Book Bot Project

Group 7

December 9, 2017

Abstract

This file is a project report about Group 7's Book Bot Project for CmpE451 class. Information in this file is up to Milestone 2.

1 Project Description

Book Bot Project is an assistant that can help users with everything related to books. It is a bot, and it works via Telegram. Working via Telegram makes the bot more accessible; users don't need to install anything extra and it is cross platform.

The base and most important functionality is to communicate with natural language. The bot can carry on a conversation with daily English.

Users can ask the bot about books. They can search books with different keywords, genres or authors, they can filter them in any way they want and they can sort them. They also can get information about specific books, see its current popularity via ratings or read the comments about the book.

Users can also give the bot feedback about the books they have read. They can comment on a book or rate a book. This will help the bot learn more about what the user likes and help it work more user oriented

The bot can also recommend the user some books depending on their previous ratings and taste of books.

2 Project Requirements

While implementing the project we've realized that some of the requirements needs some modification in order to work convenient and clean with the tools and frameworks that we're using. The followings are the requirements that we decided to modify.

1. 1.2.1.1.1.1.3-4-6-7-8-9 are removed
   Reason: We can only get information with the keywords of author, title and genre by using GoodReads API.

2. 1.2.1.1.4.3 is removed
   Reason: Since we're not getting publish date information, we cannot sort incoming data according to this feature.(GoodReads API)

3. 1.2.3.2.2 is removed
   Reason: Our backend system is not compatible for storing images and also our frontend side doesn't support showing images.(Django Framework)

4. 1.2.2.1.2 and 1.2.3.1.2 are removed. Reason: Signing up with invitation codes is not a necessary function, Admin can create a user credentials with specified type of user.

3 Project Milestones

3.1 Milestone 1

Book bot works fine for getting information purposes. (End of October)

3.2 Milestone 2

Book bot works fine for giving information purposes. (End of November)

3.3 Final Milestone

Book bot is able to recommend books. (Project is finished)

4 Project status

4.1 Deliverable List

1. Bot working on the cloud
2. Plan tool replacement
3. Giving Rating
4. Giving comments
5. Get comments
6. Moderator type
7. Admins/Mods Flagging Comments
8. Wit Training from Admin Dashboard
9. Testing Demo

4.2 Deliverable Status

TODO: write deliverable status. It is going to be in tabular form. See Table 1

Row Name	Row Name
Bot working on the cloud	Delivered
Plan tool replacement	Delivered
Giving Rating	Delivered
Giving comments	Delivered
Get comments	Not delivered
Moderator type Model	Delivered
Admins/Mods Flagging Comments	Delivered
Wit Training from Admin Dashboard	Partially Delivered

Table 1: An deliverable status table.

4.3 Deliverable Evaluation

1. Bot working on the cloud & Delivered
   (a) It is working now. However we should not run more than one server at one time (Even if it is a localhost, it confuses the chatbot).
2. Plan Tool Replacement
   (a) We couldn't utilize it too much, but the tool is working properly and old plan is transferred here.

3. Giving Rating & Delivered

   (a) This feature is working as it was expected. It creates the rating directly if the database have the book. If the book is not in our database. First book is created and then its rating inserted on the database.

4. Giving comments & Delivered

   (a) This feature is working as it was expected. It creates the rating directly if the database have the book. If the book is not in our database. First book is created and then its rating inserted on the database.

5. Get Comments

   (a) We couldn't implement this one.

6. Moderator type

   (a) We have added AdminUser and ModeratorUser types to the Admin Panel. AdminUser can modify nodes of the conversation tree, while ModeratorUser cannot. They both can modify Comments, but they both cannot modify TelegramUser objects.

7. Admins/Mods Flagging Comments

   (a) Comments now have a boolean field isFlagged. Both AdminUser and ModeratorUser can modify this field, therefore we have control of possible-harmful comments.

8. Wit Training from Admin Dashboard

   (a) We can send training data from our code. However we didn't implement it as scheduled procedure.

9. Testing Demo

   (a) We couldn't tested the demo version very well due to intensity of the other exams and projects.

5 Coding Work

You can find each team member's contribution to the code in the Table 2

Name	Coding Work
Ali Goksu Ozkan	Giving Comment to the database Giving Rating to the database Implemented Json format of intent-template sentence data
Irmak Kavasoglu	Created Comment data structure Created Rate data structure Created Book data structure Added AdminUser, ModeratorUser types to admin panel
Salih Sevgican	Fixed GoodReads API not giving info properly Created a copied test file for TelegramBot Created test users for AdminUser, fixed AdminUser not working properly
Melih Mutlu	Telegram bot bugs such as not working after a while is fixed Solved some Goodreads API problems

Table 2: Coding work table.

6 Summary

This project is a chat bot works on Telegram. It helps users to find books that fits their interests. Users only have to communicate with the bot in their natural language. One of the things that we want to achieve in this project is to give users a feeling of real conversation. While people think that they're having a smooth conversation, the bot should understand intentions and respond appropriately. We used Telegram as a platform for our bot. Telegram provides HTTP-based interface to implement chat bots. Implementing Telegram Bot interface into our project was an important part of first milestone since we need user interaction. It has been completed and Book-O-Bot is now working on Telegram. To understand what user send, we implemented a conversation tree structure. Telegram User model has a node field so that the bot can understand the current point in the conversation. When new message is received, bot has to first recognize intention of the message. At this part, we used Wit AI. It is a NLP tool to get a right action for a given sentence. Bot sends received message to Wit AI and gets its intent. This intent is used to move to right node in the conversation tree. After we have the right node for Telegram User, the bot is able to send back a reply. For example if user wants to search a book, bot sends keywords to Goodreads API. Goodreads returns us a list of books and we send the result to the user. User is then carried back to the initial node.