Restaurant Recommendation System Using Linked Open Data

Tapaswi Reddy Busireddy
Computer Software Engineering
Arizona State University
Tempe, AZ
tbusired@asu.edu

Shreya Reddy Bolla
Computer Software Engineering
Arizona State University
Tempe, AZ
sbolla3@asu.edu

Abhishek Mohabe
Computer Software Engineering
Arizona State University
Tempe, AZ
amohabe@asu.edu

Mythri Meka
Computer Software Engineering
Arizona State University
Tempe, AZ
mmeka@asu.edu

Sasikanth Potluri
Computer Software Engineering
Arizona State University
Tempe, AZ
spotlu11@asu.edu

***Abstract*—Although there are several applications that recommend restaurants to a user, each of the existing application lack one or more necessary filters in a single place. To resolve this issue, the paper introduces a new restaurant recommendation system that, in addition to showing the ratings, price range and other common filters related to a restaurant, will showcase the calories of the dishes at the restaurant. The system will also recommend restaurant options based on the emotion selected by the user. Along with this filters, the system also provides option for choosing Michelin Starred restaurants from across the country. By utilising several restaurant-related datasets, linked data is created which can then be queried to retrieve restaurants based on user-selected filters.**

Keywords—restaurant, recommendation, rating, price, calories, emotions, semantic web, linked open data, ontology, fuseki server

I. Introduction

Food is one of the most important aspect of life for a human. And many people around the world spend hours in choosing a restaurant which is favourable to them in terms of ratings, prices, and cuisines. In addition to this, people, nowadays, are being conscious about their eating habits and are trying to maintain a healthy lifestyle. For this reason, they prefer to know the calories of a food item in the restaurant before opting for it. However, none of the existing systems today provide a note of the calories in a dish at their restaurants. Thus, the restaurant recommendation system will provide the number of calories, or the calorie range for a particular restaurant thus providing the user with more information regarding a restaurant. A recommender system helps find personalised results for a user based on filters in a complex, vast system [1].

In addition to presenting data about the calories of an item, the restaurant recommendation system will also help user choose a restaurant based on the emotion they are feeling. For any emotion presented to the system, it will provide a set of restaurants with suitable cuisines, along with any other filter selected by the user.

The data is accessed from several datasets which will help the system be more accurate by achieving thousands of restaurants. This data is then converted into the format of linked data which can then be retrieved with the help of semantic queries.

The ultimate goal of restaurant recommendation system is to create a user-interactive system where the consumer will be able to access thousands of restaurants and also filter them extensively based on their interest.

II. Problem Definition

There are lot of recommendation systems. We have all the typical constraints including price, ratings. But along with that we are planning to having calories and suggesting places to eat at based on emotion and location. The food industry is a multi-billion industry with millions of restaurants and fast food chains always innovating their menus to attract more customers. Using the Web Semantic Engineering, we can recommend the restaurants based on different cuisines to the customers. Tremendous amount of data such as different types of dishes, ratings, calories, etc. can help in luring new customers to try out different restaurants. We will be using an already available restaurant dataset in the Kaggle and we will be able to produce insights from it. The dataset is freely available and we will create triples from this dataset and further creating SPARQL Protocol and RDF Query Language (SPARQL) queries for the user to view the recommended restaurants.

Although, there are a lot of systems which recommend the restaurant based on the user preference. Our system has all the necessary features that a system existing alone doesn't have. So, this recommendation system helps people i.e., customer with the best way possible to choose the needed restaurant in the best way possible. To achieve this we are using the semantic web engineering which helps to fragment data according to the user interest and create the application with the all different features included as discussed. Semantic web defines the meaning of the data to the system and can add additional features to wherever possible based on the user requirement. The ontologies created are transferred using the Application Programming Interface (API) which retrieve the best restaurants based on user's preference. Our system also has a additional feature where we can select our comfort food based on the emotion. This system can be built in many ways possible but the semantic web can help the system to be built according to the user emotion and requirement with the datasets which can help in building the system in easiest way possible. Hence, this would be a system which links different things together to make sure everything falls under user satisfaction as personalised product recommendation helps in serving the customers with the relevant products which are highly served.

Several datasets are being used to extract information related to restaurants. The data is then cleaned and processed to extract necessary information, which is then transformed into Linked Open Data (LOD) thus obtaining data that is only necessary [2].This data can then be retrieved using the API's based on the user interest. This system on the whole is combination of all the features that a user could ever want based on the need. Finding the right system with all the features is also tough. So, that can be eliminated with using this system where the user finds whatever he needs in a single restaurant recommendation system.

