Project Proposal

Open source based Learning App

Group 9

07/10/2019

1. Introduction

1.1 Background

Primary software developers often encounter many problems while browsing open source sites to learn programming and these open source sites often do not contain problem solving modules, because existing synchronous communication technologies (such as voice chat) have high scheduling costs, and asynchronous communication tools (such as BBS) require developers to carefully describe their code context in order to produce useful responses. Those problems which can be solved immediately could become an obstacle for beginners.

1.2 Vision Statement

Our goal is to develop a website extension implemented in an open source site that automatically returns an explanation based on the content of codes the developer asks. If the questioner is not satisfied with the answer, they can use the online question and answer module to get further explanation of the code from the engineer.

1.3 General goals

- To implement the online question and answer module. When beginners encounter obstacles during the learning process, they can simply open the online chatting window, then send the detailed description of the question to the responder. The respondent returns the explanation of the question immediately. The questioner can ask the question multiple times until the problem is solved perfectly.
- To implement the automatic answer reply module. Regrading to the circumstance of full-line, which means all responders are busy, we add auto reply as extended functionality of online question and answer module. When learners face a problem and no host is available at the moment, they can firstly trigger auto reply function, it will automatically give a considered answer according to the key words in the question, which is trained by machine learning algorithm, similar to the implementation of chatbot. Furthermore, if the learners are not satisfied with the answer, they can always hold the line and wait for available responders.

1.4 Optional Goals

In this part, there will be some optional functions or goals to upgrade our application if we can finish the basic deliverables early. It also can be seen as an expand of our project which can be developed and improved in the future.

There will be a bonus system for those people who is active in the application, for example, the users who can raise up useful questions or general questions will be rewarded with some points or special titles; and the users whose answer satisfies the questioner and widely accepted can be awarded as well.

Building a chatbot mechanism which can answer users question automatically, and we will train the chatbot to make it intelligent enough to give satisfying answer. The programming language will not be limited to python only.

Creating an extension for wording function. When the mouse or the detector stops on the top of a line of code, it can detect the code and generate explanation automatically.

1.5 Group/Team Goals

Front-end team:(Wanze Liu, Xiyao Li)

Designing the homepage and set the interfaces layer for input point of the project.

Designing the interfaces layer for the data respond from backend and display in the front page.

Deciding the style of the whole project and improve the interaction experience for the users.

Back-end team:(Jiachen Li, Peiguo Guan)

Database design and control the data flow from input and output.

Designing the algorithm that we use and manage the data.

Creating APIs and implement the interface.

Achieving the function as requirement.

Testing team:(Yang yang)

Testing the function and performance of the application.

Checking the format of code and review the details.

Continuously testing the code and giving feedback.

1.6 Individual Team Member Goals

Xiyao Li: Front-end designer, develop the interface of the extension.

Peiguo Guan: Response for application interfaces design. and write. functionnal APIs; Managing data in database.

Wanze Liu: Front-end design, write the interface that allows users. interact with a filed specialist or automatic interaction system

Jiachen Li: Chatbot design and training.

Yang Yang: Response for design format recognition and testing part for the program running; find the drawback of program and discuss with whole team

2. Problem Statements

2.1 Problem Statements

1. There is no such existing open-source assistant system. The traditional way is that the beginners tries to google the problems by themselves.
2. The beginners who want to use an open-source platform like Github may find it difficult to understand and learn codes on Github.
3. Without any mentor who can guide the beginners, the learning process would be tough and tedious.
4. Although there could be some friends can help you, but some of the problems cannot be solved by them and they are not always ready for help.
5. The beginner may lose their interest in learning open-source project.

2.2 Methods used to infer problem statements

We inferred our problem statements in 3 stages:

First, we have read some research paper focusing on these problems. According to Igor and Tayana[1], lacking of answer and delayed answers are one of these reasons that will make the newcomers give up.

Second, we have asked some friends and classmates who are learning

new open-source projects. They said that the most difficult thing they met while learning new projects is that they cannot completely understand the principle and the meaning of codes and google them sometimes may not get the correct answer.

Based on the information collected from above steps, we make a list of functions. First of all, we will provide a communication platform for the newcomers and previous contributors. Secondly, making a chatbot that could response easy questions from the newcomers which could reduce the pressure of the contributors.

3. Evaluation of ideas

According to group discussion, the scope of this project has been reache. In order to evaluate this project, SWOT analysis has been applied to the entire project to determine how closely a sprint is aligned with its growth trajectories and success benchmarks

Strength:

The main advantage of this project is that we can provide beginners with a better understanding of the open source project (mainly coding) that they are willing to learn. Moreover, they can learn this by talking to friends or developers who are professional in this area and talk with a chatbot.

Weakness:

Lack of users may have influence on user experience, especially when the automatic interaction system cannot answer satisfying answers to the questions, and some potential risks may occur during the process of project. In order to handle this, we need more communications and interactions in the group.

Opportunities:

With the increasing number of people start to learn coding, this project will provide beginners an easy way to get started and become expert in the future.

Threats:

Without the perfect reward system, it is difficult to keep developers answering questions which will cause a drop in the number of users that can lead to vicious cycle.

In conclusion, the project function we have planned is to combine online chatting and automatic interaction system together and apply it to an open source website (like Github).

Firstly, we will provide an immediate response function to open source website which could help the newcomers to understand the code better. This function also could alleviate the workload of the developers and give a quick response to easy questions.

Secondly, if the previous function cannot answer the questions that meets the requirement of the questioners, they could choose talking to the developers in the backend.

References

[1] Steinmacher, I., Conte, U.T., Gerosa, A.M. and F. Redmiles, D. Social Barriers Faced by Newcomers Placing Their First Contribution in Open Source Software Projects(CSCW), ACM New York(2015), 1384-1385.