

# Knowledge Discovery and Management

## Project Report



## Tech Champs

### Team 1

1. Jakkepalli, Rama Charan Pavan (8)
2. Puthana, Sujitha (24)
3. Yalamanchili, Sowmya (30)
4. Nandanamudi, Sreelakshmi (17)

# **1.Project Motivation, Objectives, and Significance**

## **1.1 Motivation:**

Data Science is a system to extract knowledge of data in various forms, either structured or unstructured from various domains, like Knowledge Discovery in Databases(KDD). Natural language processing is used for processing the text which is machine understandable and which will help for fast retrieval of data. Ontology plays an important role with respect to entity classification to answer questions. Visualization of the data by classification into classes, subclasses, data properties, object properties using the concept of ontology tool protégé. Protégé tool has its unique features for fetching the related information by using either spark SQL or DL query, which are the simplified query to fetch instances.

## **1.2 Specific Objectives:**

### **1.2.1 Easy search of information from huge amount of text.**

In present days, the amount of data is increasing and this is leading to the difficulties in handling the data. So, we need the machine learning algorithms to handle these huge data. We are making use of artificial intelligence algorithm for machine learning to handle data and search the data.

### **1.2.2 Helps in precise answer for customized questions.**

As we are using these AI algorithms for handling data, this helps in getting through different algorithms available including TFIDF, NLP algorithms, word2vec algorithm, kmean algorithm, classification of data using all these processes and analyzing the accuracy. This tremendous process leads to precise answer of the question.

### **1.2.3 Increase the knowledge management process.**

In the process of going through different AI algorithm to classify data and handle them. We could understand the importance of each algorithm with the specified uniqueness. By making use of all these algorithms simplify the management of data.

### **1.2.4 Visualization of the data.**

Human are more prone to understand the visualized data than the text data. Visualization includes the presentation of the data in the form of knowledge graph. The text data is

classified into classes, subclasses, properties are extracted. We generate an owl class and give it as input to protégé and visualize it using either plugin VOWL or webvowl.

WebVowl: <http://visualdataweb.de/webvowl/>

### **1.2.5 Simple query to fetch information.**

Spark sql is like normal sql commands that can be used to fetch information in the form of schema. In our application, we are using the spark sql commands to answer some questions. Protégé tool has its own query language DL query which is more simplified version of querying. DL query fetched the instances of the classes which can answer few questions.

## **1.3 Specific Significance:**

This application helps in fetching the answer to questions by using NLP Process, word2vec, TF-IDF, N-gram. NLP, kmean, Classification of data, NLP algorithm is useful step for text processing and then we are extracting the relevant data. Visualization of the knowledge graph is also of great use.

However, all the algorithm we are using in the project have its own significance. Comparing all these processes to find the best process with respect to time, accuracy, cost to select the best process.

Query using the spark sql or DL sql can fetch the information from the entity classified. These queries are very fast to extract the information to answer the relevant questions.

## **2. Domain and Q/A application**

We are taking News as our domain for our project and applying NLP operations on it and further applying question answering system for the dataset. For this question answering system, we are considering two datasets from News domain.

Question and answer application is build where a user can ask question like what, why, who, when related to the domain dataset. Then the algorithm is implemented to search the huge data. With very high processing speed and high accuracy, we will fetch the precise answer and display it to the user.

### **3. Related Work:**

In the present days, where the data is huge leading to data management issues. There are many algorithms already existing but the main problem in the existing algorithms are completeness and correctness. To solve this problem, we need to consider all these algorithm and judge wisely which all are the algorithms that we can use to easily maintain data and give us the high accuracy. But a single algorithms or approach cannot solve this problem. Hence, we should integrate multiple algorithm for high accuracy in designing the search engine.

Searching the huge amount of data is very difficult. Knowledge Graph represents the graphical representation of the entities and interrelated relationship. There is different knowledge graph available in the market but googles knowledge graph is the popular search engine algorithm. Best knowledge graph can be designed solving the completeness and correctness issue by integrating different approaches of knowledge graph available in the market.

