CS411 Team Backpacker Project

Final Report

Yifei Li, Bangqi Wang, Shengze Wang, Zhengbang Chen

Description:

Backpacker is a online platform where users can post articles about their past travels with specific details including locations, category and unique experience or stories. After a user submits an article, others may view this article and rate for the post.

Basing on the rating a user gives to a post, the website will automatically recommend other posts rated similar scores by all the users. Moreover, the user may find users with similar experience and find locations that might interests this user basing on the articles that he or she posts. And posts by such users and about such locations will be first analyzed by its text sentiment and be recommended to the user order by the positiveness of the post.

The social part of the website consists of two parts: firstly, we have an online chat-room where users may talk to all other users currently online and in the chat room. Secondly, user may send messages to and reply messages of other users.

After all, we would like to offer a great experience to our users to share their experience, read other users' adventures and find companions or make friends for potential travels in the future.

Usefulness of your project, i.e. what real problem you solved?

There are many websites where people can find information for a tour or a travel such as Expedia, Travelocity and booking.com where user may find the pricing for flight, hotels and attraction tickets. And there are websites such as the Atlantic travel, Google Travel and so on where users can find articles about a location written by a professional writer. There isn't actually a website or a platform where people can find first-handed experience by other travellers most of whom may be ametur writers but just trying to share their most real feelings and unique experiences instead of some over-polished articles which often seems distanced to many travellers like ourselves. Also, we usually find it hard to get connected with other travellers around the world who may have the intention or share similar value and excitements. Yes you can go onto Facebook and search whatever groups for travellings. But it's still hard and buggy when it comes to finding companions who wrote about their experience and whose self you can learnt about through their most authentic experience.

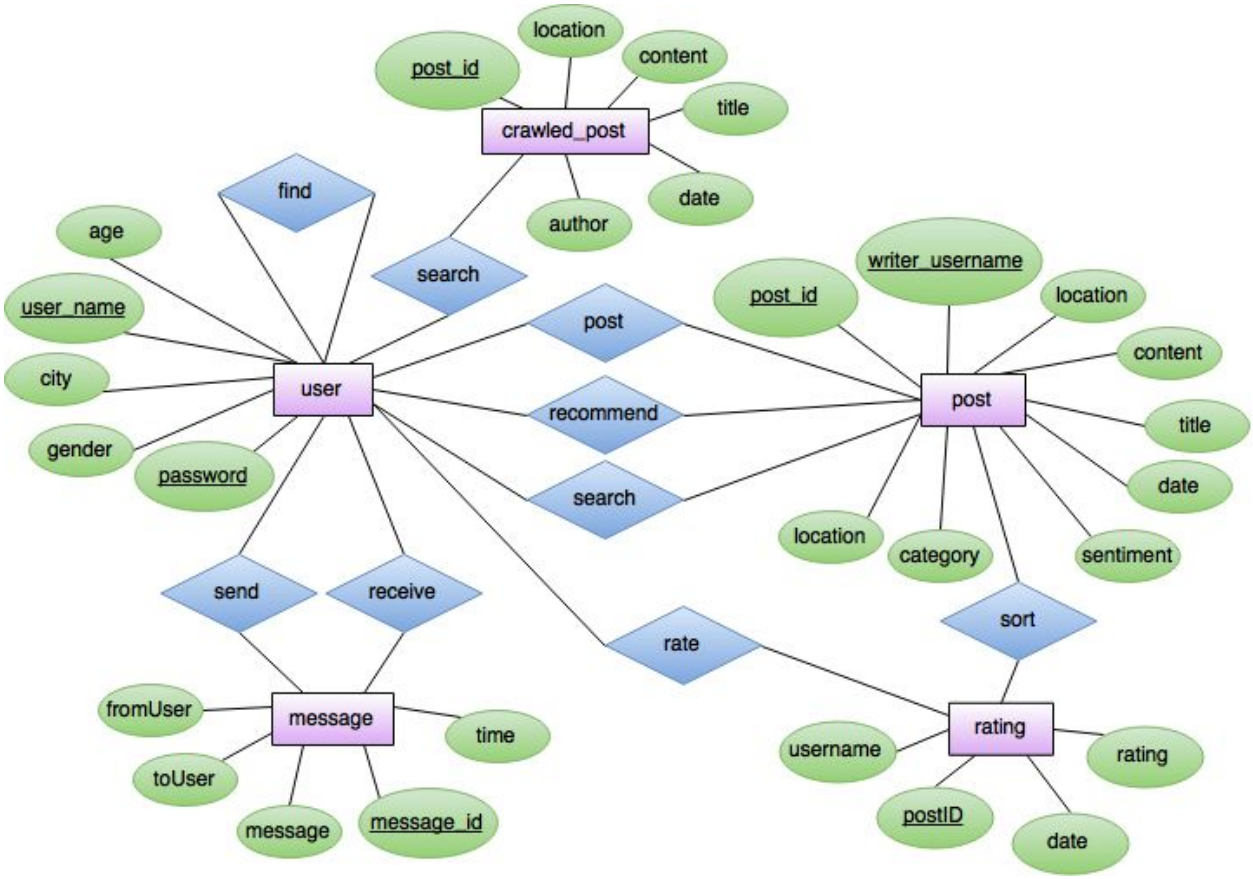
So the Backpacker, an online social platform, is born for this. It solves the real life problem that travellers might find it hard to share their experience to a certain group of people, the travellers and backpackers, and hard to find other first-handed authentic experience by other backpackers. On this platform, backpackers can also easily connect with other backpackers who share the same value and same preferences.