III. RELATED WORK

A. Nutritional Semantic Recommender System For The Elderly

The nutritional semantic recommender system is a recommender system for elderly which provides the users an opportunity to build their own diet plans based on the recommendations and guidelines provided [3]. The system uses semantic similarity measure to provide recommendations about the information present. The system also ensures to use reasoner based on which items are closely aligned to the user's requirements. In addition to this, the system has a user-interface that is elderly-friendly. The database is represented in the form of Web Ontology Language(OWL) ontology which contains a nutritional, as well as, a user profile ontology.

B. Drug Encyclopaedia Linked Data Application

E-learning recommender system is another system that play a crucial role in assisting the students in finding useful and relevant learning materials that match their learning needs [4]. A fascinating area of research is the use of ontologies for knowledge representation in knowledge-based recommender systems for online learning. Ontology is used to convey knowledge about the student and educational materials in knowledge-based recommendations for e-learning resources. Additionally, it became clear that combining knowledge-based recommendations with other recommendation methods might improve their effectiveness of E-learning recommendation systems.

C. Movie Showtimes Recommender System

In this recommendation system, the location, crowd and time of the movie are considered and showtimes are suggested [5]. The user initially updates their profile after which time and location related filtering is carried out and then suitable showtimes are recommended. The information about these aspects is stored in an ontology by linking the data in the form of linked open data. The linking of the data facilities easy retrieval of information.

D. Publishing Historical Places and Old Maps

Geocoding applications for Linked data applications maintains historical data of geographical locations such as gazetteers and maps using SPARQL end points. The service is used for real time maintenance of maps and geo ontologies in real time [6]. This is made possible by using legacy catalog systems.The locations may be displayed on both historical and contemporary maps, as well as with extra contextual Linked Data connected, to help people better comprehend past locations. Also the service may be utilised and expanded in a federated manner by including additional distributed SPARQL endpoints or other web services with an appropriate API into the system.

E. Semantic Web Technologies at Pinterest

OWL has a wide range of application in the present applications and many applications are moving towards semantic web technologies to improve the application. One such example is Pinterest. In 2018 Pinterest opted to use semantic web for optimising the data and to create graph of users and content [7]. It uses a knowledge graph for representing vast amount of data in its database as well as information about its users in the application. This feature would help to discover the type of content to display for every Individual based on their interest as well as the suitable ads to display for marketing a product.

F. Uber Eats

Market-based recommendations are one resemblance. Uber Eats cater the selection and ranking of restaurants to our understanding of what eaters seek, from search results to the list of eateries and explicit suggestions on the app's home screen. By using multi-objective optimisation, we can ensure that restaurant partners get a fair amount of exposure in the app depending on user interest and assist diners in discovering a wide variety of eateries [8]. Eaters, restaurant partners, and delivery partners make up the three sides of the UberEats marketplace. Customers use the site to find and order food. The platform serves as a sales channel for restaurant partners to find clients. Additionally, delivery partners make money by procuring and delivering meals from restaurants to consumers

G. Content-Based Recommendation System

Another interesting system is the Content-based recommendation system. Similar products are recommended based on a specific item using content-based recommenders. This technology generates these recommendations using information like the description, genre, type of restaurant, etc. These kinds of recommendation algorithms work under the premise that if a person enjoys one item, they will also enjoy something similar to it [9]. For instance, if a customer orders a lot of spicy Indian food, they may start to realise that there are more Indian establishments serving spicy food in the app. At the same time, more Asian recommendations could appear.

The restaurant recommendation system uses a similar methodology of retrieving data and presenting it to the user. The data obtained from the datasets will be processed and converted into the format of linked open data in an ontology. This data will then be retrieved with the help of queries (based on the filters selected by the user). However, the restaurant recommendation system will also be different from the previous existing systems as this system will present more filters to the user and also several new features(in the form of filters) which are not available to the consumer until now, such as the calories of a food item.

IV. Approach

The problem that we are trying to solve is to recommend the restaurants listings in a good visualisation based on the preferences set by the user. To solve this problem we will utilise the semantic web engineering techniques, utilising open source data sets, and writing the logic to suggest the best restaurants. A sample query could involve querying the restaurant listing based on the cuisine type such as European, Asian, American, Continental, etc. or the user could also get the list of restaurants based on the review ratings.

