IDS Final Report: Stack Overflow Helper

Team members:

- Contact person: Yeju Zhou yejuz@andrew.cmu.edu
- Jiayi Weng jiayiwen@andrew.cmu.edu
- Xuanyi Li xuanyil@andrew.cmu.edu
- Xinyue Chen xinyuech@andrew.cmu.edu

Track: Interactive Visualization/Application

PDF: https://github.com/CMU-IDS-2020/fp-zhihu/blob/main/Report.pdf

Abstract: As the biggest online question-answer site for programmers in the world, Stack Overflow has a large impact on the developer community. However, through their public dataset, we discovered a continued decline in community activity and question-response rates. We propose our Stack Overflow helper with interactive visualization design by adding the recommendation system for tag-user and user-user recommendations to reduce the answer time, increase the answer rate, and facilitate interaction between users. Our tool fills in the gaps in the existing functionality of the Stack Overflow website. It is developed by Streamlit and currently available online.

Introduction

Background of Stack Overflow

Stack Overflow is an online community for programmers to learn, share knowledge, and advance careers. It generally features topics in computer programming. Stack Overflow follows a question-and-answer form that allows users to post questions, answer questions, and make comments. Users can actively participate in other users' questioning and answering by voting questions and answers up or down. Users can earn reputation points and "badges" by receiving upvotes. Moreover, Stack Overflow is also a platform where employers and professionals can connect, exchange views, and discover potential career opportunities. Until January 2019, Stack Overflow has over 10 million users and has over 16 million questions posted. However, there are no social features in the current website.

Limitations of Current Functionality

We identify two problems with Stack Overflow that limit its continuous growth.

The first problem is that, despite the large number of questions being posted on Stack Overflow, a lot of questions have never been answered. Starting in 2008, the percentage of questions posted that are not answered continues to increase every year. In the year 2020, over 35% of all questions have not been answered. These statistics indicate that when a user posts a question or searches for a question on Stack Overflow, there are about one-third of chances that he/she cannot get an answer, which discourages users from using Stack Overflow.

The second problem is that around 62% of users have never asked a question or answered a question. The number can be even larger if we take into account the users who only search for questions and answers by Google with a "Stack Overflow" keyword and have never officially registered as a Stack Overflow user. The poor engagement of Stack Overflow users is driving users away from asking questions, answering questions, and interacting with the other users.

In general, the current Stack Overflow functionalities fails to encourage users to get more involved in question answering and social connection with other users.

Overview of Our Solution and Contribution

To get more questions answered and get more users engaged in question answering and social connection, we propose Stack Overflow Helper, a recommendation system that 1) recommends users to answer particular questions based on the users' post history and 2) recommends users for a particular user to connect based on the similarity between their post history and technical expertise.

Our proposed recommendation system is motivated by the observation that a user prefers to answer questions with some particular tags (or topics). More specifically, we extract all posts (questions and answers) from a random user's history and find out that a large proportion (over a half) of posts are associated with the same or similar, small number of tags while the other small proportion of posts distribute across multiple tags. Inspired by this phenomenon, we consider the majority tags to be reflective of the greatest expertise of the user as well as the top answering preference of the user. Therefore, to increase the chance of getting a currently posted question answered, the Stack Overflow Helper recommends users with the greatest expertise in the technical area associated with the question and the most willingness to answer the question.

What's more, given the expertise and preference of every user, the Stack Overflow Helper also recommends users with similar expertise and preference to connect, which we call "friends". To enhance the interaction between connected users, an activity page is developed for users to track and monitor and the activities of their friends, including posts activity, badges earned, reputations earned, etc. The activity page acts as a stimulus that encourages users to engage more in question answering and social connections to gain more influence in the community.

In general, the main contribution of our work is a recommendation system that increases the answer rate of questions and enhances the engagement of users on Stack Overflow, **which does not exist in the current Stack Overflow website**.

