Question Answer System for US Election 2016

Authors:
Ganesh Taduri - gt784@mail.umkc.edu
Swatvik Gunamaneni - sgf3f@mail.umkc.edu
Tejo Kumar Manchu - tmcx4@mail.umkc.edu
Venkata Sasidhar kanumuri - vk6fb@mail.umkc.edu

Abstract:

There are many advantages in extracting the information from the slew of sources available on the internet. Not a decade ago, we realized that we can process this large amount of data and extract meaningful information from it. There is a growing importance on the development of information technologies that are capable of accessing and analyzing the data available via social media. Twitter is one such source of large amount of raw data is which contains worldwide news, public opinion on several events which can be processed and relevant information can be extracted. On average, every second around 6000 tweets are tweeted on twitter. Along with this there is static data available in knowledge bases like Wikipedia. The mixture of static and dynamic data is used to a build a question answer system. In this paper we explain the detailed structure and working of the question answer system developed on US Election 2016.

Introduction

Siri, the artificial intelligence personal assistant on Apple Inc. phones and Watson, a question answering computer machine developed by IBM motivated us to select the domain of question answering system. We selected the topic of 'USA elections 2016' to develop our system. Our main objective of this project is to provide a short answer for a given question that is related to US 2016 elections such as "*Who is President nominee from republic party*". The input question is given in natural language. We extract the answer from an ontology that is developed based on the data related to the elections using SPARQL. On the whole we use natural language processing techniques and some of the machine learning algorithms. Most of the existing question and answering system available online namely AQUA, ASKMSR etc lack the implementation of machine learning algorithms. In this paper, we attempt to give comprehensive details of our system.

Proposed solution

In the first place, we present the architecture of our question and answering system.

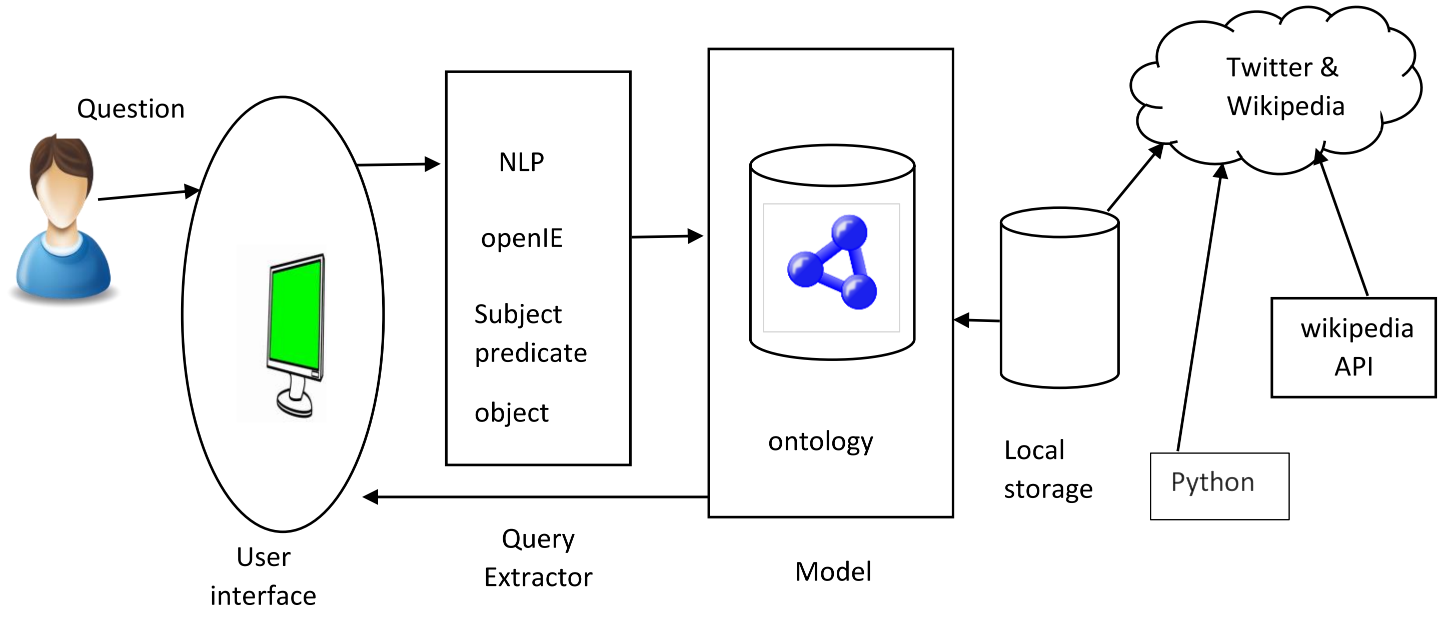


Fig.i System Architecture

The above diagram represents architectural diagram for question and answering system. The user gives his question in natural language to the user interface. The question is then processed using natural language process techniques like lemmatization, POS tagging etc. After processing we will find out he subject, predicate and object. Using these values, we construct the query in SPARQL and using this query, we extract the related answer from the Ontology. The Ontology is developed using the data collected from Twitter and Wikipedia. The extracted answer is then displayed. This is the basic working of our system. In the subsequent sections, we present and discuss each of the process in detail.

Implementation: we use various technologies to build our system handling different kinds of files.

Tools and Technologies used: Apache Spark frame work for Hadoop, Python, Java, Scala, IntelliJ IDE, Protege.

