ONTOLOGY SUPPORTED WEB SEARCH

Team :
S2

Team Information :
Shalini Hemachandran sxh163230
Sivagurunathan Velayutham sxv176330

Type of Project :
Custom Project

CONTENTS

1. INTRODUCTION

Search Engines like Google, Bing and Yahoo provide users with autocomplete search suggestions based on a partially entered search queries. These suggestions are generated based on the following.

1. Users' previous search histories
2. What other people are searching for, including Trending stories
3. The terms being typed.

The drawback of this approach is that the suggested completions are often ambiguous in nature. They do not provide any support for disambiguating homonyms i.e term with two or more meaning for the search term.

Our goal in this project is to provide better search suggestion by disambiguating homonyms and providing semantically relevant information with the help of building an ontology supported web search (OSWS). The project has been inspired from the work of the reference research paper called Google Knows Who Is Famous Today [1] in which an ontology based web search is constructed using DBPedia and YAGO data sources. A big shortcoming of the project developed in the research paper is the use of DBpedia and YAGO data sources which may be outdated since these data sources are updated only once in two years. Another shortcoming is that the YAGO and DBPedia end points have a very slow response times.

The main idea of our project is to address the shortcoming of the project developed in the research paper by using a new data source called **Babelnet** - which provides very fast API response times and contains latest up to date information.

2. TARGET AUDIENCE

Providing semantically relevant and disambiguated search to everyone. Specifically targeting the **web-users** who want relevant search suggestions without ambiguous homonym information.

3. DESCRIPTION OF DATA SOURCE

The dataset for the project has been derived from three primary sources. They are

1. US Census Data - US Census Data contains latest demographic information which includes names of people

2. Google Search Suggestions API - Given a text, the API returns the most popular search suggestions for the text.
3. Babelnet - Babelnet is a multilingual lexicalized semantic network was automatically created by linking Wikipedia to the most popular computational lexicon of the English language, WordNet.

4. DATA INTEGRATION

In this project we build an ontology supported web search for famous personalities with semantically relevant, disambiguated search suggestions.Since a concrete dataset is not available for the project, dataset had to be generated from scratch from the data sources mentioned in the previous section. Data had to be collected over a period of two days owing to the restrictions on Babelnet's endpoint hit count. The data thus collected was then preprocessed and persisted in an RDF knowledge base

4.1 Data Collection

The following are the steps involved in data collection and generation.

1. Top 1000 popular first names was procured from US census data.
2. For each first name, a Google Search Suggestions API call was given to fetch possible last name suggestions that pop up.
3. For each name (first name-last name pair), Babelnet's SPARQL endpoint was queried to verify if the name refers to a famous personality.
4. If the name refers to a famous personality, then additional details about the famous personality was retrieved using Babelnet's java api.
5. The data collected from the above steps was then persisted as json files.

4.2 Data Pre Processing

The json files, one for each famous personality was parsed and loaded into memory for preprocessing. Preprocessing involves the following steps.

1. The data of famous people was reorganized based on their types. Babelnet provides a clear IS A relationship which
2. Useful relationships were retained using a probabilistic threshold criterion.

