

CSE 535: Project Report

Bazinga

Question Answering

Aniruddh Chaturvedi	[5020 6958]
Arnav Ahire	[5020 8006]
Ashwin Nikam	[5020 7368]
Kedar Paranjape	[5020 5932]

Contents:

- Overview
- Key Features
- Implementation Details
- Team
- Sample Tests
- Resources

Overview

Question Answering is an inter-disciplinary field which combines concepts of Information Retrieval and Natural Language Processing. This project mainly focuses on developing a system for effective question answering between the user and the system. The motive for such a system is to find answers from the available data. The system which has been developed is mainly effective for answering questions of the type Who/ When/ Where etc. The questions have to be queried against a dataset of tweets which are indexed using Solr. The tweets have been extracted using the hash-tag (#Demonetization). The final goal of this project is to find tweets which are most relevant to the user's query and able to answer the user's question in an acceptable way. The project includes concepts like named entity recognition, entity extraction, parts of speech tagging and natural language processing.

Key Features

The key features of the project include the following:

Natural Language Processing: This project involved the use of Natural Language Processing. The Apache OpenNLP library has been used for this purpose of processing natural language text. Part-of-speech tagging is the NLP task which has been implemented in this project.


Named Entity Recognition: The purpose of Named Entity Recognition is to locate and classify named entities which are present in the text into predefined categories such as persons, organizations, locations etc. Google's Cloud Natural Language API has been used for this purpose.

Implementation Details

- The main purpose of this project is to provide answers to questions of the type Who/ Where etc.
- A large volume of tweets was extracted using the hash-tags #Demonetization and #Demonetisation in JSON format.
- These tweets then have to be indexed using Solr so that queries can be fired against them.
- The user query is entered into the system using the User Interface (UI).
- The implementation of this project consists of two main parts which is:
 - Query processing
 - Results processing

1) Query processing

- The query which is entered into the UI first undergoes part-of-speech tagging which is done using OpenNLP library. Since we are finding answers for Who/ What questions, it would be better if the 'Wh-word' was excluded from the query along with some common stopwords. This is done basically to reduce the number of irrelevant tweets which would otherwise be extracted because of them.
- Eg: query: Who has suffered due to demonetization?



```
"responseHeader":{
  "status":0,
  "QTime":6,
  "params":{
    "q":"Who has suffered due to demonetization?",
    "indent":"on",
    "fl":"tweet_text",
    "wt":"json"}},
"response":{"numFound":18710,"start":0,"docs":[
  {
    "tweet_text":["The plank on which @BJP4India should hav been attacked is how #demonetization has not impacted it or its benefactors while poor hav suffered"]},
  {
    "tweet_text":["@SuPriyoBabul do you know anyone who has died due to cash crunch/ #demonetization? No wonder your lack of empathy for poor. sushmitadevmp"]},
  {
```

- In the above example, the answer which was expected was a person or an organization entity however a tweet was returned which had the words 'due to' matched to the query.
- For this reason, we need to exclude such words while querying. The POS taggers are used to classify each word as a noun, verb, adverb, adjective etc. We only extract the

nouns (including proper nouns and common nouns), verbs(excluding stopwords like is/ was/ have), adjectives and adverbs. We have also given a boost of 5 to nouns so