The entire system can be viewed as involving four subparts

i) Data collection
ii) Ontology creation
iii) Question classification
iv) Query Generation and Answer Extraction

Data collection:

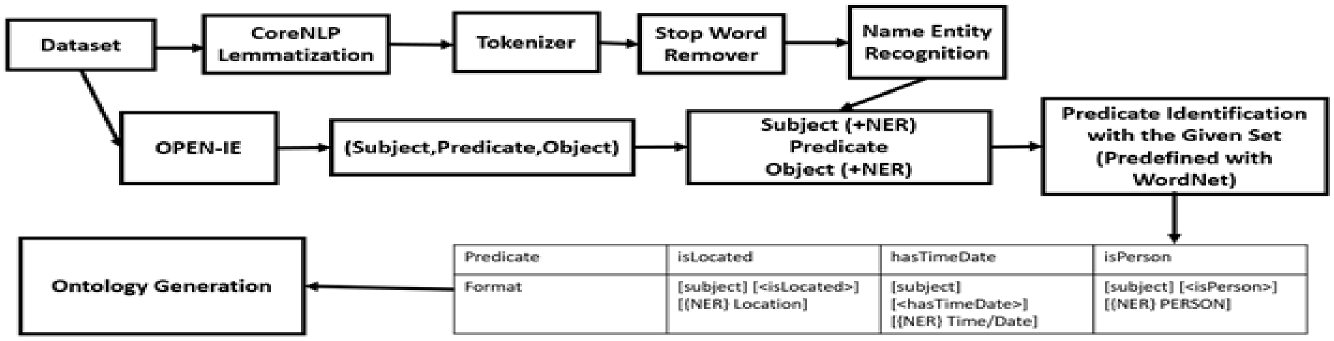
Our data is related to US election 2016.We collected data from various sources like Wikipedia and twitter using python and their respective API. The twitter data is collected based on #election2016.The Wikipedia data is a text file but the twitter data is extracted in the form of JSON object. We extracted the tweet text from the object. The data is distributed among several documents depending upon its content. The extracted data from Wikipedia is a meaningful one whereas most of data from twitter is inconsistent with various errors like spelling and grammatical mistakes and unrelated data. We ran stop word removal (NLP technique) on the twitter data to make it meaningful. We collected data from Wikipedia pages like Donald Trump, Obama, Hillary Clinton etc. using Wikipedia API for python.
Data collection URL: www.twitter.com & www.wikipedia.com

Ontology Generation:

After extracting the data, we generated the Ontology as per the workflow shown in Fig.ii. The data set is the data collected in the previous phase. The ontology generated using the complete data contains more than seven thousand triples and this file is too large to be visualized in webVOWL and process to query extraction. Hence, we used part of the data to generate the ontology. On this data we perform core NLP Lemmatization which gives the root word of every word in the data. The tokenizer gives words from the sentences. The stop word remover removes the irrelevant words contained in the data set. The name entity recognizer will classify the words into names of person, place, company etc.

Parallelly we perform OPEN IE on the data which gives the relationship between the arguments and we then find the subject, predicate, object using NER. we then compare the predicates from NER and OPEN-IE and identify the predicates using mapping technique.

To create the ontology, we pruned our data to maximum extent to make the ontology meaningful. We developed two ontologies dynamically using twitter and Wikipedia data. Of the two, ontology based on Wikipedia data is more meaningful. This is shown in Fig.iii. Even though the ontology is meaningful it is not useful enough to extract sensible data. Hence we developed a static ontology based on part of the collected data set. This ontology is shown in fig. iv.


Dataset
CoreNLP Lemmatization
Tokenizer
Stop Word Remover
Name Entity Recognition
OPEN-IE
(Subject,Predicate,Object)
Subject (+NER) Predicate Object (+NER)
Predicate Identification with the Given Set (Predefined with WordNet)
Ontology Generation

Predicate	isLocated	hasTimeDate	isPerson
Format	[subject] [<isLocated>] [(NER) Location]	[subject] [<hasTimeDate>] [(NER) Time/Date]	[subject] [<isPerson>] [(NER) PERSON]