Data sources that are available to us are limited. We can increase the accuracy to provide the best answer to any question is by considering all the data sources that are available on the web. The solution for this approach is the knowledge vault that was made available to us by google that takes the data in RDD triplets i.e., subject, object, predicate. After collecting the data and finding the entities our next problem would be organizing the data. We Deep Dive approach helps in resolving the problem of extraction of data and its integration to fetch accurate prediction making the training process easy.

After the data is represented in RDF triplets, the semantic relationship can be organized using the FehSen to merge the related information leading to more simplified data. It is known fact that structured data is easy to handle than unstructured data. Fonduer is the approach in focusing the

construction of the structured data from the plain text. By using all these approach helps in improving the handling the data and solve the “completeness and correctness” problem.

Optimization of the questions is important to get high accuracy. Latent dirichet allocation is used to extract the topics. Applying the LDA algorithm on the question is used to cluster the question topic, measuring the similarity based on semantic between multiple questions. OpenIE algorithm is also applied on the questions to generate the RDF triplets to understand the question.

Visualization of the data plays a keys role in understanding and process huge data. Visualization is done by extracting the key entities and relationship between them. Object properties defines the property relationship between two instances. Data properties defines the relationship between two entities. Modern algorithm “Concept Net” which is an improved version to visualize the data using the labels and edges.

In our world where there exists data in multiple languages. In order to achieve the high accuracy information, we need to consider data from all the available sources in all the languages. DBpedia algorithm is the best approach for this process. After completing the data extraction, data retrieval our main task is to improve the processing time and accuracy to fetch the most relevant answer to the question. One good approach is the query to fetch the relevant answer. Spark query and DL query are the highly used fast processes query languages. Thus, we are processing question and data through all the available algorithm to fetch the answer very fast.

## **4. Specific Datasets**

### **4.1 Our Own Dataset**

For our project implementation, we have considered two datasets as follows:

#### **4.1.1 WikiRef220**

- WikiRef220 is the collection of the news article, taken from the Wikipedia pages.

- This dataset includes the information in the form of text data.
- The articles included in this dataset are November 2015 Paris attack, Flight 370 Malaysian Airlines, premier league, Michelle Obama, Samsung Galaxy.
- URL: [http://mklab.itι.gr/files/WikiRef\\_dataset.zip](http://mklab.itι.gr/files/WikiRef_dataset.zip)

#### **4.1.2 BBC News-** In this especially we have selected politics area and sports.

- This dataset includes the news article from collected from BBC.
- This dataset was made available mainly for machine learning research. We are using this dataset for our process.
- We mainly selected the political area and sports area of the BBC news dataset.
- URL: <http://mlg.ucd.ie/datasets/bbc.html>