- that tweets in which the nouns are present are given much more preference.
- The above query would be processed in order to give: suffered demonetization
- Querying on this processed result would give much better results which are shown below:

```
← → ↻ ⓘ localhost:8983/solr/core1/select?fl=tweet_text&indent=on&q=suffered%20demonetization&wt=json ☆ ⋮

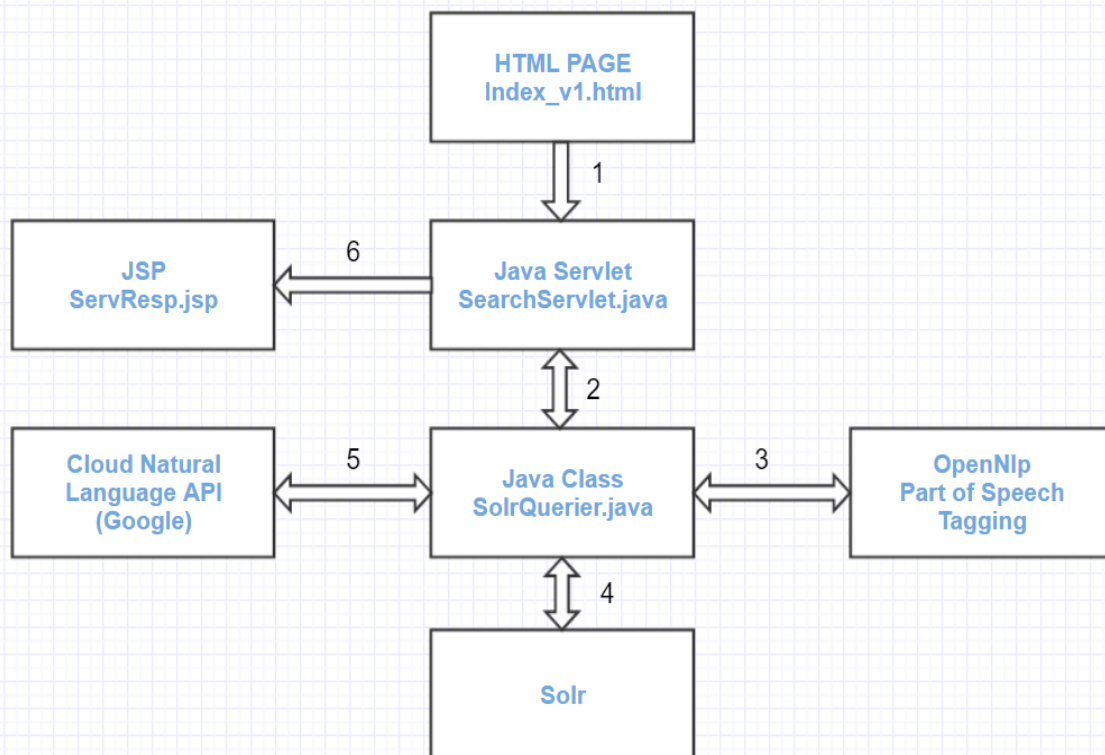
{
  "responseHeader":{
    "status":0,
    "QTime":2,
    "params":{
      "q":"suffered demonetization",
      "indent":"on",
      "fl":"tweet_text",
      "wt":"json"}},
  "response":{"numFound":42935,"start":0,"docs":[
    {
      "tweet_text":["I've suffered the harassment, while supporting PM @narendramodi for #DeMonetisation. But realized, that I won't get...
https://t.co/LUCxK0UJT6"]},
    {
      "tweet_text":["Ppl suffered heart attack but most of them were not in Q, they suffer d jolt worrying bout their #blackmoney erosio...
https://t.co/bIZRmC3Qp0"]},
    {
      "tweet_text":["Anecdotal evidence: Business, trading community suffered massive loss of hoarded #cash. Change curve very steep for them.
\n#Demonetization"]},
```

2) Result processing

- The processed query is actually fired into Solr to get the results. Among the results, the top 50 tweets are extracted which are the most relevant. This relevance to the query is being decided using the BM25Similarity model which has been implemented in the Solr core.
- After the top 50 tweets are extracted, they are being processed using Google's Cloud Natural Language API. The reason why the CNLAPI is being used on only 50 tweets is latency. Querying and retrieving 1000 tweets from Solr is much faster than performing entity recognition and extraction on 1000 tweets using CNLAPI.
- Depending on the query fired, the CNLAPI will look for different entities in the tweets.
- For example, if the query is a 'who' type query we are mainly looking for a person or an organization. Hence, only the top tweets in which the person or organization entity is present, are returned back to the user.
- Similarly, if a 'Where' query has been fired and the same 50 tweets have been extracted as most relevant from Solr, then CNLAPI would look for the 'Location' entity and return the top 20 tweets to the user.
- Thus depending on the query CNLAPI gives different results for the same data.
- If the question is a 'When' type question we need to find the 'time' and 'date' entities. The CNLAPI can't extract time and date entities hence we use the OpenNLP library for this purpose. The models used in OpenNLP are Time name finder model and Date name finder model. We can also train these models to give better results.

