Verity : Volunteer crowdsourcing of fact checking

Aman Jain 160050019	Shreyas Pimpalgaonkar 160050024	Shubham Anand 160050060	Vinod Shekokar 160050016

27 November 2018

Contents

List of Figures

A Abstract

We propose to create an article routing system to route tagged articles to volunteers based on user ratings and expertise. One of the main goals of the project is to detect and report errors in articles. The routing of articles is performed via a custom rating algorithm designed by us. The final system will be in the form of a mobile application for both roles of the article writer and the volunteer.

B Specifications of the Application Interface

B.I Interface for the article writer

We plan to implement the interface of the article poster as in the same flutter application.
The interface for the article poster will contain:

1. **Login Page** : The same common login page for both volunteer and writer
2. **All Articles** : All articles page shows the list of the articles posted by the user till date. It also contains a floating widget to Create Article and one can click on an article to view article overview.
3. **Article Overview** : It gives the overview of article, like the number of comments on it, number of reviewers, average rating etc. It has a button to go to article details.
4. **Article Details** : Article details shows the summary of article including description, number of comments, number of ratings.
5. **Article View** : Shows the entire article details and color coded clickable sections. On clicking on those sections, a redirection is done where you can see the comments, ratings and average rating to those comments.
6. **Create Article** : Create Article page contains a form with following fields :
   (a) Provide article title and description
   (b) Write article body, newlines will be used to partition the article into sections
   (c) Select tags from auto-complete dropdown list.
   (d) Choose the number of users he wants to route the article to (optional), has a certain default value
   (e) Choose the time duration for which he wants to route the article to (optional), has default value

B.II Interface for the volunteer

The volunteer will be implemented as an Android Application using flutter.
The volunteer interface will have the following parts:

1. **Login Page** : A login page where he can login (and also sign up in the beginning) to review articles
2. **Home Page** : An card stack that contains article headings, description, tags, and publish time. The volunteer will be provided two gesture options on cards:
   (a) **Swipe right** : On swiping right on the article, the user will be taken to the article page.
   (b) **Swipe left** : This indicates that the user doesn't want to read this article, and will be shown the next article in the stack
3. **Menu button** : The menu button is present on all screens and will have options to view/edit personal information, edit tags and log out.
4. **Article Page** :
   (a) On swiping right from the home page, the volunteer will be taken to the article page
   (b) The article page is scrollable and has article body divided in sections.
   (c) On clicking on a section in the article, the user will be forwarded to a section details page.
5. **Section Details** : The section details page shows the section text followed by a comments ListView. All the previous comments on that section will be shown which he/she can rate (by options - strongly disagree, disagree, neutral, agree, strongly agree) and there is a form in which the user can add a new comment.

C ER Model

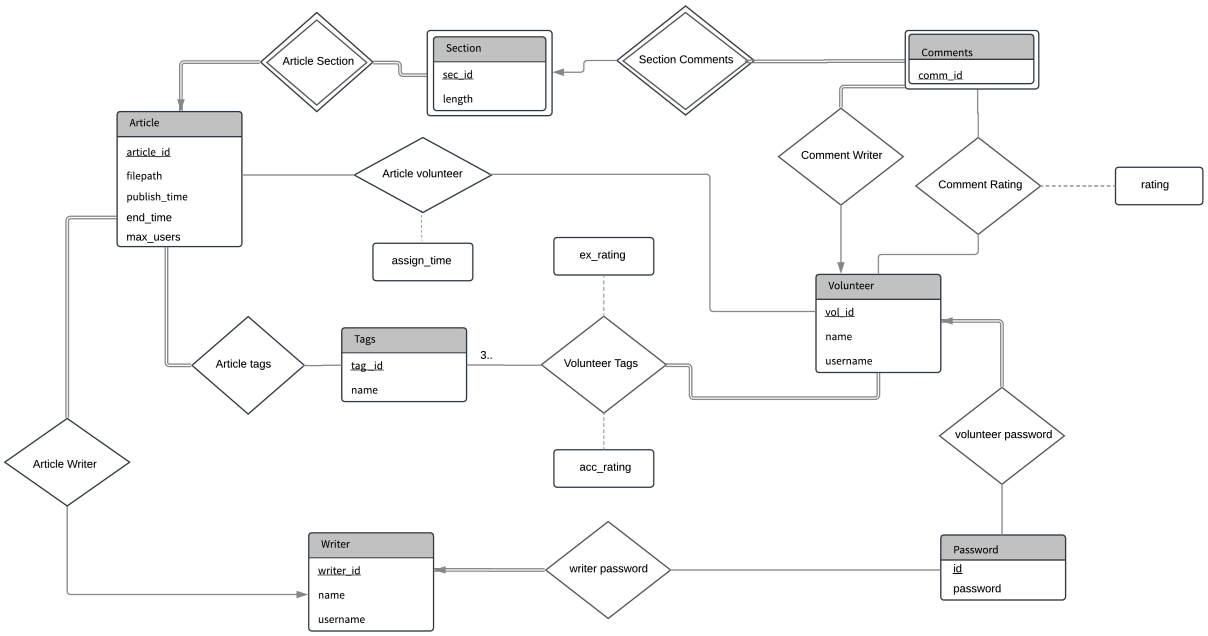

Figure 1: Entity Relationship Diagram

D Specifications of the Routing Algorithm and Rating System

D.I Routing Algorithm

For every article that has been created, it is be routed to the best volunteers selected by our rating scheme. The ratings are calculated on the basis of tags to the article and the rating of the volunteers (both experience and accuracy ratings) in his tags of expertise.
Then, as they begin to comment on the sections, those comments will be reflected to the other users who have not yet seen or not yet reported their comments on the article. Also, then these articles are routed to somewhat lesser rated people and this will keep going on until a saturation decided by our algorithm.