Discuss the data in your database:

We stored the user information, posts information, ratings of each post by the users and the relative ratings among posts for the purpose of recommendation, online chat room chat history,

messages going back and forth among users and data crawled from other website to recommend after a search.

Include your ER Diagram and Schema



Briefly discuss from where you collected data and how you did it (if crawling is automated, explain how and what tools were used)

Backpacker's data collection is largely based on the user inputs. Users post their experience online and rate for the posts that they read. The rating data will then be processed using Weighted Slope One Algorithm and a table dev will be generated to stored the relative ratings

between two posts; so that we may recommend other posts rated similarly. We also support online chatting and messaging so we also store the chatting history in our database. All the data above is generated by users through our front end.

Moreover, we crawled and stored data from several major trip-advising websites for the purpose of our search and recommendation system. And we also do sentiment analysis on the posts and thus generate sentiment scores basing on the post's contents. The scores is also stored in the database to be used for later suggestions.

Clearly list the functionality of your application (feature specs)

1. User registration and authentication
2. User can submit, update or delete a post
3. User can rate a post and received recommendations of posts with similar ratings by all users
4. User can find other users who has been to similar places and received recommendation of places that the user might be interested in
5. User can search a post, a writer or a location. The search bar will automatically prompt possible names in a drop down menu for the user to choose
6. Our application can analyze the sentiment in a post and give a score for recommendation purpose
7. Users can send messages to our users and reply messages from other users
8. Users can chat with others in a online chat room

Explain one basic function

One of our basic function is creating user post. It is mainly supported by SQL insert command shown as below:

```
$conn = new mysqli($servername, $username, $password, $dbname);
$sql = "INSERT INTO post (writer_username, location, content, title,
category, sentiment)
VALUES('$su_username','$su_location','$su_content','$su_title',
'$su_category', '$su_sentiment')";
mysqli_query($conn,$sql);
```

Generally, user can create an article which contains information of the city visited and trip type. Meanwhile, the sentiment rating of the article, one of our advanced features, will also be stored into database. This serves as the major function of our websites. Other users can view and rate these posts.

List and briefly explain the data flow, i.e. the steps that occur between a user entering the data on the screen and the output that occurs (you can insert a set of screenshots)

1. User can sign up for a user account which gives this user ability to create new posts and update and delete posts submitted. This user may view a drop down bar on the top right side of the screen with options for new posts, my posts, send message and messages received and logout;
2. In the new post page, user may enter information of his or her article including the title, content, location and category of the post and click submit;

3. Upon clicking “submit” the post will be uploaded to the database and the page will jump to “my post” where user can see the posts he or she has uploaded and can update or delete the post;
4. if user click delete, the post will be deleted permanently from the database and if user click update, the page will jump into a update page where the user may revise the content retrieved from the database and upon submission the data in the database will be updated;
5. On the search bar, user may type in writer name, post and location. The web page will retrieve a post with the name, or all posts of a writer or all posts about a location. These posts will be displayed on the page.
6. User can rate a post either in home page, the “find” page or the search page. The rating will get into our database and be processed using Weighted Slope One Algorithm and also generate the relative rating scores with all other existing posts. The processed data will be stored in a table called “dev”;
7. Now after the user has submitted the rating, the user will jump to another page with suggestions of other pages with similar ratings. This will use the data in the dev table: we recommend the posts whose relative rating to the post just rated is minimal, meaning most users have rated these two posts same scores and we would suggest this post to the user;
8. In the find page, if a user click “find” the page will checked with the database and find users who has been to similar places like this user and recommend places this user may like basing on those users’ experience

9. All posts are analyzed about their sentiments and the scores will be stored in the database. Upon recommendation, the system will recommend with more positive sentiments to the user on “find” page (which is different from the recommendation upon rating);

Explain your two advanced functions and why they are considered as advanced. Being able to do it is very important both in the report and final presentation.