Related Work

Recommender System

A recommender system often refers to the system that seeks to predict a user's preferences for different items. Well-received approaches include collaborative filtering and content-based filtering. Collaborative filtering assumes that if two users share similar preferences for items, then their tastes will also agree in the future. And a user's preference for an item is estimated by the similarity between his/her history and the other users', and other users' preferences for this item. On the other hand, content-based filtering decides a user's preference for an item using the item's features.

However, our application's scenario is different from the aforementioned ones because we are instead recommending users. But our approach also focuses on featuralization (of users) as does content-based methods (featuralization of items). And the actions that the users take (posted/answered questions) can be viewed as items in the recommender systems.

Q&A Platforms

Q&A platforms enable users to post questions and engage in the discussion of others' questions for knowledge sharing. Some platforms, such as Stack Overflow, Stack Exchange, and Ask Ubuntu focus on specific domains. There are also open domain Q&A platforms, such as Quora and Zhihu. Open-domain platforms tend to equip themselves with functionalities of social networking sites (e.g. follower/followee relationship). And we believe such functionalities (with careful design) can also benefit domain-specific technical platforms.

Methods

User Features

In the context of user recommendation, we take into account the following factors: a user's overall reputation, which is readily provided by the Stack Overflow dataset, the user's posted questions, and the user's answered questions. We distinguish between posted questions and answered questions because even though they both reflect a user's technical interest, they do not necessarily overlap. For example, one might be answering questions about *C* while posting questions about *Python* because he/she is a newcomer to *Python* while being proficient at *C*.

Reputation

The reputation metric is officially introduced here. It is a comprehensive metric for the user's overall contribution to the Stack Overflow community based mainly on the recognition of other users (including the question poster), instead of the mere quantity of actions (posting questions and answering questions) that the user takes. Users can recognize a question or an answer by upvoting and marking it as "accepted".

Posted Questions and Answered Questions

We embed a user's history of posted questions and that of answered questions based on the questions' tags.

We denote the current user set as U and the current tag set as T. $X_Q \subset \mathbf{R}^{|U|\times|T|}$ is the feature matrix of users w.r.t. posted questions and $X_A \subset \mathbf{R}^{|U|\times|T|}$ is the feature matrix of users w.r.t. answered questions. And $X_Q^{(i)} \subset \mathbf{R}^{|T|}$ and $X_A^{(i)} \subset \mathbf{R}^{|T|}$ correspond to the i^{th} row, or the i^{th} user's posted question history and answered question history respectively. Denote $Q^{(i)}$ as the set of questions that the i^{th} user have posted. For a question q, q_T is the set of question tags.

We denote $f(\cdot)$ as a mapping from a question to an one-hot vetor $\subset \mathbf{R}^{|Q|}$, where

$$f^{(k)}(q) = \mathbf{1}(T^{(k)} \in q_T)$$

We regard $f(q)$ as the embedding of the question q as an alternative to embedding the question using language models, as we did not manage to find either language models trained on domains similar to SO or supervised tasks and datasets in this domain. Then $X_Q^{(i)}$ is defined as:

$$X_Q^{(i)} = \sum_{q \in Q^{(i)}} f(q).$$

X_A is similarly defined. We further normalize $X_Q^{(i)}$,

$$\tilde{X}_Q^{(i)} = X_Q^{(i)} / ||X_Q^{(i)}||_2^2,$$

so that the normalized vectors have unit L_2 norm.

The definition of $\tilde{X}_A$ is a little different. For a user's answer history, we are not only interested in what questions he/she has answered, but also the contribution he/she has made. As the exact number of upvotes an answer receives is not given by the dataset, we assume that a user's answers receive similar amounts of upvotes in general.

$$\tilde{X}_A^{(i)} = R^{(i)} \cdot X_A^{(i)} / ||X_A^{(i)}||_2^2,$$

where $R^{(i)}$ is the reputation (scalar) of the i^{th} user. In this case, $\tilde{X}_A^{(i)}$ estimates the i^{th} user's contribution to the answering of questions from different fields (tags).