To come up with this model, we approached the problem to solve it rationally. Our main focus to develop a good restaurant recommendation system was to utilise the data, logic, and the available data analysis instead of plain intuition based approach. Therefore, the dataset is the core of building this restaurant recommendation system. We had researched for open source datasets available on the internet and were able to get the best suited datasets for our use. We collated all the datasets and merged them into one complete dataset suitable for our use for the restaurant recommendation system.

Additionally, we performed data processing tasks such as data cleaning on the merged dataset in order to ensure the integrity of the collated dataset. We utilised Python scripts such as Pandas for the data processing. The data processing involved removing null values, removing the duplicates, and removing the missing values from the dataset.

Post the data manipulation activities, we started to solve the problem in the ontology context since we had to create ontologies in order to solve the problem using semantic web engineering. While defining the ontologies, we came across defining the structured way of classes, object properties, data properties, and mapped their correlation to ensure good compatibility. We utilised the Protege tool to create the ontologies by mapping the CSV datasets to an ontology.

Once we are concrete with the ontology and the OWL file, we will deploy the instance in the AWS EC2 server that will support the SPARQL queries to be triggered using Fuseki server through SPARQL endpoint [10]. The user could basically query through a web GUI application(in the form of filters) and the results will be showed in a trivial GUI page for the user's preference. We will try to populate the details of the recommended restaurants as per the user's preference.

V. High Level Design

The work flow of system at high level is mentioned below In the Figure 1. This Application contains of three high level components Front End , Back End and Cloud servers such as Micro soft Azure, Google Cloud and Amazon AWS.

A. Front End

Frontend consists of Graphical User Interface, which enables the user to select the filters for the restaurants based on their choice, such as Cuisine, Location, Emotion, etcetera. The REST APIs are integrated into the front end user interface which gets triggered based on the user input.

B. Back End

In backend we write the REST APIs that are responsible for fetching data from the cloud servers. The API message queue is passed through various layers which is responsible for querying the cloud server. The backend API connects the user interface to the cloud server.

C. Server

In backend we write the REST APIs that are responsible for fetching data from the cloud servers. The API message queue is passed through various layers which is responsible for querying the cloud server. The backend API connects the user interface to the cloud server.

The linked data that we have cleaned and filtered will be loaded into the cloud server. It consists of both owl files and the restaurant dataset which will be considered. We are going to use AWS cloud server which provides good compatibility for querying the linked data. The REST API's query the data present in the cloud server and allows for filtering and displaying the data based on user's input.

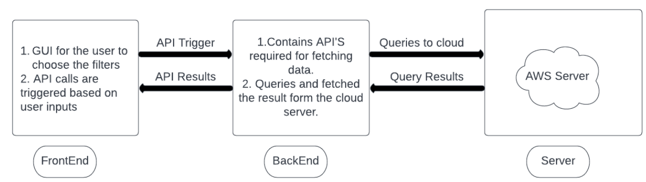

Figure 1: High Level Design

VI. Ontology Design

After the data is collected from different sources it is combined into a single dataset with all the features we want. Now, the data is cleaned and processed so that all the null values, missing values from the dataset are removed properly, with only the elements and the features necessary included in the project. After the data cleaning, the data is now processed into the CSV file with all the data required. The design now can be processed into the ontologies with a structure corresponding to the data that is cleaned and processed.

The design of the ontology is then created with the dataset processed. We have created the ontology file from scratch looking at the data we have. First we created the classes(to indicate domain concept)[11] which are disjoint and which are subclasses to the main class. Then we have created the properties(describe relation between objects) which have the domain and range. We have also included the transitive, symmetric and functional properties for the ontologies with the features to ensure the ontology has all the class hierarchies, property hierarchies and also the object properties with some of the important data properties to the classes.

In our design, we have the following classes in the ontology like the Category class which acts as a feature in the system to classify Breakfast, Dinner, and Lunch. The Cuisine class has different subclasses like American, Asian, Austrian, Californian, etc which defines the various cuisines featured in the system. The Emotion class has emotions where the user can choose the food based on the emotions and has subclasses like Boredom, Stress, Depression etc. We also have a restaurant class which has subclasses in which the restaurant ID, Location, Calories, Price and Rating are part of. The Price class is again sub divided into three different subclasses which helps users to select the restaurant based on their budget. For the classes designed we have different object properties which define the properties of the classes.

For the classes defined we have classified some of the object properties like has_category, has_name, has_price etc in which the has_SameLocation is marked symmetric as the restaurant with location is equivalent to another restaurant having the same location. The has_BetterRatingThan is marked transitive as for instance if the restaurant A has better rating than B and B has better rating than C then A will obviously have better rating than C. So, hence we have these object properties defined which helps in creating the better system. We also defined the data properties for ID, location, name, rating etc which are fixed. So, hence we defined them as the data type properties.