The routing algorithm has parameters as follows :

1. Tags of article
2. User rating and expertise
3. Number of comments and users forwarded to
4. Time since posting

The article writer gives the parameter of the max number users the article he wants to route.

The routing of the articles is calculated on users calculated by the similarity of tags vector with the weighted average of expertise vector and experience vector. The article has an expiry duration for the user. If the article has expired or a user has submitted his comment, it will then be forwarded to the next best volunteer in the priority queue.

Every article will be stopped routing when either of the two cases happen.

1. The global timer of the article expires

2. The desired number of users have given their comments on the article

D.II Rating the volunteers

The volunteers will have **two** ratings :

1. Related to their **experience** - The experience rating is defined for a user and a tag. It is between 0 to 10. It starts with zero for every tag and increases with comments.
2. Related to their **expertise** - The expertise rating is also defined for a user and a tag. It starts with 5 and increases or decreases according to the ratings of the comment by other users and his ratings to other users.

Now, the user's ratings are dynamic and will change after every trigger events.

The user's experience rating will **increase** if they comment on an article or they vote on other comments on the article.

The user's expertise rating will **increase** other people rate their comment as good.

The user's expertise and experience rating will **decrease** if they do not comment on the article which is routed to them (left swipe).

E Servlet Specifications

Servlet Specifications			
Servlet Name	Input	Output	Comments
Login	username, password	whether only writer logged in or volunteer	Common Login servlet for both volunteer and article - if volunteer role available then login with volunteer role, else login with writer role
Switch	vol_2_writer (0/1)	status	Switches role of user from writer to volunteer and vice versa
VolunteerAllArticles	-	all elements of article except body routed to them where volunteer_id is taken from session variable and tags of each article	Only those articles are routed which have status '0' or '2' with sorted in order of status and time_posted
VolunteerCreateComment	article_id, section_id	status	Creates comment and gives comment id, increases experience and expertise rating of the volunteer
VolunteerGetComments	article_id,section_id	all comments in the section	-
VolunteerGetSections	article_id	section_number, length	returns the length of all sections in the article
VolunteerIgnoreArticle	article_id	status	Changes the status of entry in article-volunteer table, decreases experience and expertise rating of the volunteer
VolunteerLogout	-	status	Invalidates Session

Servlet Specifications			
Servlet Name	Input	Output	Comments
VolunteerProfile	-	Profile information of Volunteer - name, tags and their experience and expertise rating	-
VolunteerRate-Comment	article_id, section_id, comment_id	status	Increases experience and expertise rating of the volunteer
VolunteerSignUp	list of tags	status	The tags are added to the volunteer-tags table
VolunteerSubmi-tArticle	article_id	status	The status of article in volunteer_article is changed to submitted, article is routed further, experience rating is increased
WriterAllArticles	-	List of all articles written by the writer	-
WriterArticleDe-tails	article_id	Metadata of the article like total comments, number of reviews, etc.	-
WriterCreateArti-cle	title, description, body, list of tags, end_time, max_users	status	Inserts into article table and article_tag table, select best n rated volunteers according their tags rating and insert top m of them to volunteer_article table with status - routed and rest with status - to be routed in future
WriterSectionDe-tails	article_id	list of sections of the article and number of comments in them	-
WriterSignUp	username, password, name	status	Insert into password table and writer table
GetTags	-	List of all the tags with tag_id	-
Config	-	-	Configuration and server variables
DbHelper	-	-	Provides helper functions to execute queries

F Testing

We have carried out testing in the following way:

1. We added users with predefined expertise and experience to the database, created tags and created some writers using SQL scripts.
2. We created some articles by different users in different areas of expertise
3. We observed cases of routing of articles with maximum users, beyond expiry time, and other corner cases.
4. We observed the accuracy of our dynamic rating policy and improve parameters by tuning them while testing.

G Future Work

These are some interesting additional features that can be added to our existing project:

1. We have provided all the functionalities in the app as of now but we can further extend our project to make the web app for the volunteer interface. This makes the interface really easy to use and convenient for the volunteer to verify articles, see statistics and rating existing comments.
2. We can analyse volunteer's behaviour to get statistics regarding which types of articles he generally ignores, which topics he comments on very enthusiastically and so on. This will help us to build his profile overtime and make our system better by suggesting him articles accordingly.
3. Writer can add geographic location to the article and then the articles will be routed to volunteers near or in that geographic location.