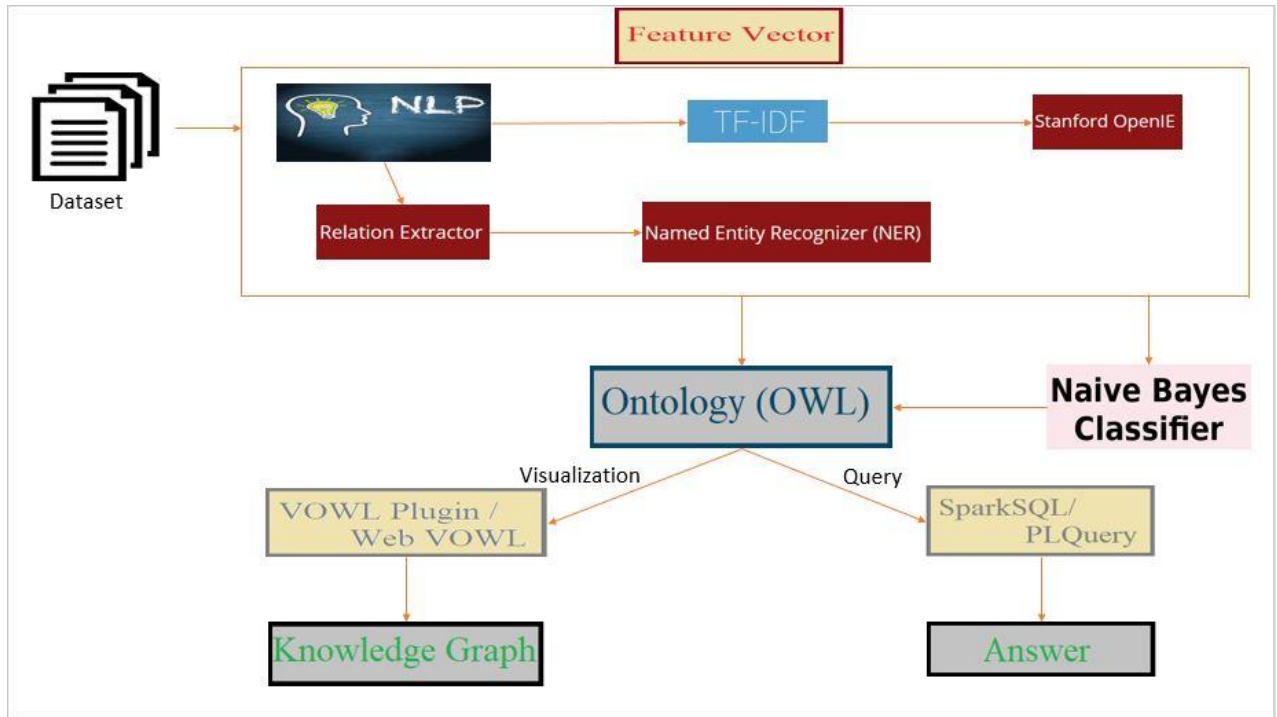
### **4.2 Stanford Dataset**

- Stanford question answer dataset is the collection of the data from Wikipedia. All this data was collected from real time question answer from google.
- This data set consists of more than 100000 pairs of question answer from more than 500 articles.
- URL: <https://rajpurkar.github.io/SQuAD-explorer/explore/1.1/dev/Construction.html>

### **4.3 Yahoo Dataset**

- Yahoo question dataset are the collection of question answer pairs from yahoo community forum.
- URL:  
[https://www.yahoo.com/?err=404&err\\_url=https%3a%2f%2fanswers.yahoo.com%2f%29](https://www.yahoo.com/?err=404&err_url=https%3a%2f%2fanswers.yahoo.com%2f%29)

## 5. Design



### a) Workflow

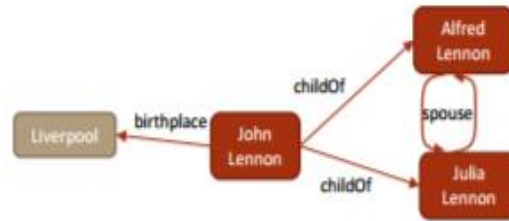
Step 1: Natural language processing – This process includes the identification of token, lemmatization, named entity reference(NER), co-reference resolution.



Step 2: Information Retrieval – Retrieving the information from the text. We are including the identification of the NER i.e., PERSON, LOCATION, ORGANIZATION.

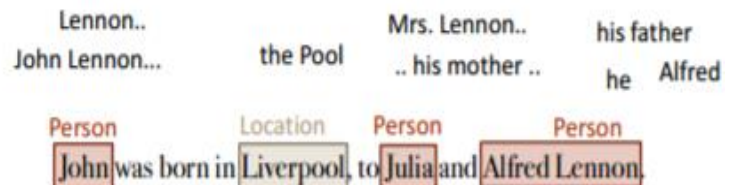
Information  
Extraction

Entity resolution,  
Entity linking,  
Relation extraction...



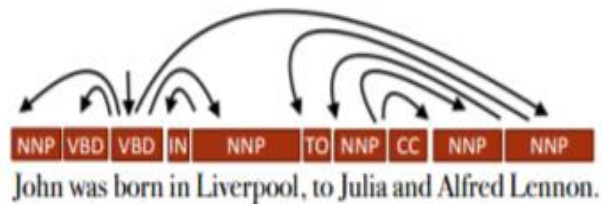
Document

Coreference Resolution...



Sentence

Dependency Parsing,  
Part of speech tagging,  
Named entity recognition...



Step 3: Topic Discovery – Topic discovery helps identification of the topics from the context question.

Step 4: Knowledge Graph construction – Construction of the knowledge graph from generated NER.

Step 5: Preparing query for the question.

Step 6: Execute query to generate the answer.