Fig.ii Workflow of Ontology Generation

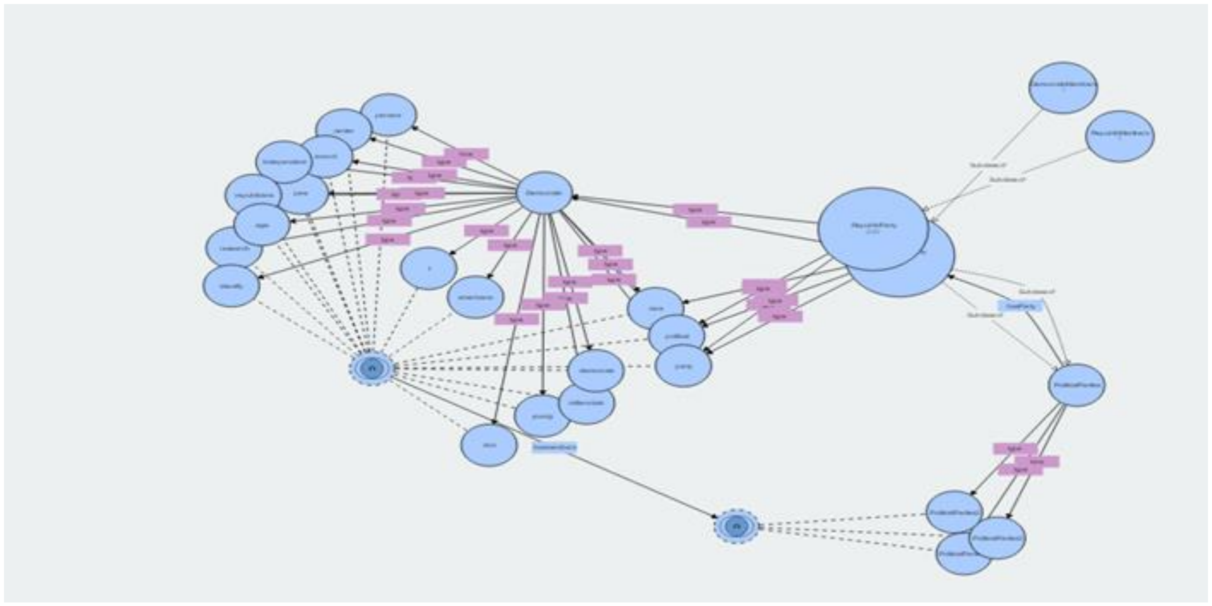


Fig.iii Dynamic Ontology developed from Wikipedia data

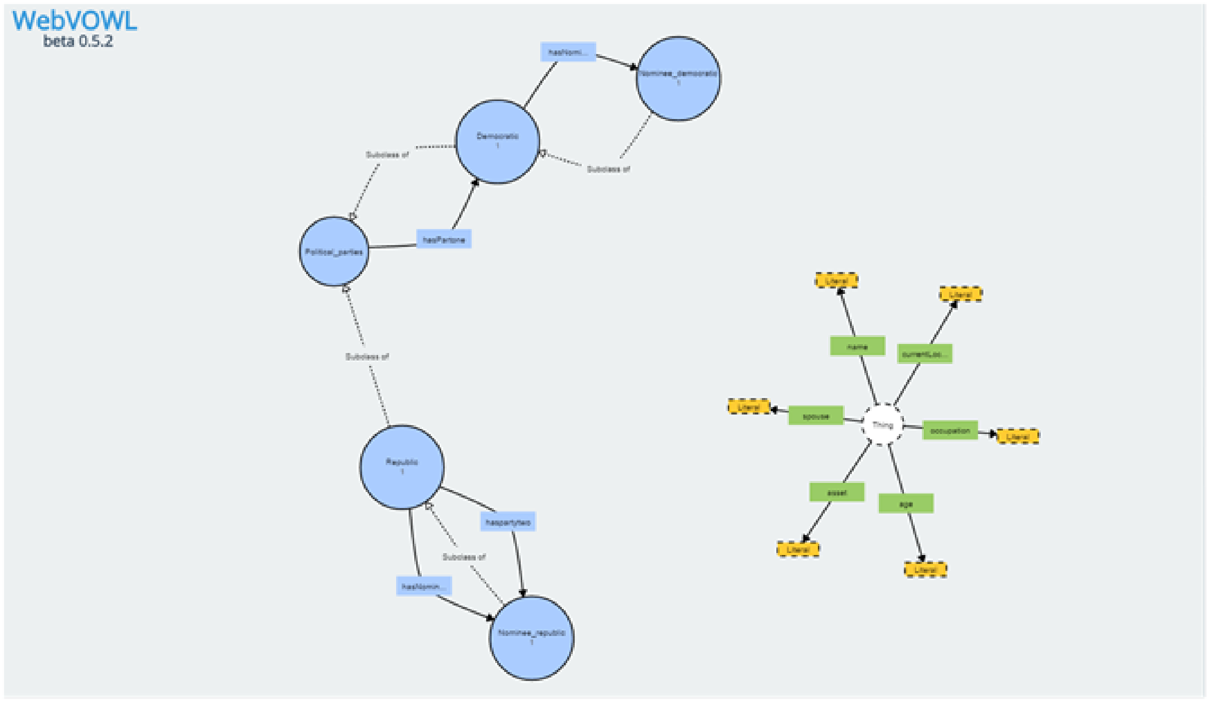
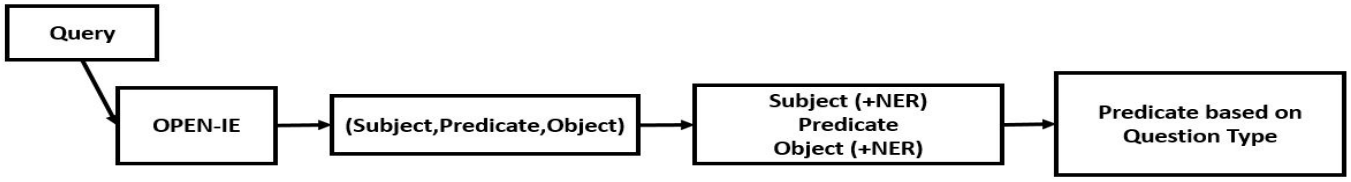


Fig.iv Static ontology developed from Wikipedia

<u>Question Classification:</u>

In this phase, the system processes the input question given by user in natural language. The workflow is shown in fig.v. During this process we find out the subject, object and predicated of the question using POS tagging technique. In POS tagging a noun in the question is identified as subject and object. The first encountered noun is considered as the subject while the other one is treated as the object. The verb obtained during the process is considered as predicate. The identification of subject, predicate and object is the key part of this phase. We also use WORDNET technique to map the noun and the question categories and achieve question classification. Following are some of the examples:

who, whom - the category of the answer is person.
when - the category of the answer is date.
why - the category of the answer should be a reason.
where - the category of the answer is location



Figv. Workflow for question classification

Query Generation and Answer Extraction:

The input question is fed to the Open IE module and then NER technique is performed on the question. The workflow is shown in Fig.vi.Therefore, from the input question we identify the subject, predicate, object. Using these values, we generate SPARQL query. The query is then fed as input to the ontology from which we extract the respective answer.

Example:
Input Question: Who is Republic Nominee?
SPARQL query:
PREFIX elec: http://www.kdm.com/OWL/elections2016#