Tag-User Recommendation

The major use case of tag-user recommendation is recommending potential users who can engage in the discussions of a specific question based on the tags of the question. In this case we wish to quantitize the affinity of a given set of tags and a user. Denote $f_{TU}(\cdot)$ as the function that measures the affinities of a tag set and the users.

$$f_{TU}^{(i)}(\mathbf{t}) = \langle \tilde{X}_A^{(i)}, g(\mathbf{t}) \rangle,$$

where $\mathbf{t}$ denotes a certain tag set and $g^{(k)}(\mathbf{t}) = \mathbf{1}(T^{(k)} \in \mathbf{t})$. We then sort f_{TU} and return the top-k users.

User-User Recommendation

We design a user-user recommendation function so that users can find potential users to follow. Here we wish to quantitate the affinity of a user in the position of a question poster and the other users in the position of question answers so that a user can learn from his/her followees. This is accomplished by

$$f_{UU}^{(i,j)} = \langle \tilde{X}_Q^{(i)}, \tilde{X}_A^{(j)} \rangle,$$

where $f_{UU}^{(i,j)}$ is the score of the j^{th} user in the i^{th} user's recommendation view.

Deployment of the Recommendation System

For efficiency, we pre-compute $\tilde{X}_Q$ and $\tilde{X}_A$ and save them as sparse matrices. Besides, we allocate a dictionary for the current followee list of the users. In real-time user-user recommendation, the system will not recommend users that are already followed.

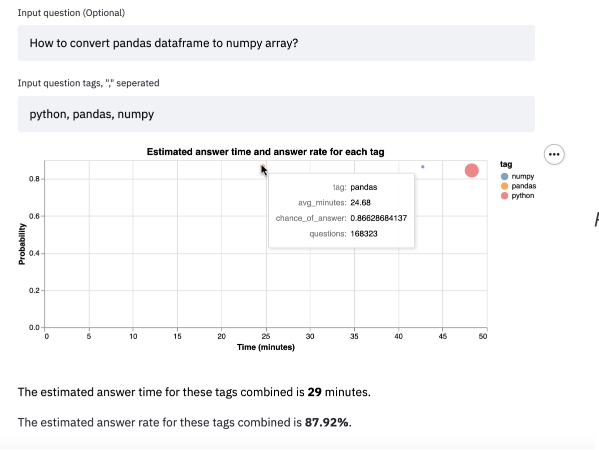
Results

We developed an application to visualize the whole recommendation system using streamlit. Based on the functionality, the visualization of our recommendation system is divided into two parts: Tag-User and User-User.

Interactive Page for Tag-User Recommendation

Tag-User Recommendation page consists of two parts: 1) tag-based analysis and 2) tag-based recommendations. Once the user type in the question he/she wants to ask and add proper tags for these questions on our application, we provide the above information on the page.

For tag-based analysis, we estimate the time need to get an answer and the chance of being answered based on the question tags. In Figure. 1, we show a sample use case here. Once the user inputs the question and tags using the text input, we processed historical data on answer time and answer rate for each tag and use a chart to show the data. We also provide an estimated answer rate and answer time for all tags input. The user could clearly see the distribution and could have a general idea of whether he/she could get an answer and how long he/she need to wait to get the answer.



Input question (Optional)

How to convert pandas dataframe to numpy array?

Input question tags, "," seperated

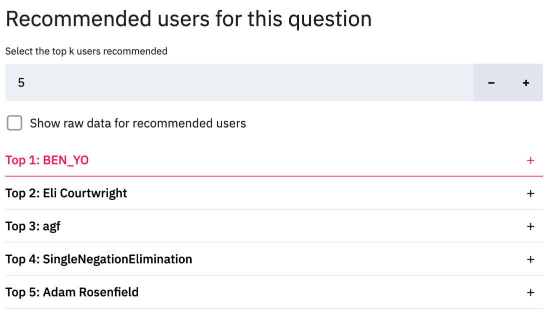
python, pandas, numpy

The estimated answer time for these tags combined is **29** minutes.