## b) Preprocessing using NLP

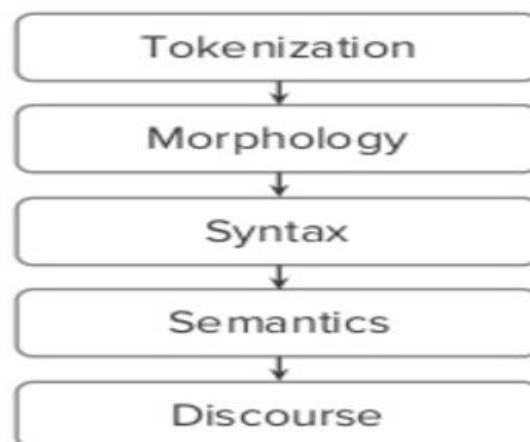




Natural Language Processing is the process that's makes the computer to understand, analyze and extract meaning from human understandable language in a useful and smart way. NLP algorithms helps the organizing and to structure data to perform automatic summarization, named entity recognition, translation, relationship extraction, speech recognition, sentiment analysis, topic segmentation.

Steps in NLP designing:

- Tokenization – Break the text data into sentence, words.
- Lemmatization – Recognizing the base form of word.
- Morphology – Includes Part of Speech recognition, stemming i.e., excluding the postfix words to get the base root word, Named entity recognition.
- Syntax – Parsing Constituency or dependency
- Semantic – Coreference resolution i.e., finding the context that belongs to same entity.



### **c) Information Retrieval**

Information retrieval is the process of tracing through the stored data and recovering specific information from huge amount of stored data. It is very difficult to find the specific data from such a huge amount of data. So, we are using the below approaches to simplify the information retrieval process.

#### **i) Term Frequency Inverse Document Frequency(TFIDF)**

TFIDF is the numerical weight of the tokenized word that demonstrate the importance of the word in the huge document. The weight of the word increases with the repetition of word in the document. TFIDF is can be represented as TF\*IDF i.e., product of term frequency i.e., occurrence of word in a document and Inverse document frequency i.e., log value number of document the word exists divided by the total number of documents.

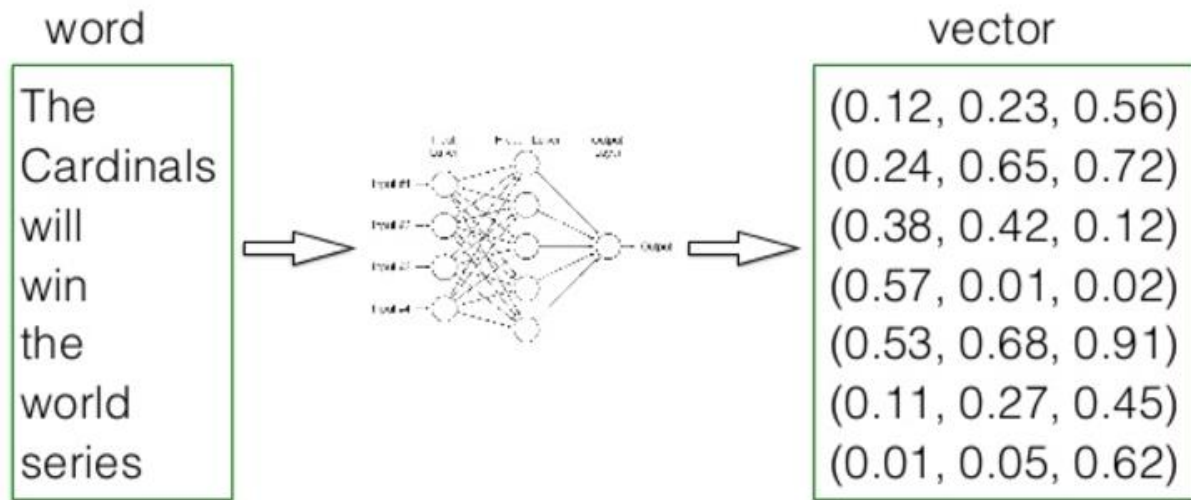
$$w_{i,j} = tf_{i,j} \times \log \left( \frac{N}{df_i} \right)$$