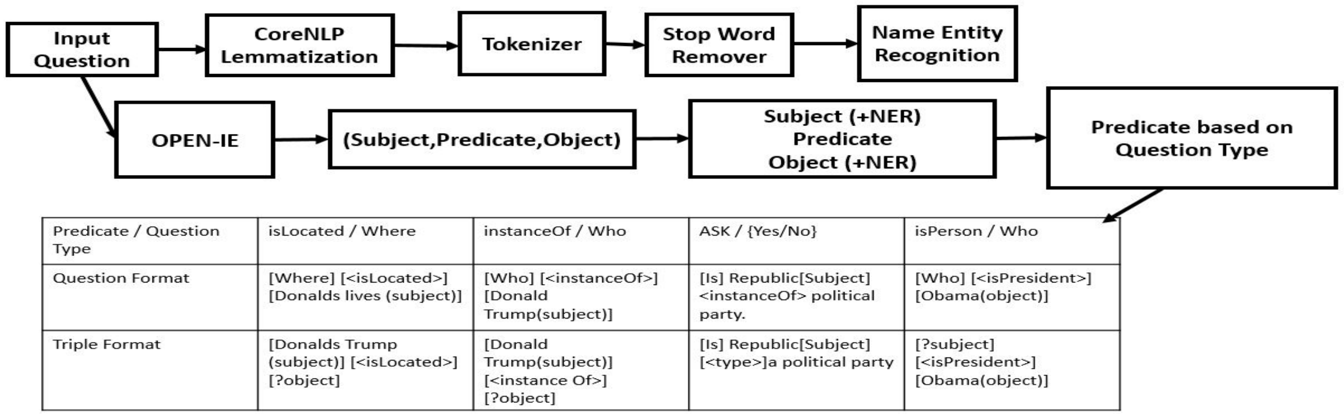
SELECT? name

WHERE {

    Elec: Republic Nominee Elec: name? name

}

Answer: Donald Trump



Predicate / Question Type	isLocated / Where	instanceOf / Who	ASK / {Yes/No}	isPerson / Who
Question Format	[Where] [<isLocated>] [Donalds lives (subject)]	[Who] [<instanceOf>] [Donald Trump(subject)]	[Is] Republic[Subject] <instanceOf> political party.	[Who] [<isPresident>] [Obama(object)]
Triple Format	[Donalds Trump (subject)] [<isLocated>] [?object]	[Donald Trump(subject)] [<instance Of>] [?object]	[Is] Republic[Subject] [<type>]a political party	[?subject] [<isPresident>] [Obama(object)]

Fig.vi. Workflow for query extraction

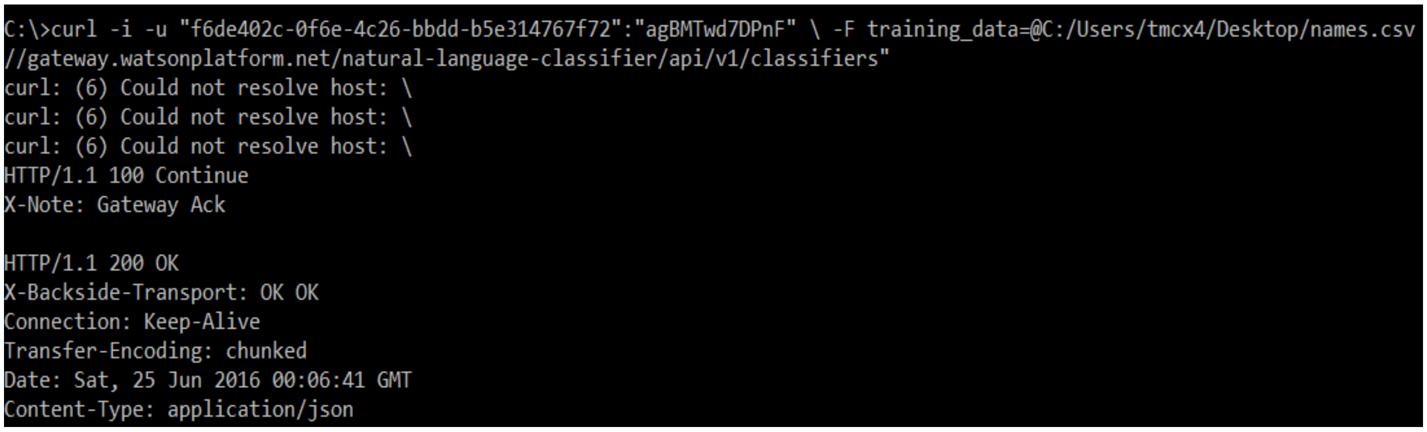
Project GitHub Link: https://github.com/ganeshtaduri/KDM

YouTube demo URL: https://www.youtube.com/watch?v=CLRLTchjkoY

Related Work: In addition to the modules developed for the system we implemented natural language classifier service from alchemy API which implements complex machine learning algorithm. In this section, the demo of the same classifier is shown. In this demo we train the classifier with twenty-five male names and twenty-five female names. Then we give a test name which the classifier takes as input and gives the percentage confidence of male and female classification.

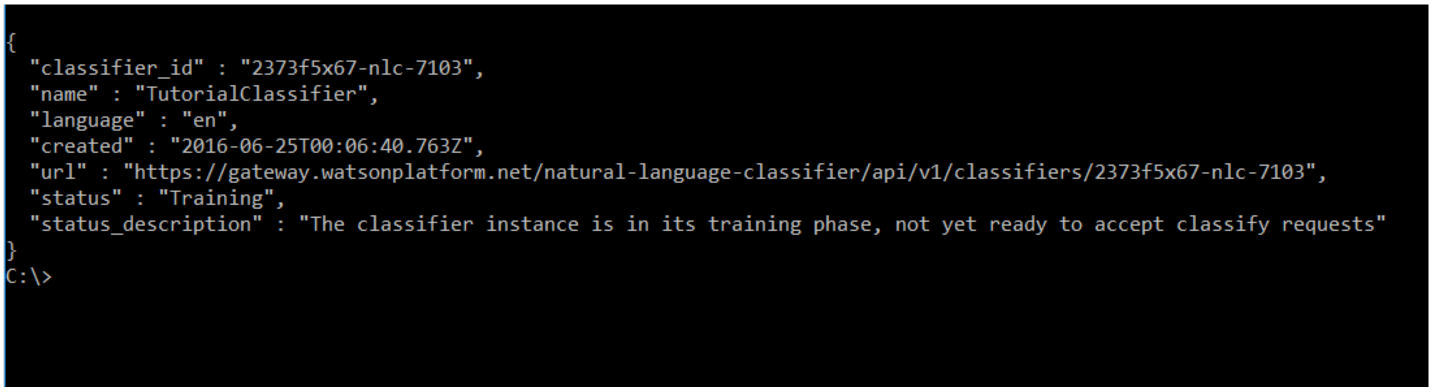
With respect to our project, we plan to train this system to achieve question classification with a set of different kinds of questions and then when a question in natural language is given to this system, the classifier detects the question classification

Screen shot of giving training data:



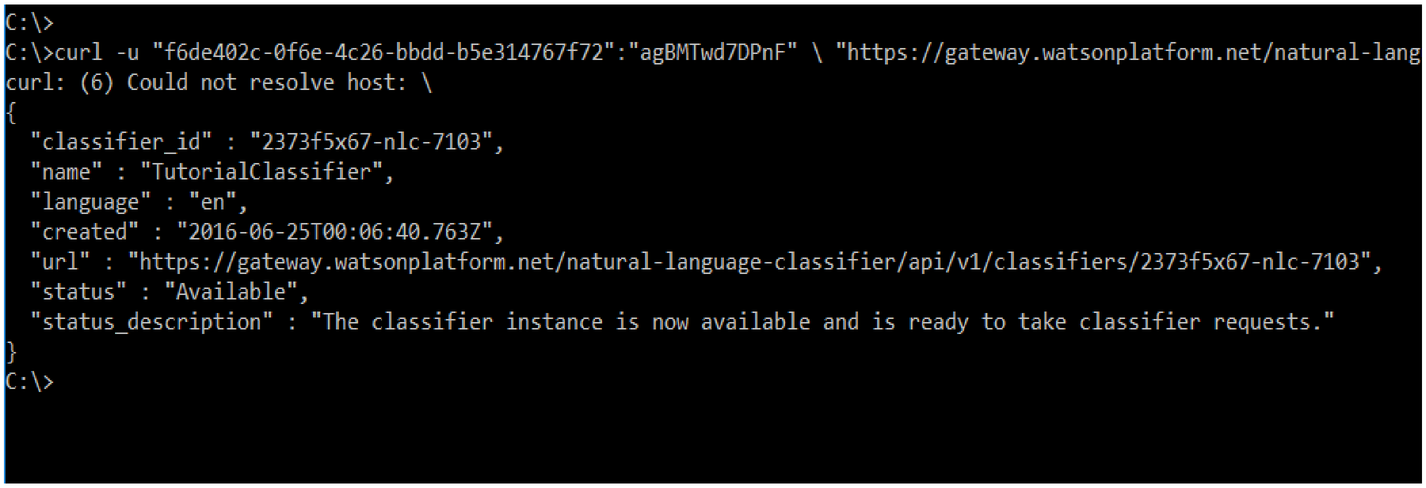
```
C:\>curl -i -u "f6de402c-0f6e-4c26-bbdd-b5e314767f72":"agBMTwd7DPnF" \ -F training_data=@C:/Users/tmcx4/Desktop/names.csv
//gateway.watsonplatform.net/natural-language-classifier/api/v1/classifiers"
curl: (6) Could not resolve host: \
curl: (6) Could not resolve host: \
curl: (6) Could not resolve host: \
HTTP/1.1 100 Continue
X-Note: Gateway Ack

HTTP/1.1 200 OK
X-Backside-Transport: OK OK
Connection: Keep-Alive
Transfer-Encoding: chunked
Date: Sat, 25 Jun 2016 00:06:41 GMT
Content-Type: application/json
```

Classifier response:



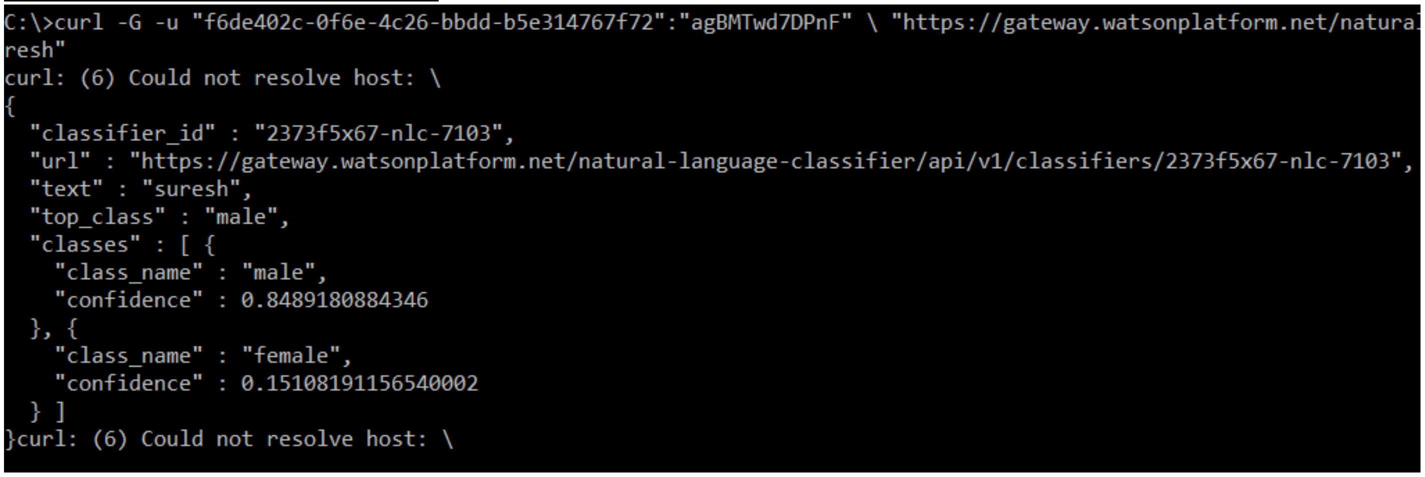
```
{
  "classifier_id" : "2373f5x67-nlc-7103",
  "name" : "TutorialClassifier",
  "language" : "en",
  "created" : "2016-06-25T00:06:40.763Z",
  "url" : "https://gateway.watsonplatform.net/natural-language-classifier/api/v1/classifiers/2373f5x67-nlc-7103",
  "status" : "Training",
  "status_description" : "The classifier instance is in its training phase, not yet ready to accept classify requests"
}
C:\>
```