The estimated answer rate for these tags combined is **87.92%**.

Figure. 1: Tag-based Analysis

For tag-based recommendation, we provide the users who are more likely to answer this question. As shown in Figure. 2, the user could select how many recommended users he/she would like to see and click the "+" sign for each user to view detailed information. We provide personal page links and self-introduction in the detailed information.



Recommended users for this question

Select the top k users recommended

5 − +

□ Show raw data for recommended users

Top 1: BEN_YO +

Top 2: Eli Courtwright +

Top 3: agf +

Top 4: SingleNegationElimination +

Top 5: Adam Rosenfield +

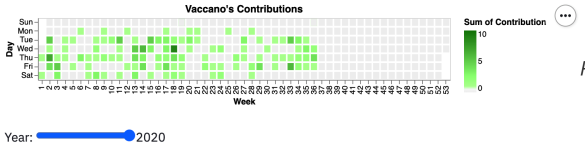
Figure. 2: Tag-based Recommendation

Interactive Page for User-User Recommendation

User-User Recommendation page focuses on the historical usage data for each user and helps to understand a Stack Overflow user better. For each user, we analyze the usage of Stack Overflow such as questions he/she asked and answers he/she provided. For each pair of users, we analyze their similarity and give the potential people he/she might interest in. We use two subpages to display the data: 1) Single User Profile and 2) User Recommendations and Timeline.

Interactive Page for Single User Profile

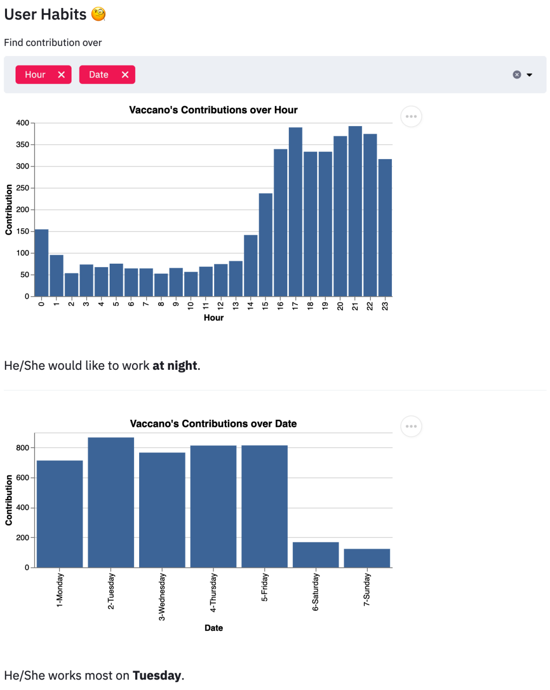
For single-user profile, the overall usage activities are displayed using a contribution graph as Figure. 3. Given a user ID and the year number, we generate the contribution distribution of that user on that year. This graph could provide a general idea of how this person worked annually.



Year: 2020

Figure. 3: Contribution Graph of a Single User

What's more, we provide visualizations of users' usage habits to better understand when a user usually works. Based on the selected dimension on time, we show the distribution of the contribution over that dimension and provide a simple analysis of that. For example, in Figure. 4 below, we show Vaccano's contribution over hours and days and it could be told that he/she usually works at night and on weekdays. These visualizations help to understand a user better.