1. The first advanced feature is our recommendation system, which consists of two parts:
    a. the rating-based posts-to-posts recommendation
        i. users may rate a posts and this rating will be stored into our database
        ii. then this rating will be processed by Weighed One Slope Algorithm to calculate the relative scores of this post with all other posts on the database. For example, post 1 is rated 100 by all users and post 2 is rated 120 by all users, their relative ratings will be 20 and stored in the dev table;
        iii. then for suggestions for the posts that is just rated, we will go back to the dev table and look for posts whose relative ratings by all users is minimal, meaning most user think these posts deserve similar ratings with the post just rated. And these posts will be retrieved and suggested to the user
    b. the user-location-sentiment-based recommendation
        i. when user clicks ‘Find’ button in page ‘find’, our recommendation system will go through the post table and look for this user’s travel history.

ii. our system will look through the post table again to find out other users that have common travel history and sort the temporary output table by the number of common travel history as ‘rank’.

iii. then the system will find the cities the user may interested. That is, the cities that selected users visited but the user didn’t. Calculating the attractiveness index by sum all the ‘rank’ of all people visited this city before and recommend the top 3 cities. People have no common travel history will not be counted.

iv. Searching the posts that talk about those three cities as primitive pool. Running sentiment analysis function to estimate the sentiment score of those posts. The larger the number is, the more positive the post will be. Only the posts that have more than 3 sentiment score will be recommended to the user.

v. user can also rate those recommended posts to get more recommendations with our rating-based posts-to-posts recommendation system.

I think our recommendation system is an advanced feature because of the difficulty of building connections through ratings among posts and locations among users. The intention is building a recommendation system like Amazon.com which can recommend items of simialr ratings or items that face to similar group of people. The abstraction and implementation of the collaborative filtering and the Weighted Slope One Algorithm is challenging since we hard coded it without using any scripted or APIs; What’s more,

with the help of sentiment analysis, our recommendations are sound and valuable after two rounds of filtering.

2. Another advanced feature of our DB project is our text sentiment analysis tool. Sentiment analysis aims to determine the attitude of a speaker or a writer with respect to some topic or the overall contextual polarity of a document. The attitude may be his or her judgment or evaluation or the emotional words. We improved popular natural language processing library, Insight and use our self-developed word-based sentiment analyzer. We use a word-by-word comparison between input text and our trainset to analyze the article using Naive Bayes rule. The datasets of our sentiment analyzer is trained by IMDB movie review (Sentiment polarity datasets v2.0), provided by Cornell University. We then combine two analyzers to generate our own outputs. We created a scaling for each sentences instead of creating polarity originally (pos vs neg). The results generated by our sentiment analyzer are then integrated by the recommendation system to process user preference.

   This sentiment analysis system should be considered as advanced feature. The sentiment analysis system is based on two different analyzers with different data set. They are both based on Naive Bayes Algorithm, and each perform well on different kind of sentences. They compensate for each other's defect and result in much more accurate sentiment output. To combinate two sets is complex and we need to find the way by ourself.

Describe one technical challenge that the team encountered. This should be sufficiently detailed such that another future team could use this as helpful advice if they were to start a similar project or were to maintain your project.

We've encountered several difficulties during development of Backpacker. The major problem shows up in sentiment analyzer. Since natural language processing is still not a fully developed concept, we have huge problem when dealing with our own sentiment analyzer. Since the analyzer is trained by mainly movie review. It works perfectly for short review while for long article, the analyzer shows its limitation. If there are complicated structures or in the article, our tools can not correctly get the result. So we tried stemming the word and test the left and right word of current word, for example "incredibly wrong", in the article. So we tried to use linear regression and adaboost technique to combine results from two different libraries. It turns out that result is not satisfying.

State if everything went according to the initial development plan and proposed specifications, if not - why?!

Initially, we planned on using real-time crawler as our advanced feature. However, we spent most time working on the setup for the python environment. Setting up connection between php and python is a painful experience. We first tried to create a process in the php script and passing parameters through the proc_open() function call. However, python had trouble receiving whatever came from php. Then, we used shell_execute() to call external program to fix the issue.

Another problem shows up when we try to deploy our web crawler on the server. We imported third party module beautifulsoup and mechanize and found out Cpanel doesn't allow student to install module on it. We solved the issue by adding module path as syspath. Also, it's extremely difficult for us to debug on the server even if our code runs locally perfectly. When encountering bugs, the browser just shows blank page, which is so painful for us to continue working on python in Cpanel.

Describe the final division of labor and how did you manage team work.

1. For basic functions, Shengze Wang and Zhengbang Chen were mainly focusing on front end design and implementation and Bangqi Wang and Yifei Li were mainly working on backend, building up tables and writing phps for submission and authentication;
2. For advanced functionalities, Bangqi Wang and Zhengbang Chen were working on the recommendation system including rating-based and user-location-sentiment-based parts. Yifei Li and Shengze Wang were working on crawling data and sentiment analysis systems
3. We managed to work as a team by dividing work well according to personal strength and preferences. We also have hackathon-alike meetings to hard code the website and lots of inspirations came out. Another benefit is that we were able to fixed any bugs aroused when we connect different parts together very quickly.