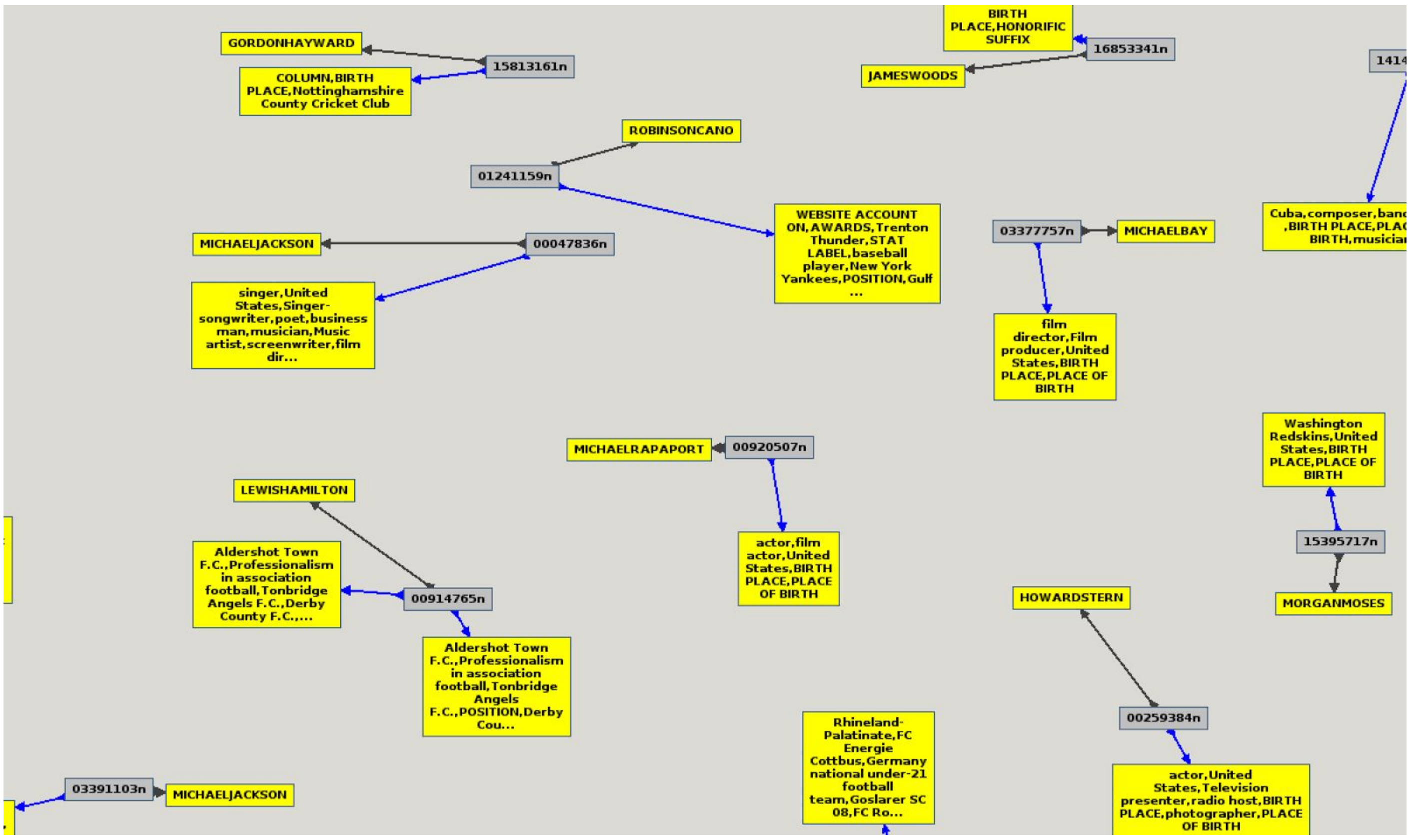
$$\frac{\textit{Number of people in a given class with a particular relationship}}{\textit{Total number of people in a given class}} < p$$

   where p was taken as 0.5. Thus only those relationships which are relevant with respect to a particular class are retained.
3. Redundant relationships were removed.