User Habits

Find contribution over

Hour × Date ×

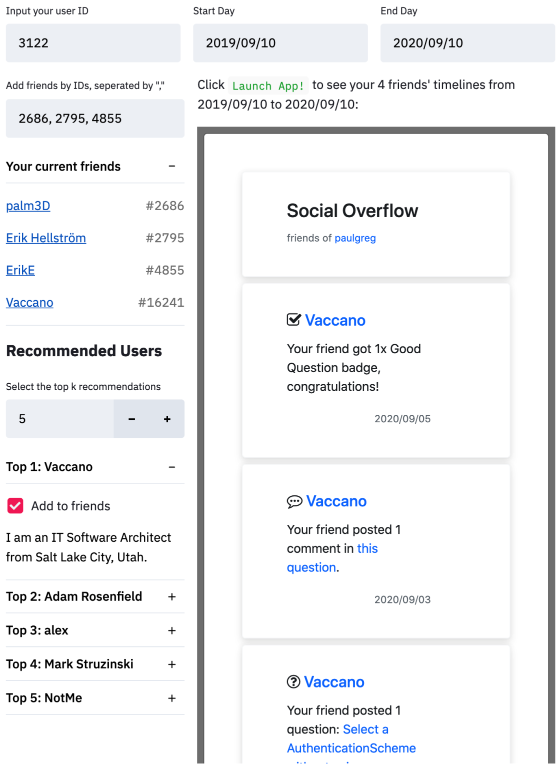
He/She would like to work **at night**.

He/She works most on **Tuesday**.

Figure. 4: User Habits Visualizations

Interactive Page for User Recommendation and Timeline

For user-user recommendation, we build the application page as a "Social Network" on Stack Overflow. This would allow you to be able to follow the activities and contributions of users that you want to watch/monitor. We show a sample use case in Figure. 5 below. On the left-hand side, we show the friends list and recommendation list. The user could add friends by searching user IDs or directly add from the recommendation list. On the right-hand side, we show the timelines for all "friends" within the selected time range.



Input your user ID: 3122

Start Day: 2019/09/10

End Day: 2020/09/10

Add friends by IDs, seperated by ",": 2686, 2795, 4855

Click `Launch App!` to see your 4 friends' timelines from 2019/09/10 to 2020/09/10:

Your current friends −

palm3D	#2686
Erik Hellström	#2795
ErikE	#4855
Vaccano	#16241

Recommended Users

Select the top k recommendations: 5 − +

Top 1: Vaccano −

☑ Add to friends

I am an IT Software Architect from Salt Lake City, Utah.

Top 2: Adam Rosenfield +

Top 3: alex +

Top 4: Mark Struzinski +

Top 5: NotMe +

Social Overflow

friends of paulgreg

Vaccano

Your friend got 1x Good Question badge, congratulations!

2020/09/05

Vaccano

Your friend posted 1 comment in this question.

2020/09/03

Vaccano

Your friend posted 1 question: Select a AuthenticationScheme

Figure. 5: Social Network Page Based on User-User Recommendations

On this page, our users could manage their "friends list" and follow their activities and contributions. One could learn from the questions and answers posted by their friends and promote the social relationships in the Stack Overflow community.

Discussion

Our application is easy to access online. The user can input their own user id and discover their own additional SO homepage and user-recommend system. An informal user test gave us several feedbacks:

- The recommendation system may not work with the new user which has no action on the Stack Overflow website. This problem can be solved by users adding their favorite tags at the beginning, therefore the question is the same as a tag-user recommendation.
- The definition of four types of SO users (questioner, answerer, question&answerer, do-nothinger) is too rough. The actual user group that contributes most actively may distribute in the first three classes evenly. It needs a more precise classification.
- The recommended user on the recommendation page needs to provide more polished information such as his/her preferred tags. Only the self-intro is not enough.

Future Work

Several promising directions can refine and polish this work.

For the recommendation system, currently, we only use a subset of Stack Overflow users' information to achieve the appropriate memory consumption in both preprocess and deployment phases. Future works include improving the efficiency of our recommendation algorithm and let it run with a memory-saving style.

For the interactive visualization and application, we can make the Google BigQuery requests run in parallel to reduce the latency. The visualization result also needs to improve, such as provide some refined results in user information summarization, and provide results that can be combined with existing functionality in the Stack Overflow website in a better way.