<u>Classifier Status:</u>



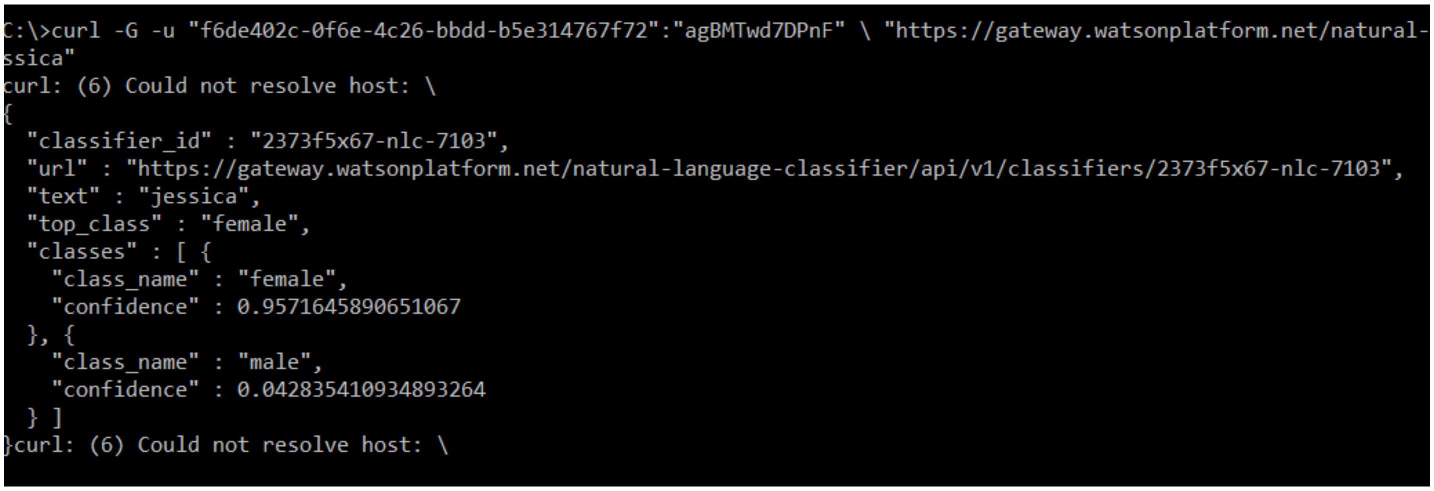
```
C:\>
C:\>curl -u "f6de402c-0f6e-4c26-bbdd-b5e314767f72":"agBMTwd7DPnF" \ "https://gateway.watsonplatform.net/natural-lang
curl: (6) Could not resolve host: \
{
  "classifier_id" : "2373f5x67-nlc-7103",
  "name" : "TutorialClassifier",
  "language" : "en",
  "created" : "2016-06-25T00:06:40.763Z",
  "url" : "https://gateway.watsonplatform.net/natural-language-classifier/api/v1/classifiers/2373f5x67-nlc-7103",
  "status" : "Available",
  "status_description" : "The classifier instance is now available and is ready to take classifier requests."
}
C:\>
```

<u>Classifier Output for test data:</u> The name “Suresh” is classified as male with 85% confidence.



```
C:\>curl -G -u "f6de402c-0f6e-4c26-bbdd-b5e314767f72":"agBMTwd7DPnF" \ "https://gateway.watsonplatform.net/natural
resh"
curl: (6) Could not resolve host: \
{
  "classifier_id" : "2373f5x67-nlc-7103",
  "url" : "https://gateway.watsonplatform.net/natural-language-classifier/api/v1/classifiers/2373f5x67-nlc-7103",
  "text" : "suresh",
  "top_class" : "male",
  "classes" : [ {
    "class_name" : "male",
    "confidence" : 0.8489180884346
  }, {
    "class_name" : "female",
    "confidence" : 0.15108191156540002
  } ]
}curl: (6) Could not resolve host: \
```

The name “Jessica” is classified as female with 95% confidence.



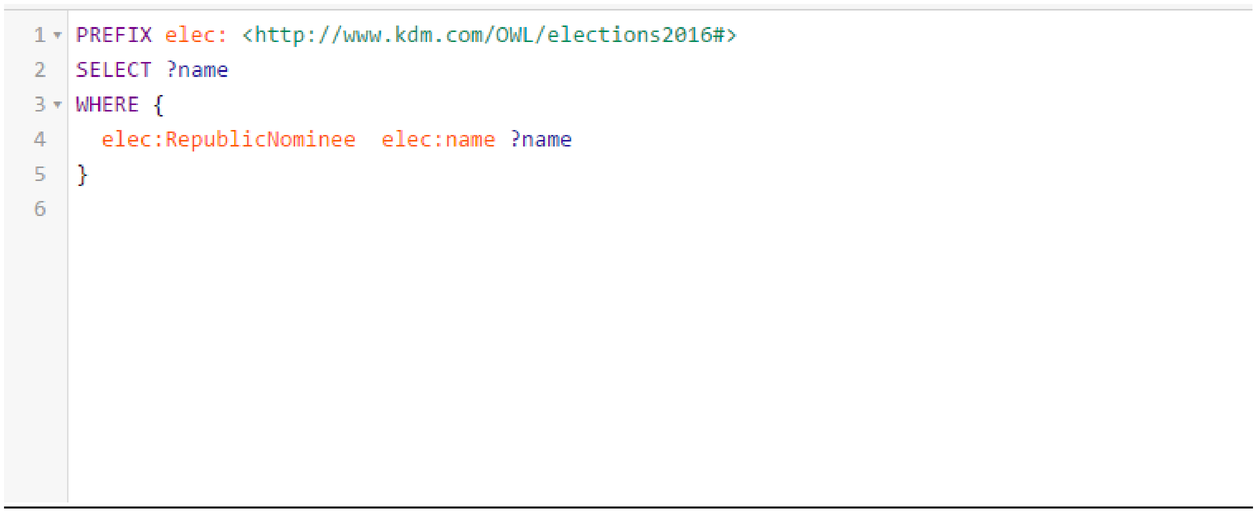
```
C:\>curl -G -u "f6de402c-0f6e-4c26-bbdd-b5e314767f72":"agBMTwd7DPnF" \ "https://gateway.watsonplatform.net/natural-
ssica"
curl: (6) Could not resolve host: \
{
  "classifier_id" : "2373f5x67-nlc-7103",
  "url" : "https://gateway.watsonplatform.net/natural-language-classifier/api/v1/classifiers/2373f5x67-nlc-7103",
  "text" : "jessica",
  "top_class" : "female",
  "classes" : [ {
    "class_name" : "female",
    "confidence" : 0.9571645890651067
  }, {
    "class_name" : "male",
    "confidence" : 0.042835410934893264
  } ]
}curl: (6) Could not resolve host: \
```

Results and Evaluation: We tested our system on the statically developed Ontology by using protege tool and we obtained the results

Question: Who is Republic Nominee?