4.3 Conversion to RDF Knowledge Base

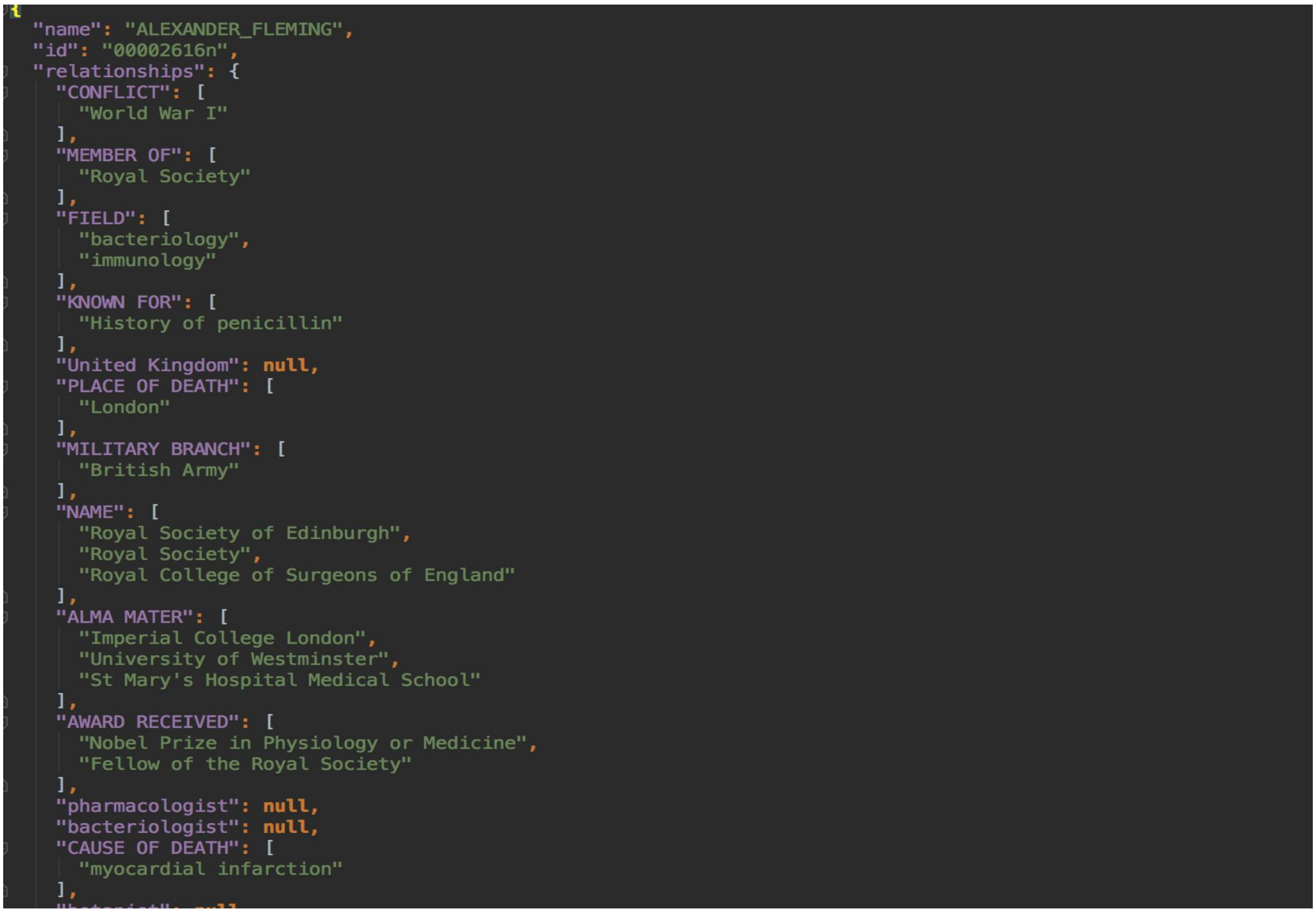
After preprocessing, every famous person is converted into an RDF resource using the id generated by Babelnet for each famous person. The name of each person is added as an FOAF:name property. The possible search suggestions for each person is added as an FOAF:depiction property. The converted dataset is then stored in a TDB datastore and also written as an RDF/XML.

A snapshot of the RDF knowledge base loaded into Gruff is shown in the figure given below.