After defining all the things inside the ontology file we then need to run the reasoner which is needed to ensure that the data and the restrictions within the property provided are consistent and can be used efficiently in the system. Running a reasoner and ensuring the consistency of data is an important step in the ontology design as the reasoner helps to identify if the classes are proper or not and if the design is consistent with the properties. We used the Hermit reasoner in the Protégé tool, which is a type of the reasoner, to make sure that the data is consistent with the classes and properties defined.

This way, we cover all the properties in the ontology file which will be the integral part of the data when trying to access it through the SPARQL queries. This design is implemented so that it is efficient and retrieves back to the system when the user needs it through the linking applications which are connected to the AWS Cloud server.

VII. Ontology Visualization

The visualization of the restaurant ontology has been created using the VOWL plugin in Protégé tool and displayed in Figure 2 in a graphical format [12].

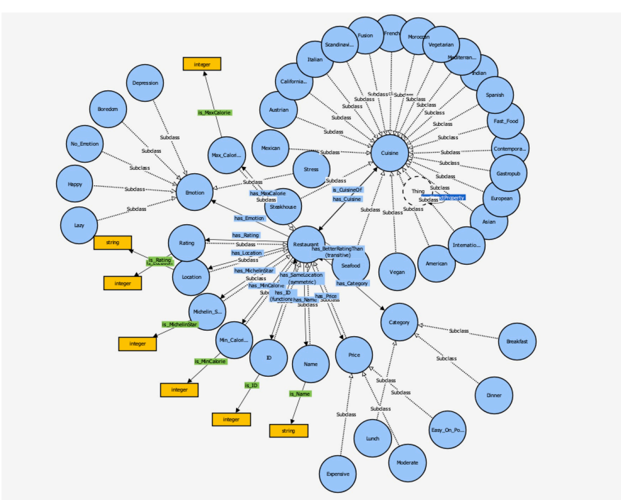

Figure 2: Visualization of the Ontology

VIII. Data Collection and Processing

The raw data was collected from the website 'Kaggle'. This raw data has statistics regarding the restaurant details such as name, location, reviews and ratings. These unprocessed numbers provide us with a solid base for the additional manipulation needed for this project.There are numerous ways for data to be duplicated or incorrectly categorized when merging multiple data sources. Even if results appear to be correct, they are unreliable if the data is inaccurate. Because the procedures will differ from dataset to dataset, there is no one definitive way to specify the precise phases in the data cleaning process. We have followed the below steps:

- Removed unwanted observations from our dataset, including duplicate observations or irrelevant observations.
- Fixed structural errors.
- Filtered unwanted outliers.
- Handled missing data.

A. Dataset - 1(Pizza Restaurants) [13]:

This dataset named "Pizza Restaurants and the Pizza They Sell" has columns which are restaurant name, location across US, ratings, and reviews. After cleaning this dataset, it was then merged into the main restaurant dataset under Italian cuisine.

B. Dataset - 2(Fast Food Restaurants) [14]:

This dataset named FastFoodRestaurants.csv has columns - address, city, country, keys, latitude, longitude, name, postalCode, province, websites. After cleaning this data set and including it in the final dataset which we will be working on this will have the details about the address.

C. Dataset - 3(Asian Restaurants) [15]:

The Asian Restuarants dataset has information about various asian restuarants across the United States. The dataset has restaurant name, location(in various forms), cuisine, and ratings. This data was then cleaned to get rid of duplicates and was then merged with the restaurant dataset along with necessary columns.

D. Dataset - 4(Subway Restaurants) [16]:

The dataset "Subway.csv" is a combination of all the subway restaurants across the country. The dataset has columns like location, calories, timings, zip codes. The subway dataset is a large dataset which has a lot of data so we decided to preprocess the dataset before combining or merging it to the main dataset. The dataset is preprocessed and removed all the unwanted data like null values, duplicates and missing values which made the dataset to be large. After processing the dataset, now the dataset is merged to the final dataset which was again cleaned and processed. In addition to this we have concatenated the word "Subway" to each of the restaurant as the dataset only had locations of several subway restaurants.

E. Dataset - 5(Michelin Star Restaurants) [17]:

This dataset that is named michelin-restaurants.csv (including one-star-michelin-restaurants.csv, three-stars-michelin-restaurants.csv, two-stars-michelin-restaurants.csv) have columns name, city, longitude, cuisine, zipCode, city, region, latitude. After cleaning this data set and including it in the final dataset which we will be working on this will have the details about the cuisine and michelin star category.

