A.R.K.Y. to include. It has been an ongoing process to solidify the specific goal that our collaborative bot aims to achieve. The major takeaway we got from the feedback on our project proposal was that our problem statement was not clear enough. With such an open-ended project choice, there is a great danger of "feature creep." As mentioned in our problem statement, there is now a specific demographic that this bot is aimed towards. If we end up adding too many features to

S.P.A.R.K.Y., it will lose its focus on the overall scope/goal due to the overload of features. Our brainstorming sessions have therefore been focused on filtering our long list of potential features into a set that benefits our target demographic - small, project based teams. The plan is to have one instance of the bot per channel which tracks the activity of each team member and changes state or level based on the collective activity of the channel.

Roles

In terms of individual roles for the project, we have decided to take the approach of each of us contributing fairly equally to all aspects of the project. However, as the project has moved forward we have found ourselves gravitating towards specific roles, as follows:

Jonathan Grandfield: Artwork, Bot State Transformations, Creative Direction
Alison Goshulak: Literature Review, Requirements and Use Cases
Jake Rothwell: General Development and Implementation, Evaluation Strategies

We all contribute equally to completing the reports and presentations as well as implementing features for S.P.A.R.K.Y. Although these roles suggest that we each work on exclusive aspects of the project, we are meeting on a regular basis to collaborate and collectively implement the proposed features.

Implementation

Currently in the project we have implemented the groundwork for the more advanced features planned for the second build of the bot. As mentioned in our proposal, we will be using Python 3 and the Slack API for developing S.P.A.R.K.Y., as well as Wit.ai for NLP. In our first build, the bot is able to hold a persistent state - either “content” or “sleepy.” “Content” is the default state, where users in the channel are regularly posting messages, which keeps S.P.A.R.K.Y. happy. However, if no messages have been sent in the channel in x minutes, S.P.A.R.K.Y. will transform into a “sleepy” state. Currently, “content” and “sleepy” are the only two states we have implemented, but we plan on adding the following bot states to the second build, and their corresponding triggers:

- **Emotive** - many emojis/reactions detected in channel
- **Hard-working -** many Google Drive links or code snippets detected in channel
- **Silly** - many memes/gifs detected in channel
- **Inquisitive** - many questions detected in channel

An important characteristic of S.P.A.R.K.Y. that we want to add is that its speech and behaviour changes based on its state**.** At the moment, the only change that occurs when S.P.A.R.K.Y.’s state changes is a .gif image and a short statement, such as “I am content,” corresponding to the current state. While this feature gives S.P.A.R.K.Y. a little personality, the bot is so far quite devoid of human-like behaviour. Since we want S.P.A.R.K.Y. to feel life-like, we would like to make its behavior more dynamic by changing its speech and emoji reactions when it’s state

changes. For instance, while in the “content” state S.P.A.R.K.Y. could periodically react with smiles, or in the emotive state have a more wide range of reactions with a sillier tone to its interaction with the users. Ultimately, we want to uphold one of the main pillars of S.P.A.R.K.Y. that we decided early on, which is an appeal to emotion of the users.

One problem we ran into is that by uploading a gif every time a user asks for a status update or an automatic update occurs, we are flooding the channel with repetitive files, taking up space and making it harder for users to find the files they are posting. It also takes on average over 10 seconds for the gifs to fully upload which we feel is too long of a response time for the type of agent we want S.P.A.R.K.Y. to be. This can be mitigated by including a feature that gives S.P.A.R.K.Y. a “user typing” message in order for other users to at least be aware that the command is in progress. Ideally we can get the gif response time closer to the normal message response time. This could be accomplished by using animated emojis (which may prove too small to allow for the same emotional connection provided by being able to see finer details in the bot’s state changing), by “sharing” the .gif files if they have already been uploaded in the channel, or by some alternative means through the Slack API we are not yet aware of.

Continuing on with the proposed features, we want S.P.A.R.K.Y. to be able to detect the amount of questions being asked in the channel, as well as whether or not they have been answered. This will involve first detecting a question, logging it, and waiting for a response to that question. If an answer is detected for that given question, S.P.A.R.K.Y. will keep track of this “questions answered” metric which will become useful for changing its state. The crucial problem we have to solve is how to link the questions to their corresponding answers. A simple solution would be to have S.P.A.R.K.Y. detect a mention of the questioner following the message containing the question, but this might not be a reliable method. Another potential solution would involve using Wit.ai, an NLP library which can be manually trained depending on context [7]. In our context, we can feed examples of questions, as well as examples of answers, and if S.P.A.R.K.Y. detects an example of an answer directly after a question, it will consider that question answered.

S.P.A.R.K.Y.’s detection of questions answered leads to a more advanced feature that we want to implement - user activity tracking and reputation. Recorded individual users’ activity would include messages sent, questions asked and answered, as well as emoji reactions to answered questions. We hope that this information is both quantitative and qualitative, and with this information S.P.A.R.K.Y. will form dispositions towards each user with the reputation mechanic. Furthermore, S.P.A.R.K.Y. will have “favourite” users, which we hope will create a dimension of competition to the collaborative effort to nurture the bot’s well-being amongst the members of the channel.