GORDONHAYWARD
15813161n
COLUMN,BIRTH PLACE,Nottinghamshire County Cricket Club
BIRTH PLACE,HONORIFIC SUFFIX
16853341n
JAMESWOODS
ROBINSONCANO
01241159n
WEBSITE ACCOUNT ON,AWARDS,Trenton Thunder,STAT LABEL,baseball player,New York Yankees,POSITION,Gulf ...
MICHAELJACKSON
00047836n
singer,United States,Singer-songwriter,poet,business man,musician,Music artist,screenwriter,film dir...
03377757n
MICHAELBAY
film director,Film producer,United States,BIRTH PLACE,PLACE OF BIRTH
Cuba,composer,band ,BIRTH PLACE,PLAC BIRTH,musician
MICHAELRAPAPORT
00920507n
actor,film actor,United States,BIRTH PLACE,PLACE OF BIRTH
LEWISHAMILTON
Aldershot Town F.C.,Professionalism in association football,Tonbridge Angels F.C.,Derby County F.C.,...
00914765n
Aldershot Town F.C.,Professionalism in association football,Tonbridge Angels F.C.,POSITION,Derby Cou...
Washington Redskins,United States,BIRTH PLACE,PLACE OF BIRTH
15395717n
MORGANMOSES
HOWARDSTERN
00259384n
actor,United States,Television presenter,radio host,BIRTH PLACE,photographer,PLACE OF BIRTH
Rhineland-Palatinate,FC Energie Cottbus,Germany national under-21 football team,Goslarer SC 08,FC Ro...
03391103n
MICHAELJACKSON

Fig:4.1 A snapshot of knowledge base loaded in Gruff



```
{
  "name": "ALEXANDER_FLEMING",
  "id": "00002616n",
  "relationships": {
    "CONFLICT": [
      "World War I"
    ],
    "MEMBER OF": [
      "Royal Society"
    ],
    "FIELD": [
      "bacteriology",
      "immunology"
    ],
    "KNOWN FOR": [
      "History of penicillin"
    ],
    "United Kingdom": null,
    "PLACE OF DEATH": [
      "London"
    ],
    "MILITARY BRANCH": [
      "British Army"
    ],
    "NAME": [
      "Royal Society of Edinburgh",
      "Royal Society",
      "Royal College of Surgeons of England"
    ],
    "ALMA MATER": [
      "Imperial College London",
      "University of Westminster",
      "St Mary's Hospital Medical School"
    ],
    "AWARD RECEIVED": [
      "Nobel Prize in Physiology or Medicine",
      "Fellow of the Royal Society"
    ],
    "pharmacologist": null,
    "bacteriologist": null,
    "CAUSE OF DEATH": [
      "myocardial infarction"
    ],
```

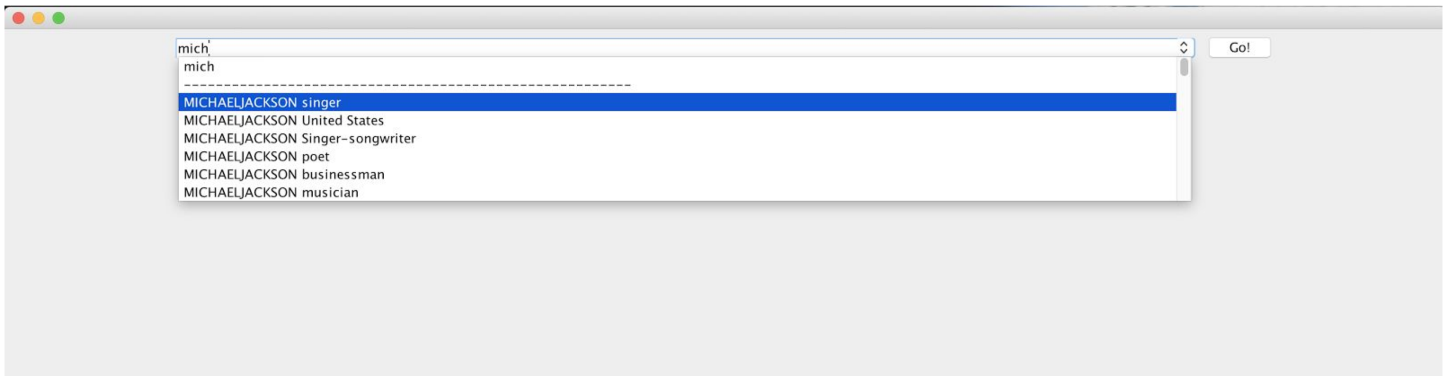
Fig:4.2 A snapshot of data collected and persisted as JSON

5. DATA PRODUCT RESULTS :

A java Swing stand alone application was built using which users will be able to search for famous personalities with semantically relevant and disambiguated search suggestions.

