IRE Assignment 2

Creating a Wikipedia Page in Hindi (Report)

Name: Triansh Sharma
Roll No.: 2019101006
Batch: UG3 CSE

What is Wikidata?

Wikidata is a free and open knowledge base that can be read and edited by both humans and machines. It is a collaboratively edited multilingual knowledge graph hosted by the Wikimedia Foundation.

Wikidata acts as central storage for the **structured data** of its Wikimedia sister projects including Wikipedia, Wikivoyage, Wiktionary, Wikisource, and others. It is a common source of open data that Wikimedia projects such as Wikipedia, and anyone else, can use under the CC0 public domain license. Wikidata is a wiki powered by the software MediaWiki and is also powered by the set of knowledge graph MediaWiki extensions known as Wikibase.

What is SPARQL?

SPARQL is an RDF query language, i.e. a semantic query language for databases that can retrieve and manipulate data stored in Resource Description Framework (RDF) format. SPARQL is a recursive acronym that stands for **SPARQL Protocol and RDF Query Language.**

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 World Wide Web Consortium (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.

Wikidata provides a SPARQL endpoint including a powerful Web-GUI Wikidata Query Service (WDQS) since September 2015. It runs a SPARQL query against Wikidata's dataset and shows you the result, bringing SPARQL and Wikidata together.

Example of a general query

Wikidata assigns each item and property a unique identifier. To search for an item, we find the Q-number for the item and use it to query about it. Similarly, for properties, we use "P:search term" instead of just "search term", which limits the search to properties. For simple WDQS triples, items should be prefixed with `wd:`, and properties with `wdt:`.

The queries are of the form subject-predicate-object. A general query can be expressed as the following:

```
SELECT ?child ?childLabel
WHERE
{
# Amitabh_Bachchan has_child   ?child
  wd:Q9570 wdt:P22 ?child.
  SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE]". }
}
```

This query shows a list of all children of Amitabh Bachchan.

Implementation Methodology

I wrote the Wikipedia page on **Amitabh Bachchan** in **Hindi**. The page is divided into 3 sections.

1. The first section gives an introduction to Amitabh Bachchan, his profession and education.
2. The second section is about the Bachchan family, which describes who all is a part of the Bachchan family. It covers Amitabh Bachchan's parents, wife, and children.
3. The third section gives an overview of his Political and film career and notable awards and achievements he gained.

I queried the Wikidata of Amitabh Bachchan to get the information about the following attributes:

- Description
- Alias (also known as)
- Occupation
- Net worth
- Social Media followers
- Education
- Native language
- Country of citizenship
- Birthdate and birthplace
- Father and mother name
- children name
- Religion
- Residence
- Start of the work period
- Number of films in which he casted
- Some examples of his films
- Positions held by him
- Awards received
- His notable achievements
- Image link

The code is divided into 3 files:

1. **templates**: Contains all templates(or rules) used to create sentences.
2. **queries**: Contains SPARQL queries used to retrieve data from Wikidata.
3. **main**: The main file to generate text using rules and data obtained by querying Wikidata.

For each attribute, there is a rule which defines how the text needs to written in a generalized manner. This rule contains either one or two placeholder marks written by ___ (triple underscore). The generalized manner means that if we try to write the page for some other film actor (preferrably male),

only values in the placeholder and the name of the person will change, if for that rule, the attribute value exists in wikidata.

Similarly, the queries file defines all the queries used. Some queries result in multiple output values, hence querying multiple attributes at the same time is avoided and hence the code may take up some time to query all the data. There are a total of 8 different queries described in the file. They are:

1. Queries description from subject's schema
2. Queries subject's alias
3. A general query to return the object value for the given subject
4. Since, place of birth and date of birth use a single template, hence these 2 attributes are queried using a single query.
5. The next query sums over the social media followers of the subject
6. The next query count the number of films in which the subject was a cast member (film has cast member subject)
7. We also provide the examples of films. This query is the most **complex** query among all. This uses **bind**, **filter** and **limit** to get 5 most popular films of the subject.
8. We also query the link to the image of subject.

Procedure

The following procedure was used.

1. Define queries for each attribute. One query was made reusable.
2. Define rules for all the attributes we are querying for the subject.
3. For each attribute, we query subject's wikidata using SPARQL query. We format a list of outputs to text and then pass a rule where the placeholder is replaced with the output text. We append this entry in a list
4. For all attributes, ending in a section, we join the sentences with a space and we repeat from step 3 to get data for the next section.
5. At the end, all strings are concatenated.

Note: For above queries, in my case, Amitabh Bachchan is the subject.

Link to sandbox

The following is the link: https://en.wikipedia.org/wiki/User:Triansh/sandbox