Report

IRE Assignment 2

Name: Pratyush Priyadarshi
Roll No: 2019101118
Batch: UG2k19 CSE

What is Wikidata?

Wikidata is a free and open knowledge base that can be read and edited by both humans and machines. It acts as central storage for the **structured data** of its Wikimedia sister projects, including Wikipedia, Wikivoyage, Wiktionary, Wikisource, and others.

It acts as a common source of open data that Wikimedia projects such as Wikipedia, and anyone else, can use under the CC0 public domain license. It is powered by the set of knowledge graph MediaWiki extension known as Wikibase and MediaWiki Software.

It uses a document-oriented database approach focused on items representing any kind of topic, concept, or object. Each item is allocated a unique, persistent identifier, a positive integer prefixed with the upper-case letter Q, known as a "QID". This enables the basic information required to identify the topic that the item covers to be translated without favouring any language.
Examples of items include the 1988 Summer Olympics (Q8470), Johnny Cash (Q42775), Elvis Presley (Q303) etc.

What is SPARQL?

SPARQL is a recursive acronym for **SPARQL Protocol and RDF Query Language**. It is an RDF query language—a semantic query language for databases—able to retrieve and manipulate data stored in Resource Description Framework (RDF) format.

SPARQL allows users to write queries against what can loosely be called "key-value" data or, more specifically, data that follow the RDF specification of the W3C. Thus, the entire database is a set of "subject-predicate-object" triples. SPARQL allows for a query to consist of triple patterns, conjunctions, disjunctions, and optional patterns.

SPARQL provides a complete set of analytic query operations such as JOIN, SORT, AGGREGATE for data whose schema is intrinsically part of the data rather than requiring a separate schema definition.

A simple example of SPARQL will be:

```
SELECT ?child
WHERE
{
# ?child  father   Bach
  ?child wdt:P22 wd:Q1339.
}
```

Additionally, Wikidata provides a SPARQL endpoint along with a powerful GUI Wikidata Query Service (WDQS). It uses the SPARQL query to search in the Wikidata database and show the result in a table-based format which can easily be imported and used in a code, thereby bringing SPARQL and Wikidata together.

Implementation Methodology

I wrote the Wikipedia page on **Mumbai city in Hindi** (since I live here :p). The page comprises an introduction and two subsections:

1. The first section provides general information about Mumbai, its attractions and tourist spots, and the most spoken languages in Mumbai.
2. The section is dedicated to the Chhatrapati Shivaji Maharaj International Airport, describing its origin name, sea height elevation, IATAS code etc.
3. The third section is focused on the Central Railway station of Mumbai and gives information about the authority in charge of the station, its operator etc.

I queried using SPARQL on wiki data to get the following attributes information:

- General description
- Which region does it belong to
- Latitude and Longitude
- Population
- Time zone
- Area covered by the city
- Airport’s General description
- Airport’s latitude and longitude
- Monuments
- Country in which the city is located
- Main Administrative Authority
- Elevation from sea level
- Variety of language used
- Postal Code
- Airport’s origin name
- Airport’s sea level elevation

- Central authority at the airport
- Social Media follower on Twitter of Airport's page
- Railway's administrative body location
- Railway operator
- Railway's latitude and longitude
- IATA code
- Railway general Description
- Railway's opening date
- Railway main Authority

The code can be divided into three parts:

1. Queries to be used: Contains all the queries used to retrieve the attributes
2. Templates to create the Hindi text: All the templates and rules used to create the desired Hindi sentences are included.
3. Primary generator function: Call all the helper functions to add data in a dictionary and then print Hindi sentences by selecting a given template and attributes that need to be replaced.

Logic

First, I call the get_info function to query the wiki data database using SPARQL to get all the attributes used to create the content.

1. The first query type (index 0) is very generic in nature and is used to find results for specific has relation properties of Mumbai city (airport/station)
2. The second query type (index 1) is used to get a description of the entity I want to describe.
3. The third query was curated to find all museums, famous hotels and architectural structures (attractions and tourist spots) located in Mumbai.

All these three queries are very generic and have a placeholder (@SUBJECT@) which is replaced by the QID of Mumbai (in add_info function) or any other city. This increases the reusability of the code for other cities too.

After the data dictionary is populated, using an English to Hindi translator, I made a template having a variety of strings that have various placeholders (again for reusabiltiy) which get replaced by the data provided as a list to the function create_line(). In the end, I append this line to a variable "text" and print it on the terminal.

The actual formatting of the Wikipedia article is done manually using the sandbox provided by Wikipedia.

Link to the Wikipedia Article

https://en.wikipedia.org/wiki/User:Chief-Blackhood/sandbox