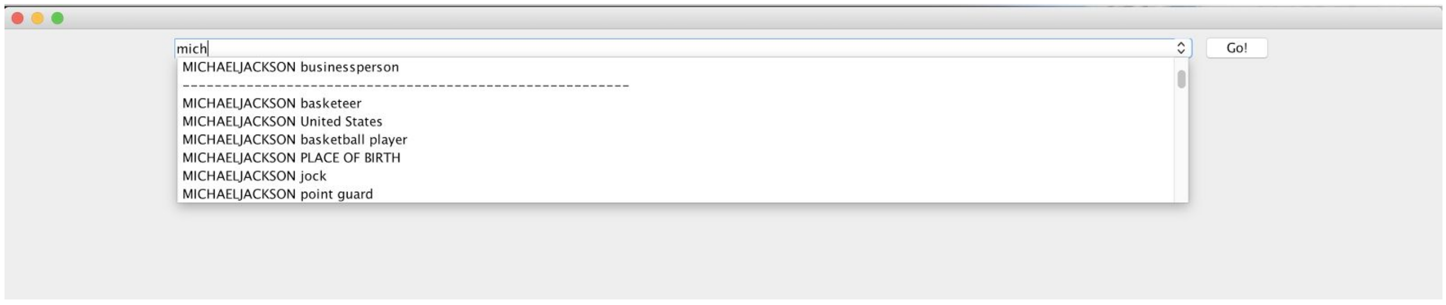
As and when the user keeps entering texts in the text field, a **SPARQL** query is initiated to retrieve suggestions from the RDF knowledge base.

The following are the screenshots of the application.


mich
mich

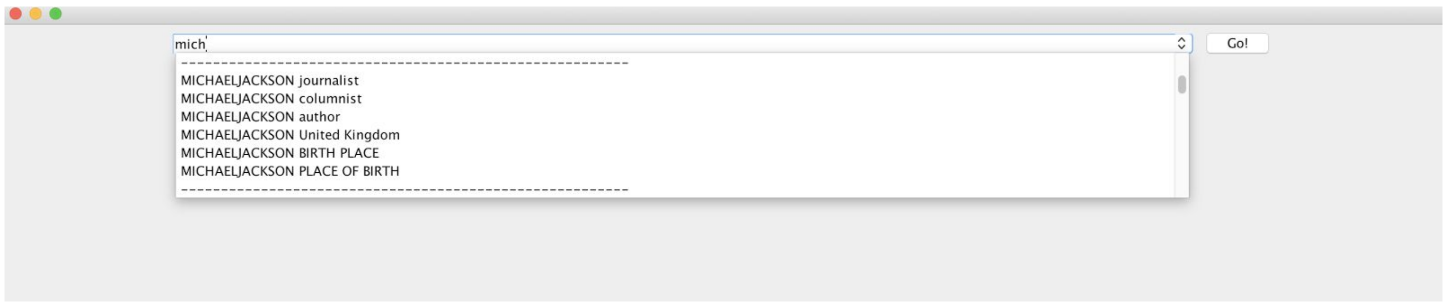
MICHAELJACKSON singer
MICHAELJACKSON United States
MICHAELJACKSON Singer-songwriter
MICHAELJACKSON poet
MICHAELJACKSON businessman
MICHAELJACKSON musician

Fig:5.1 Search Suggestions for the text "mich"
The figure above shows the search suggestions which pop up when the user types a text in the search box. The suggestions are loaded from the RDF database. The dotted line separation serves to disambiguate homonyms.


mich
MICHAELJACKSON businessperson

MICHAELJACKSON basketeer
MICHAELJACKSON United States
MICHAELJACKSON basketball player
MICHAELJACKSON PLACE OF BIRTH
MICHAELJACKSON jock
MICHAELJACKSON point guard

Fig:5.2 Disambiguated homonyms


mich

MICHAELJACKSON journalist
MICHAELJACKSON columnist
MICHAELJACKSON author
MICHAELJACKSON United Kingdom
MICHAELJACKSON BIRTH PLACE
MICHAELJACKSON PLACE OF BIRTH

Fig:5.3 Disambiguated homonyms

Figures 5.2 and 5.3 show disambiguations for the name Michael Jackson using line separation.