- If the question is a 'When' type question the date name finder model and the time name finder model are used against all the tweets extracted by Solr instead of the 50 tweets. This is mainly because CNLAPI takes up significant amount of time to process all the Solr results however OpenNLP can do the same task faster and with efficiency.

Implementation Flow



Implementation Flow

Team and Contributions

- **Aniruddh Chaturvedi**
 - Front End Web Development
- **Arnav Ahire**
 - Front End Web Development
- **Ashwin Nikam**
 - Back End NLP and Entity Recognition
- **Kedar Paranjape**
 - Back End NLP and Entity Recognition

Sample Tests

BAZINGA SEARCH

When it's Demonetization Info that you are looking for.

Query: who suffered badly due to demonitization

This is what the Twitteratis have to say :

1

@VillageSoda Hello Madam, What is the situation in your village due to #Demonetization We are suffering badly in Delhi

2

I've suffered the harassment, while supporting PM @narendramodi for #DeMonetisation. But realized, that I won't get? <https://t.co/LUCxKOUJT6>

3

Anecdotal evidence: Business, trading community suffered massive loss of hoarded #cash. Change curve very steep for them. #Demonetization

4

The plank on which @BJP4India should hav been attacked is how #demonetization has not impacted it or its benefactors while poor hav suffered

5

@Rohinisgh_ET well I see none of the target met by #demonetization only common ppl suffered and hence that'll surely be reflected in voting

In above sample, we are passing “who” type query.

Query: who suffered badly due to demonetization.

Here, we try to find tweets that contain proper nouns (person entity).

BAZINGA SEARCH

When it's Demonetization Info that you are looking for.

Query: where did people burn modi's effigy

This is what the Twitteratis have to say :

1

Ludhiana: People burn effigy of NaMo to protest against #DeMonetization <https://t.co/1K371Puwe9>

2

RT @BspUp2017: People burn effigy of Narendra Modi in Ludhiana to protest against #Demonetization, Day 25 #BSP #MayawatiNextUPCM #YoModiSoF?

3

Ludhiana: People burn effigy of NaMo to protest against #ModiSurgicalStrikeonCommonMan #DeMonetization #HeartOfAsia <https://t.co/sf3S4NI4Nn>

4

RT @BspUp2017: People burn effigy of Narendra Modi in Patiala to protest against #Demonetization, Day 28. #BSP #MayawatiNextUPCM #Ambedkar?

5

RT @rajeshpatel1278: Day 28People burn effigy of Narendra Modi in Patiala to protest against #Demonetization <https://t.co/paZQhuuuuX>

In above sample, we are passing “where” type query.

Query: where did people burn modi's effigy.

Here, we focus on relevant tweets that contain location entities.

BAZINGA SEARCH
When it's Demonetization Info that
you are looking for.

Query: when was demonetization declared
This is what the Twitteratis have to say :

- 1 Indians asked for #Lokpal to end corruption during UPA rule! Modi came & after 29 months declared #DeMonetization and issued Rs. 2000 note.
- 2 Indians asked for #Lokpal to end corruption during UPA rule! Modi came & after 29 months declared #DeMonetization and issued Rs. 2000 note.?
- 3 RT @BspUp2017: Must watch: Over 100 deaths have been reported since #demonetization declared on 8 Nov. #MayawatiNextUPCM #BSP Courtesy The?
- 4 RT @AmritaDhawan1: @ajaymaken ji apprising public about ill planned #Demonetisation.Declared #NotePeCharcha 2b organised on 13th &14th Dec?
- 5 Adhar opposed GST opposed Parliament disruption was democratic #DeMonetisation was anti poor, soon after May 2017 <https://t.co/UXW7WJtGld>

In above sample, we are passing “when” type query.
Query: when was demonetization declared.
Here, we focus on tweets that contain a time/date entity

Resources

- <https://lucidworks.com/blog/2013/02/11/a-simple-question-answering-system-using-solr-and-opennlp/>
- <http://opennlp.sourceforge.net/models-1.5/>
- <https://cloud.google.com/natural-language/>
- <https://web.stanford.edu/class/cs124/lec/ga.pdf>