F. Dataset - 6(Emotions) [18]:

This dataset that is named emotion.csv has columns emotion and emotion count. After cleaning this data set and including it in the final dataset which we will be working on this will have the details about the emotions (Stressed, Depression, Lazy, None).

G. Dataset - 7(Ratings) [19]:

This dataset named ratings.csv has columns which are User_id, Food_id, Rating. After cleaning this data set and including it in the final dataset which we will be working on this will have the details about the rating and Id itself.

H. Dataset - 8(Calories) [20]:

We have considered the dataset which provides a list of various foods and their calories. Using this data we mapped the calorie range to each restaurant(by dividing into minimum calories and maximum calories).

I. Dataset - 9(Vegan Restaurants) [21]:

The vegan dataset contains several restaurants which have vegan and vegetarian cuisines, offering restaurant locations of different price ranges across the country. This dataset was cleaned to get rid of unnecessary information and then merged to the final dataset.

In addition to these datasets, we included a column called Category, which categorizes the restaurants into Breakfast, Lunch, or Dinner. The final dataset which we have created has 11 columns named ID, Restaurant Name, Location, Cuisine, Calories(min and max), Price, Rating(1-5), Michelin Star(0-3), Emotions, Category.

IX. Implementation Plan

The final dataset has been created and the corresponding ontology has been designed showing the several classes and properties discussed by the team.

In the next steps, we will be using AWS cloud server to store the datasets. An EC2 instance will be created into which the restaurant dataset and the ontology will be loaded. Apache Jena Fuseki server will be installed in the cloud which will help retrieve required data with the help of SPARQL queries [22].

Simultaneously, the team will be working on frontend(using ReactJS) and backend, as well. We will be developing several REST APIs based on several queries which will help retrieve restaurant names based on user's selected preferences(in other words, filters) using Java. The data retrieved from the REST APIs will be fed into the application which will be presented to the user through the interface based on the selected filters. This will be the implementation plan followed by the team.

The tasks to be completed include 1) loading the dataset and ontology into cloud server, 2) setting up fuseki server in cloud, 3) developing the frontend template for user interaction, 4) writing SPARQL queries to retrieve data, 5) developing APIs to retrieve data from cloud server, 6) integrating the entire system to present a user-friendly application.

X. Roles and Responsibilities

Abhishek worked in combining and merging all the datasets from various resources, in addition to working on the Asian Restaurants dataset. He also made sure that we had the data collated in the columns specified as per our overall design. Additionally, he ensured the data consistency across the 10+ data sources while merging into a universal dataset for use for our recommendation system.

Sasi had worked on data cleaning along with my teammates. We have taken all the raw data and cleaned it by eliminating all the null values and removing all the duplicate records. He also worked on cleaning and finalising the complete Subway dataset. I have been part of designing the ontology and deciding on classes.

Shreya has worked on extracting the data from different datasets and designing the features of the system which helps the data for data cleaning and processing. She ensured that various columns were rich with data across the datasets. She also worked on creating the ontology file based on the design of the dataset with testing it with the reasoner for the consistency of the design. Thus, explaining the same in the ontology and design of the report. Shreya also helped in visualizing the ontology created for the system.

Together with a teammate, Mythri focused on data cleaning and worked on finalizing the Subway restaurants. All of the raw data was extracted and cleaned, with all null values and duplicate records being removed. In addition to this, Mythri worked on classes and several properties between the classes.

Tapaswi worked on cleaning the Michelin Star dataset, along with the Vegan Restaurants dataset and Pizza Restaurants dataset. She has also worked on designing the ontology by helping establish several properties between class(including populating transitive and symmetric properties along with teammates) and also ensured consistency in the ontology by checking with the reasoner. In addition, she has worked in visualization of the ontology for the restaurant owl file.