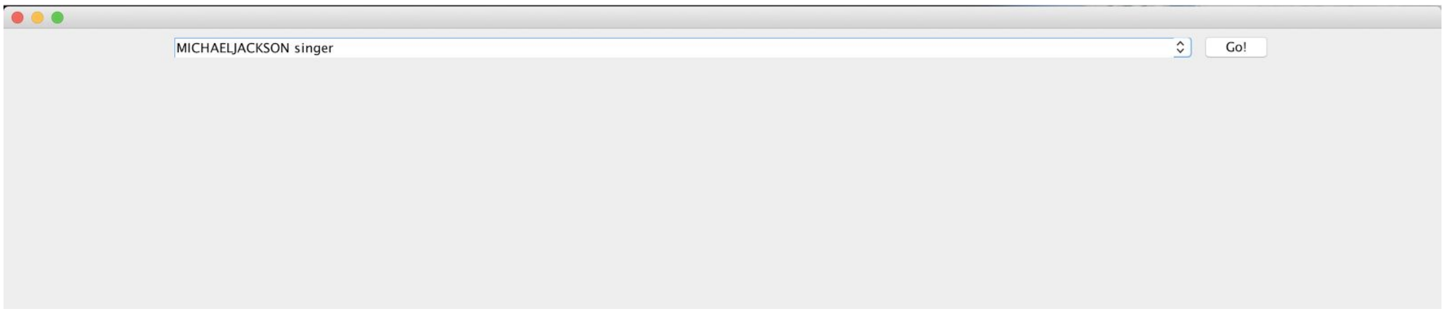
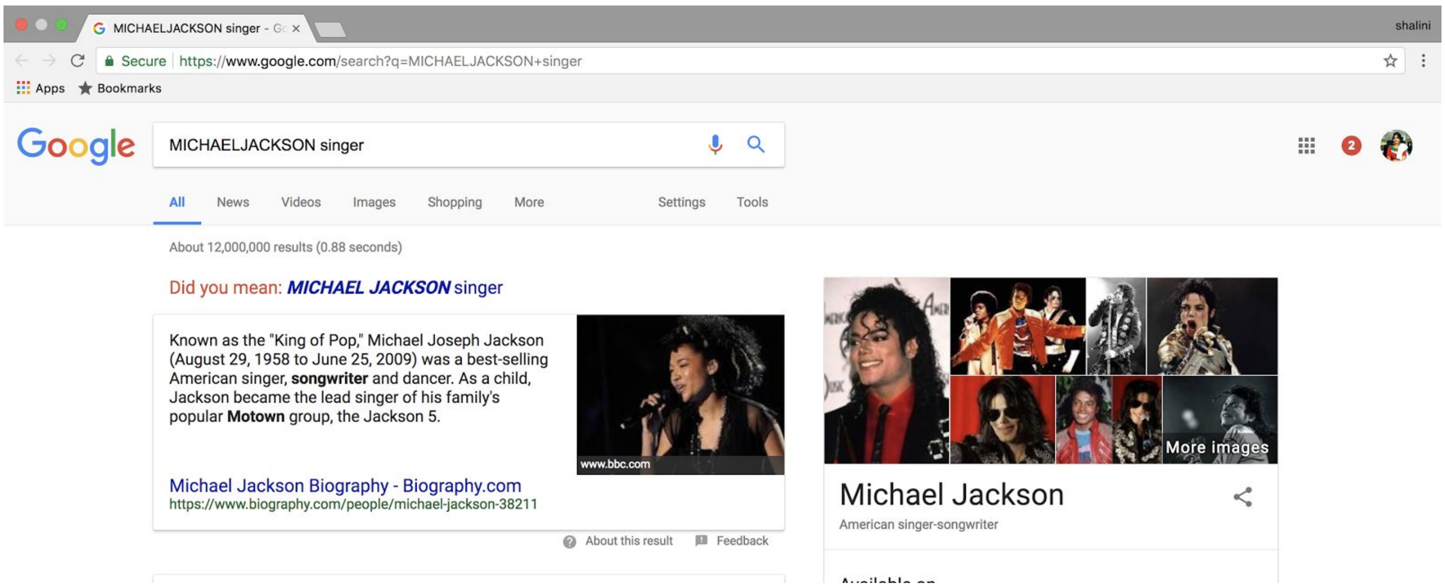


Fig:5.4 Selection from Search Suggestions
Once a selection is made from the search suggestions, the Go button can be clicked.