Finally, if time permits, we would like to implement some quality of life features for teams that use S.P.A.R.K.Y., including a fully customizable set of options that would control how active S.P.A.R.K.Y. is within the channel. For instance, the team could fine tune how often S.P.A.R.K.Y. pesters the group or goes into “sleepy” state, based on the level of synchronicity and distance of that given team. We also want the users to be able to easily access the user

and channel information that is gathered by S.P.A.R.K.Y. in the form of information visualization on a RSS feed.

Milestones Timeline

For the remainder of the term our tentative timeline is as follows. Currently we have a minimal "first build" of S.P.A.R.K.Y., so the remainder of the term will consist of steps to iterate upon this build and add the proposed set of features. The approach we want to take for the remainder of the term is to rollout features in stages, and evaluate S.P.A.R.K.Y. at each stage. Instead of focusing on achieving a "final build" and evaluating afterward in a sequential manner, we plan on taking a more agile approach to the remainder of the project. As individual features are completed, we will test and evaluate them as they roll out.

July 6th: Build 0.2 - Question/Answer detection complete
July 13th: Build 0.3 - User activity and reputation, evaluation session 1
July 20th: Build 0.4 - Dynamic S.P.A.R.K.Y. behaviour, based on state evaluation session 2
July 27th: Finalizing version 1.0 of S.P.A.R.K.Y., parse gathered evaluation data, make necessary modifications, add customizability for users of the channel

Proposed Evaluation Steps

Due to our decision to focus on smaller groups of students, an obvious decision for us moving forward is to try to use our fellow classmates' groups for user testing. Once we have a stable enough state that won't risk being overly obnoxious, we plan on requesting that users allow S.P.A.R.K.Y. to join into a channel that they are currently using to work on their projects. We may also request that users either allow us access to the channel during its use (purely as silent observers), but failing that would require a snapshot of S.P.A.R.K.Y.'s interactions and use within their channel (whether this be submitted by the group or collected by S.P.A.R.K.Y. . It may also be worth testing S.P.A.R.K.Y. in a more casual channel to see how it reacts when user input become less formal and more unpredictable.

We plan on conducting a user survey after the period of use (and potentially one or two smaller ones to "check-in" at different phases of the adoption process) in order to gain some insight into users' subjective opinions of the bot. These surveys will be conducted individually so as to avoid any groupthink or stifling of differing opinions. We would prefer the survey results not be anonymous so we may compare a user's opinion with their interactions from the snapshot.

Risks

All of our risks that we proposed in our initial report still apply, namely the issue of S.P.A.R.K.Y. proving to be too obnoxious or distracting to serve effectively as a tool to encourage meaningful collaboration. Part of this risk is mitigated by the fact that we have currently narrowed the scope of our target audience to be groups of students, who are far more likely to appreciate and have interest in a gamified or casual environment than a group in a (possibly) extremely professional

corporate setting. However, one of our related works mentioned above, Badgeville, uses gamification and has seen fairly successful adoption in corporate environments (they claim to have over 250 customers around the globe, including some giants like Samsung and NBC) so it should not be written off entirely. As long as we don't over emphasize the gamification aspects and create S.P.A.R.K.Y. more as a background agent than an active user, these risks should remain manageable.

There is also the risk of user abuse of the system (as there is whenever a user is involved), especially if we decide to use NLP. The best way for us to minimize the negative effects of this would be to simply keep possible examples of user abuse in mind during development, and to keep an eye out for it during our user testing period.

The risk of violation of privacy with regards to the bot should not be too much of an issue as S.P.A.R.K.Y. does not log content of messages, only categorizes them, but still should be kept in mind moving forward. Privacy becomes more of an issue when we enter our user testing phase as we would prefer to be directly observing user interactions with S.P.A.R.K.Y. over a longer period of time within another group's channel. We also recognize risks involving emotional manipulation which will be expanded further in the ethics section.

Ethics

For our user testing and surveys we plan to utilize the guides included in the "project resources" folder to ensure we are following proper procedure and minimizing any of the risks included with such tasks. A reasonable ethical concern we have is that of S.P.A.R.K.Y. potentially introducing conflict into a group via encouragement of competition, as well as the problem of emotional manipulation, particularly if we choose to include factors such as S.P.A.R.K.Y. calling some users out for not participating enough.

References

[1] C. Meske *et al*, "Gamify Employee Collaboration - A Critical Review of Gamification Elements in Social Software," 2016.

[2] F. Steffens *et al*, "Using Gamification as a Collaboration Motivator for Software Development Teams: A Preliminary Framework," 2015.

[3] Enterprise-gamification.com. (2018). *Badgeville - Enterprise Gamification Wiki*. [online] Available at: http://www.enterprise-gamification.com/mediawiki/index.php?title=Badgeville [Accessed 28 Jun. 2018].

[4] Slack App Directory. (2018). *StandupIy*. [online] Available at: https://cscwuvic2018.slack.com/apps/A355V71K7-standupiy [Accessed 22 Jun. 2018].

[5] Slack App Directory. (2018). *Standup Alice*. [online] Available at: https://cscwuvic2018.slack.com/apps/A454FNE64-standup-alice [Accessed 22 Jun. 2018].

[6] Slack App Directory. (2018). *Tettra*. [online] Available at: https://cscwuvic2018.slack.com/apps/A0H4VM9FG-tettra [Accessed 22 Jun. 2018].

[7] Wit.ai. (2018). *Wit — Docs Home*. [online] Available at: https://wit.ai/docs [Accessed 23 Jun. 2018].