References

1. Tommaso Di Noia, Vito Claudio Ostuni, "Recommender Systems and Linked Open Data", In Reasoning Web International Summer School, pp. 88-113, Springer, Cham, 2015. Available: https://link.springer.com/chapter/10.1007/978-3-319-21768-0_4
2. Thomas Wever, Flavius Frasincar, "A Linked Open Data Scheme-Driven Approach for Top-N Recommendations", In Proceedings of the Symposium on Applied Computing, pp. 656-663, ACM, 2017. Available: https://dl.acm.org/doi/10.1145/3019612.3019843
3. Vanesa Espín, María V. Hurtado, Manuel Noguera, "Nutrition for Elder Care: a nutritional semantic recommender system for the elderly", Expert Systems, April 2016. Available: https://onlinelibrary-wiley-com.ezproxy1.lib.asu.edu/doi/pdfdirect/10.1111/exsy.12143
4. John K. Tarus, Zhendong Niu, Ghulam Mustafa, "Knowledge-based Recommendation: a review of ontology-based recommender systems for e-learning", Springer, January 2017.
5. Luis Amar Colombo-Mendoza, Rafael Valencia-García, Alejandro Rodríguez-González, Giner Alir-Hernández, José Javier Samper-Zapater, "RecomMetz: A context-aware knowledge-based mobile recommender system for movie showtimes", Elsevier, September 2014. Available:
6. Esko Ikkala, Eero Hyvönen, Jouni Tuominen, "Geocoding, Publishing, and Using Historical Places and Old Maps in Linked Data Applications", In DHN, pp. 228-234. 2018.
7. Gonçalves R.S. et. al. (2019) Use of OWL and Semantic Web Technologies at Pinterest. In: Ghidini C. et. al. (eds) The Semantic Web -

ISWC 2019. Lecture Notesin Computer Science , vol 11779 Springer, Cham. http://doi.org/10.007/978-3-030-30796-7_26

8. Food Discovery with Uber Eats [Online]. Available: https://www.uber.com/blog/uber-eats-recommending-marketplace/
9. Brief on Recommender Systems [Online]. Available: Shttps://towardsdatascience.com/brief-on-recommender-systems-b86a1068a4dd
10. M. Udin Harun Al Rasiyad, Iwan Syarif, Ilham Akbar Hasbiya Putra, "Linked Data for Air Pollution Monitoring", In IEEE. 2017. Available: https://ieeexplore.ieee.org/document/8228565
11. Chin-I Lee, Tse-Chih Hsia, Hsiang-Chih Hsu, Jing-Ya Lin, "Ontology-based tourism recommendation system", in IEEE. 2017. Available: https://ieeexplore-ieee-org.ezproxy1.lib.asu.edu/document/7939242
12. T. Sigwele, A. Naveed, Y.F. Hu, M. Ali, J. Hou, M.Susanto, H. Fitriawan, "Building a Semantic RESTFul API for Achieving Interoperability between a Pharmacist and a Doctor using JENA and FUSEKI", In Journal of Physics. 2018. Available: https://iopscience.iop.org/article/10.1088/1742-6596/1376/1/012015/pdf
13. Kaggle(2019), Pizza Restaurants Dataset[Online]. Available: https://www.kaggle.com/datasets/datafiniti/pizza-restaurants-and-the-pizza-they-sell
14. Kaggle(2019), Fast Food Restaurants Dataset[Online]. Available: https://www.kaggle.com/datasets/datafiniti/fast-food-restaurants
15. Kaggle(2021), Asian Restaurants Dataset[Online]. Available: https://www.kaggle.com/datasets/jcraggy/asian-restaurants
16. Kaggle(2022), Subway Restaurants Dataset[Online]. Available: https://www.kaggle.com/datasets/timmofeyy/-subway-locations-in-us
17. Kaggle(2018), Michelin Star Restaurants Dataset[Online]. Available: https://www.kaggle.com/datasets/jackywang529/michelin-restaurants
18. Kaggle(2022), Emotions Dataset[Online]. Available: https://www.kaggle.com/datasets/nikhalimaton/food-recommendation-based-emotion?select=EMOTION.csv
19. Kaggle(2022), Ratings Dataset[Online]. Available: https://www.kaggle.com/datasets/schemersays/food-recommendation-system?select=1662574418893344.csv
20. Kaggle(2020), Calories Dataset[Online]. Available: https://www.kaggle.com/datasets/vaishnavivenkatesan/food-and-their-calories
21. Kaggle(2021), Vegan and Vegetarian Restaurants Dataset[Online]. Available: https://www.kaggle.com/datasets/rishidamarla/18000-vegan-vegetarian-restaurants-in-the-us
22. Prof. Hina J. Chokshi, Dr. Ronak Panchal, "Using Apache Jena Fuseki Server for Execution of SPARQL Queries in Job Search Ontology Using Semantic Web", In IJIRCST. 2022. Available: https://www.ijircst.org/DOC/99-using-apache-jena-fuseki-server-for-execution-of-sparql-queries-in-job-search-ontology-using-semantic-technology.pdf