$tf_{i,j}$  = number of occurrences of  $i$  in  $j$   
 $df_i$  = number of documents containing  $i$   
 $N$  = total number of documents

## ii) Word2Vector

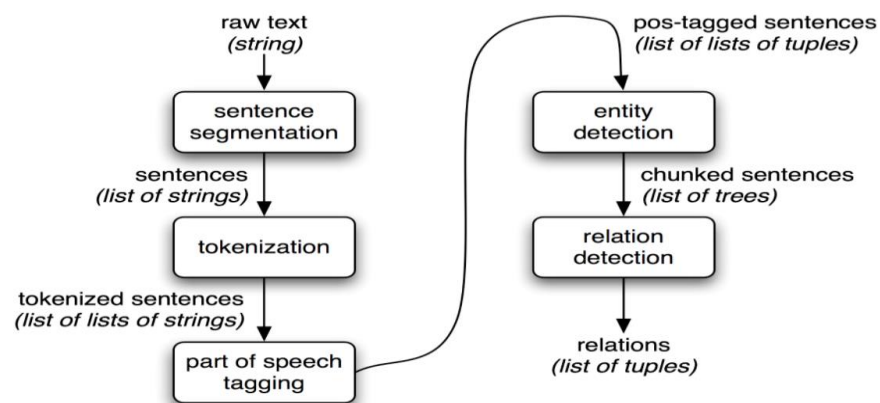
Word2Vec is the process of construction of the vector from the huge text document. All the word vectors are marked in the vector space where the closely meaning words are very close to each other. Thus, mean that they are the same grouped words.

This model leads to the other distributed representation model i.e., Continuous bag of words, Skip gram. Bag of words mean predicting the words from context and the skip gram is predicting the context from words.



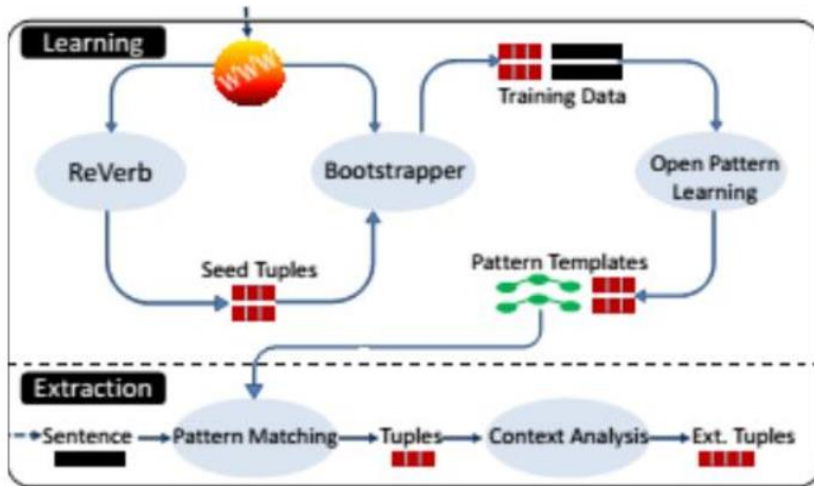
#### d) Information Extraction

Information extraction involves the process of extracting the information from the unstructured or the semi structured data i.e., normal text document. Information extraction utilizes the NLP process to extract the relationship between the entities.



#### i) OpenIE

Open information extraction is the process of extracting the RDF triplets. RDF triplets are subject, object, predicate.



Steps in OpenIE Triplets Extraction:

- Input the data to the system.
- Matching the pattern from already predefined algorithm.
- Extracting the tuples.
- Analyze the context.
- Extracting RDF triplets.

## ii) WordNet

WordNet involves the generation of the synonym for a token of word. WordNet algorithm in analyze the data to extract the correct information though we use the synonym of the word. WordNet generate the synsets, which is the group of words with similar meaning.