MICHAELJACKSON singer - G ×

shalini

Secure | https://www.google.com/search?q=MICHAELJACKSON+singer

Apps Bookmarks

MICHAELJACKSON singer

All News Videos Images Shopping More Settings Tools

About 12,000,000 results (0.88 seconds)

Did you mean: ***MICHAEL JACKSON*** singer

Known as the "King of Pop," Michael Joseph Jackson (August 29, 1958 to June 25, 2009) was a best-selling American singer, **songwriter** and dancer. As a child, Jackson became the lead singer of his family's popular **Motown** group, the Jackson 5.

www.bbc.com

Michael Jackson Biography - Biography.com
https://www.biography.com/people/michael-jackson-38211

About this result Feedback

More images

Michael Jackson
American singer-songwriter

Fig:5.5 Go Button Action
Once the Go button is clicked, the browser opens with a google search tab with the selected search query

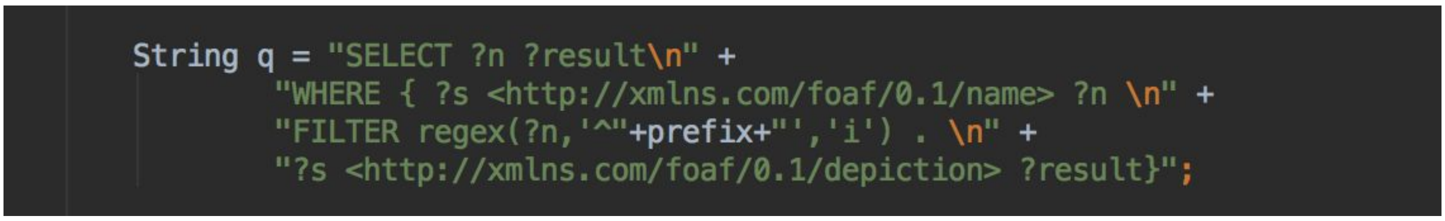

```
String q = "SELECT ?n ?result\n" +
        "WHERE { ?s <http://xmlns.com/foaf/0.1/name> ?n \n" +
        "FILTER regex(?n,'^"+prefix+"','i') . \n" +
        "?s <http://xmlns.com/foaf/0.1/depiction> ?result}";
```

Fig:5.6 SPARQL Query
The figure above shows the SPARQL Query which gets executed every time something is entered in the search text field.

6. CUSTOM PROJECT JUSTIFICATION

In the research paper, **Google Knows Who Is Famous Today**, which is used as the base of this project, DBPedia and YAGO were used as data sources. The disadvantage with both the data sources is that they are updated once in two years. Therefore there is a possibility that the data from these data sources might be inconsistent and not up to date. The API response times of both the data sources have been observed to be very slow.

In order to overcome data inconsistencies and slow response times, we have used a new datasource called Babelnet which is similar to WordNet. Babelnet's endpoint is faster and it is rich in data content compared to DBPedia and YAGO put together. BabelNet groups words in different languages into sets of synonyms, called *Babel synsets*. It contains almost 14 million synsets and about 746 million word senses in the latest version (3.7). Also, Babelnet automatically updates its data store every time a change happens in wikipedia datastores thus resolving the major problem of data inconsistency.

7. SUMMARY

An Ontology supported web search for famous people was built with semantically relevant, disambiguated search suggestions by collecting data from three different data sources namely US Census Data, Google's Search Suggestion API and Babelnet, followed by data preprocessing and finally conversion to an RDF knowledge base. Every time a search query is entered in the search text field, a SPARQL query was issued to retrieve the relevant search suggestions. Future work of this project would be to extend the ontology to include more famous people and to dynamically expand the ontology as and when a query is issued.