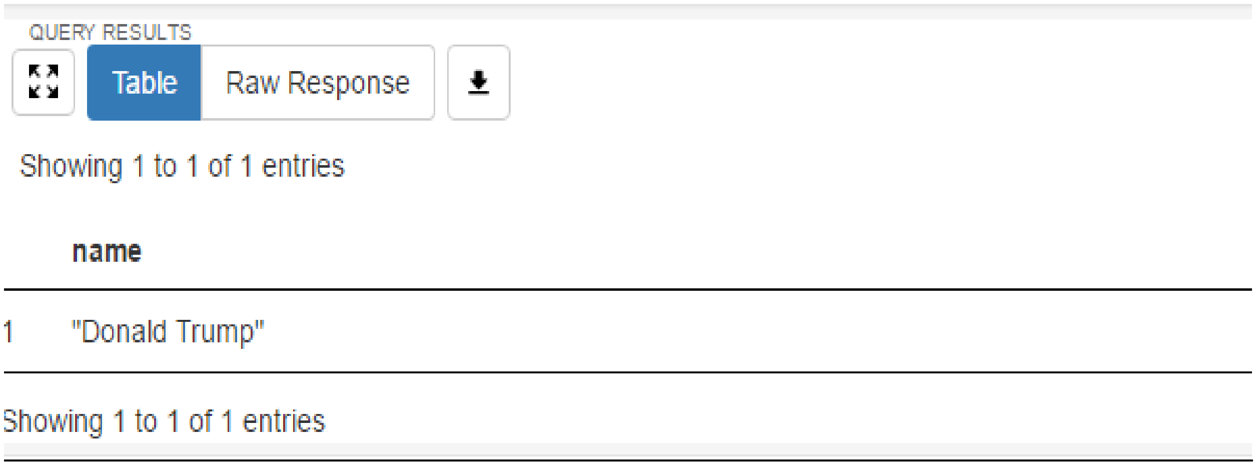
Answer: Donald Trump

Converted Query(Screenshot)



```
PREFIX elec: <http://www.kdm.com/OWL/elections2016#>
SELECT ?name
WHERE {
  elec:RepublicNominee  elec:name ?name
}
```

Answer Screenshot:

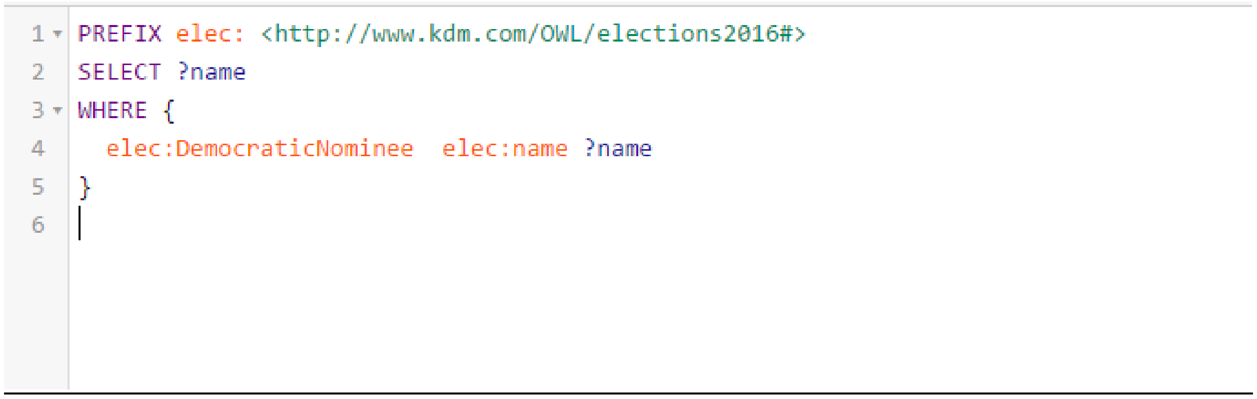


```
Showing 1 to 1 of 1 entries

     name
1    "Donald Trump"

Showing 1 to 1 of 1 entries
```

Question: Who is Democratic Nominee?

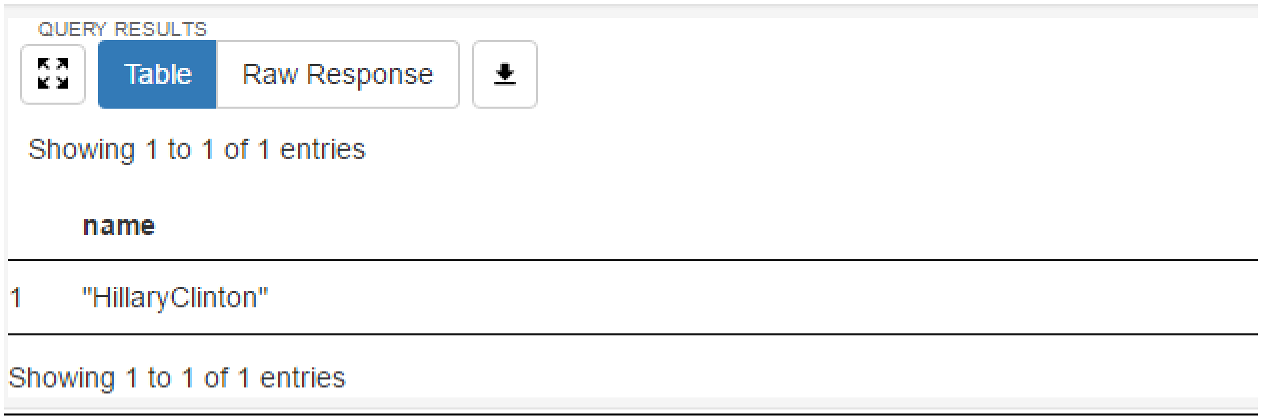
Answer: Hillary Clinton

Converted Query(Screenshot)



```
PREFIX elec: <http://www.kdm.com/OWL/elections2016#>
SELECT ?name
WHERE {
  elec:DemocraticNominee  elec:name ?name
}
```

