

CIT 58100 Project Guideline

Fall 2024

Total Points: 650

Project Description:

Objective

With the rapid growth of the Social Web, a large number of informal opinionated texts are available covering a wide range of topics. However, many people are overwhelmed with this vast amount of information and need help sifting through all of the data. Automated tools for automatically analyzing these opinions become necessary to help individuals, organizations, and governments in making timely decisions. To address this need, in this project you will develop an application that will analyze customer reviews regarding hotels. In this application, users will be provided with an overall rating of a hotel as well as ratings of the hotel's specific features. To validate the approach, you will build a query interface to analyze hotel reviews for a given user query.

One of the objectives of this project is to integrate a database to a real-life application. Another objective is to explore the full life cycle (requirements analysis, design, implementation, and testing) of a real-world problem.

Input Data Processing

On the hotel domain, a large number of customer reviews will be provided from open source datasets (e.g. opinRankData¹). This textual data needs to be processed and stored in a database to allow users to query a topic in real time. Figuring out the best way of processing this data will be one of the challenging tasks associated with this project and will be a great learning experience.

Hotel Rating Feature Selection

In this application, users should be able to perform a query for pre-stored hotel information in the database. To rate each hotel, you need to use a predefined set of features such as cleanliness, service, price, location, ... You need to analyze the input dataset to find the right set of features.

Hotel Rating Process

In this phase, you need to formulate an evaluation method (technique) of rating a hotel with its associated features (e.g. service, price, location, ...). Customers express their opinions (e.g. very

¹<http://kavita-ganesan.com/entity-ranking-data>

clean, expensive) and emotions regarding the hotel in their reviews. You could use these opinionated words or phrases (e.g. very clean, expensive) in your approach to rate a hotel. You should analyze input data set (and search on the web to expand the list) to find seed words for each feature as shown in Table 1.

Table 1: Sample seed words for different category

Features	Seed Words
Cleanliness	clean place, clean, very clean, dirty place
Service	excellent staff, poor service

Notice that seed words could be positive or negative. As a result, you need to assign weight for seed words. For example, you could assign +1 for a positive seed word (or phrase) and -1 for a negative word (or phrase). You need to use these weight values to calculate a hotel rating.

Prototype Design

As a proof of concept, you need to develop a working prototype for the hotel domain.

Basic Functionalities

Users should be able to query about a specific hotel and your application should display the overall rating and average scores of individual features as shown in Figure 1.

Figure 1: Sample query interface using Textbox

Test Display

File Edit Help

Enter Hotel Name: hampton inn majestic chicago Search

OR

Select City

Overall Rating: 4.65

Room: 4.6

Service: 4.65

Cleanliness: 4.8

Location: 4.72

Alternatively, users could select a hotel from a list for a given city. It should display overall rating of the hotel and rating for other features as shown in Figure 2.

Figure 2: Sample query interface using List

Test Display

File Edit Help

Enter Hotel Name: Search

OR

Chicago

- trump international hotel tower chicago
- park hyatt chicago
- hyatt regency chicago
- hilton chicago o hare airport
- hilton chicago

Overall Rating: 3.75

Room: 3.7

Service: 3.7

Cleanliness: 4.0

Location: 3.9

Advanced Functionalities

For a given city, users should be able to search by a specific search criterion such as “overall rating > 4.5” as shown in Figure 3. Your application should retrieve all the hotel names which fulfill the given criterion.

Figure 3: Sample query interface for advance search

Test Display

File Edit Help

Advance Search:

Chicago Search

Overall Rating: >4.5

Room:

Service:

Cleanliness:

Location:

- the talbott hotel
- sofitel chicago water tower
- trump international hotel tower chicago
- hampton inn majestic chicago
- residence inn chicago downtown river north

Under advanced search, users should be able to do a complex search such as they could search all hotels in a given city for the criterion “*room score* > 4 and *service score* > 3.5, and *cleanliness score* > 3.5 and *location score* > 4.5” as shown in Figure 4.

Figure 4: Sample query interface for complex search

Test Display

File Edit Help

Advance Search:

Chicago Search

Overall Rating:

Room: >4

Service: >3.5

Cleanliness: >3.5

Location: >4.5

the talbott hotel
sofitel chicago water tower
trump international hotel tower chicago
hampton inn majestic chicago
residence inn chicago downtown river north

Note: You need to ensure security and user privileges such as regular users should not be able to modify the database; they should only be able to query the database. On the other hand, an admin user should be able to add, modify, and delete data.

Bonus points:

Should be able to add new data such as:

- Add new features
- Add new seed words for an existing feature
- Add new review for an existing hotel
- Add new hotel and its review

Implementation Details:

Programming Language Requirement

For Database: You could use Oracle or MySQL.

You could use any programming language to implement application front end.

Implementation Requirement

After analyzing project requirements and the input dataset you need to come up with your logical design. In this process, you have to design ER diagram, DBDL by taking care of data integrity and update anomalies.

Once you finalize your logical design, you will transform that into physical design. You need to implement a workable prototype.

Submission Details:

Deliverable 1 (150 points):

Requirement analysis: Perform a through requirement analysis by analyzing project requirements and input data. Submit a logical design of your application. It should contain ER diagram, DBDL, and description of your design. Use Word or Lucid chart to draw the ER diagram.

Deliverable 2 (150 points):

Implement the *physical database*, *design the interface*, and *implement the basic features* for the project. In the process of creating the physical database:

- Store input data into CSV files. Use Java (or any program) to create these CSV files.
- Write PL/SQL scripts to import data from CSV file to your database tables.
- Write PL/SQL scripts to compute the rating by reading values from other tables. Use your script to compute and update the rating score.

You need to integrate front end application with your database to implement project basic features. Submit your project with basic features.

Deliverable 3 (150 points):

Finalize the interface and implement project advance features. Submit your project with advance features.

Final Submission (200 points):

Complete the project (including bonus features if possible). Submit the final project, project documentation, and report.

Note: Since this is a database course, I would expect most of the project functionalities would be implemented using database script instead of the front end application (such as Java).