Answer Screenshot



Showing 1 to 1 of 1 entries

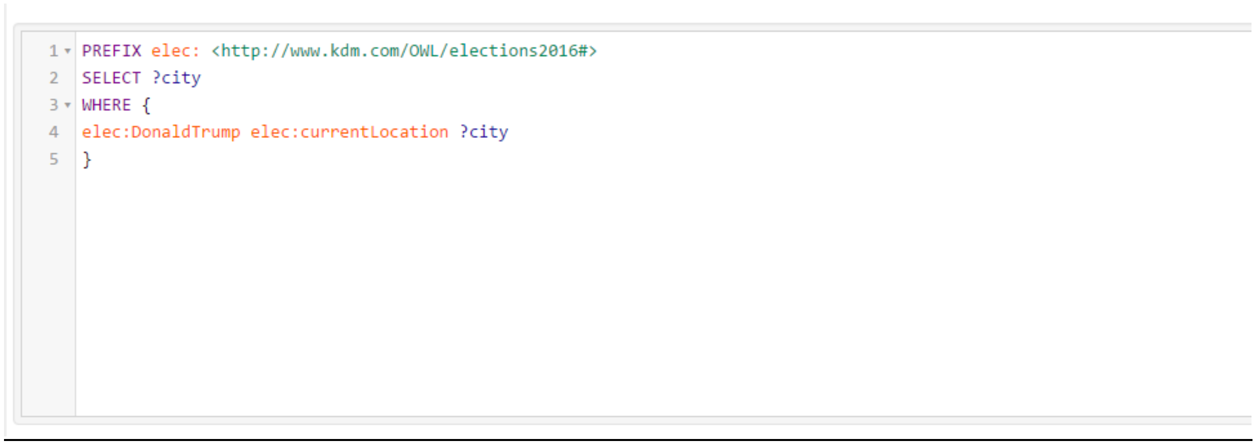
	name
1	"HillaryClinton"

Showing 1 to 1 of 1 entries

Question: Where is DonaldTrump?

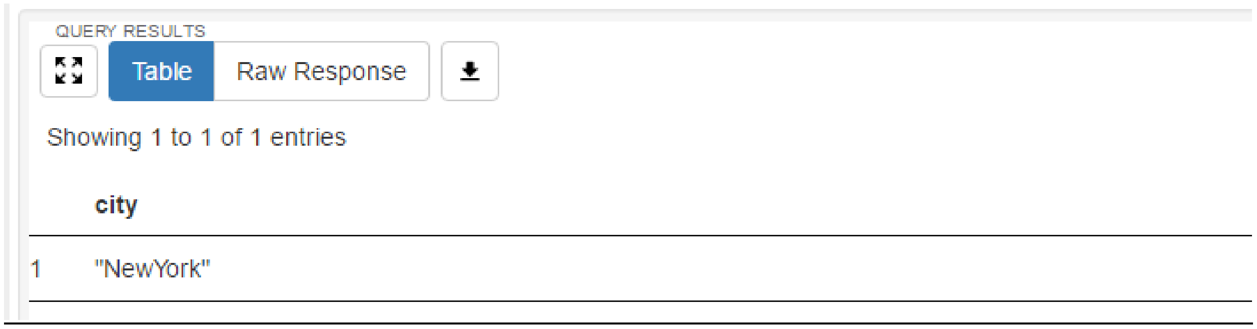
Answer: New York

Converted Query(Screenshot)



```
PREFIX elec: <http://www.kdm.com/OWL/elections2016#>
SELECT ?city
WHERE {
elec:DonaldTrump elec:currentLocation ?city
}
```

Answer Screenshot



Showing 1 to 1 of 1 entries

	city
1	"NewYork"

Future Work: Although we developed the question answering system's user interface using Bootstrap, the integration with backend was a challenge. Data extraction volume can be increased if we use DBpedia framework to extract the data from the infoboxes of Wikipedia in one shot while preserving the structure. The data from the Twitter is discarded as it is not making any sense. The pruning of the data can be improved to tap the public opinion on the question and hence we can display the public opinion along with the answer. The Ontology is scalable and hence can be increased based on the requirements. The question classification phase can be improved by training the model using machine learning algorithms to achieve better classification of question.

Conclusion: The objective of the project is to learn machine learning algorithms, natural language processing through implementation of available services and APIs like Natural language classifier etc in the question answering system. We successfully completed the construction of a working prototype of the question answer system. There is a remarkable amount of learning accomplished during this process. We thoroughly analyzed each module of the project in detail while developing the system.

References:

Lewis, Burn L. "In the game: The interface between Watson and Jeopardy! ."IBM Journal of Research and Development 56.3.4 (2012): 17-1.

Lally, Adam, et al. "Question analysis: How Watson reads a clue." IBM Journal of Research and Development 56.3.4 (2012): 2-1.

Lehmann, Jens, et al. "DBpedia–a large-scale, multilingual knowledge base extracted from Wikipedia." Semantic Web 6.2 (2015): 167-195.

West, Robert, et al. "Knowledge base completion via search-based question answering." Proceedings of the 23rd international conference on World wide web. International World Wide Web Conferences Steering Committee, 2014.

https://www.aaai.org/Papers/Symposia/Spring/2003/SS-03-07/SS03-07-009.pdf