## e) Machine Learning

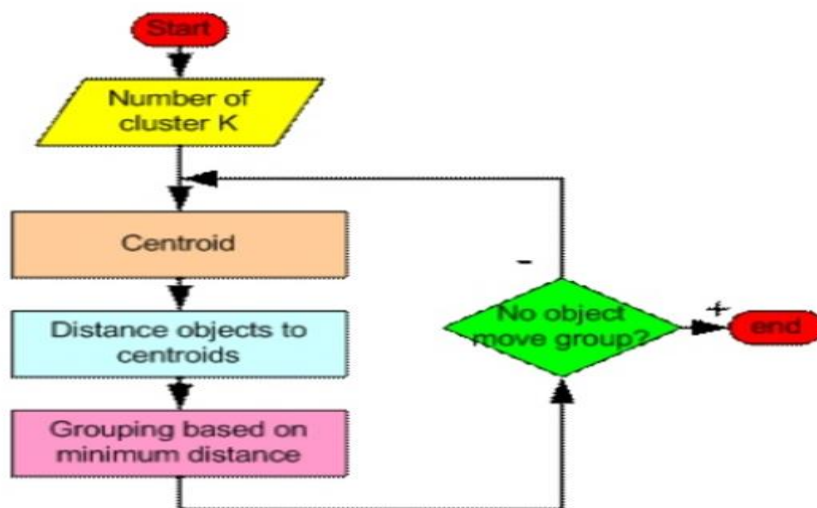
Machine learning involves the process of automatic analyzation of data using the advances artificial intelligence algorithm. This process simplifies the prediction from the existing huge data. Machine learning algorithm are very efficient.

## Clustering

Cluster represent the group of similar kind. In data analyzation, we use clustering process to group together similar words using vector.

### i) K-Mean:

K-mean is a clustering technique,

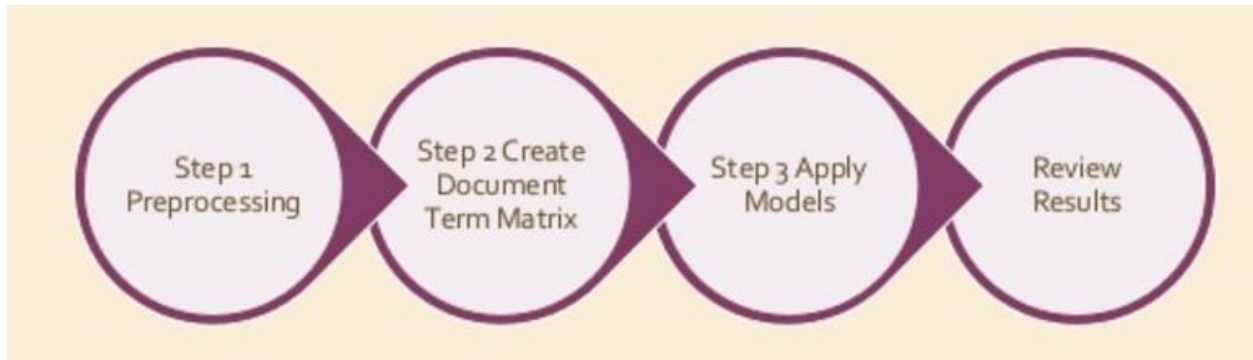


Steps involved in k-mean clustering:

- Input the dataset.
- Tokenize the input data.
- Implement the lemmatization i.e., generating the dictionary word.
- Remove the stop words.
- Generate the TFIDF.
- Determine the Kmeans.

### ii) Latent Dirichiet Allocation:

LDA is a clustering technique, used to extract the topics.



Steps involved in LDA clustering:

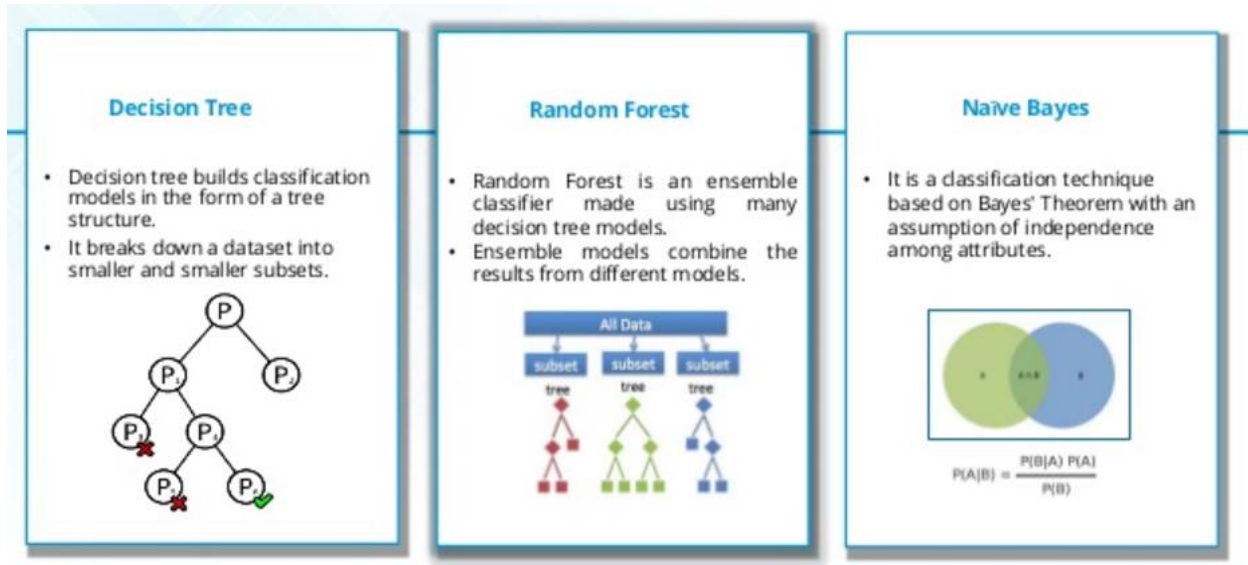
- Input the data i.e., either text data or question.
- Tokenize the input data.
- Implement the lemmatization i.e., generating the dictionary word.
- Remove stop words including punctuation.
- Run the spark LDA, to generate topics.

### iii) LDA vs Kmean Clustering:

S NO	Latent Dirichiet Allocation	Kmean Clustering
1	Output is the collection of topics from the words in the datasets.	Generate the distinct topic collections
2	More realistic approach than Kmean.	Output is k disjoint clusters.

### Classification

Classification is the extension of kmean clustering. There exists decision tree, naïve Bayes, random forest approach for classification. Below are the different classification approaches available.



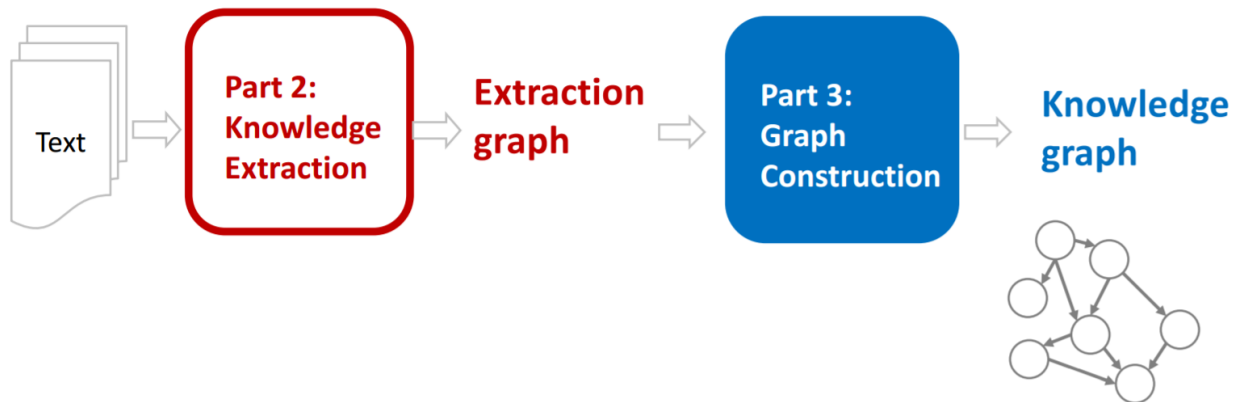
Steps involved in classification:

- Input the dataset.
- Tokenize the input data.
- Implement the lemmatization i.e., generating the dictionary word.
- Remove the stop words.
- Generate the TFIDF.
- Process one of the above classification approach.

## f) Knowledge Graph Construction

Knowledge Graph is used to simplify the search results. This graph represents the graphical representation of the flow of the text data. The main advantage of using this knowledge graph is simplified diagrammatical representation of the huge data, helps in easy knowledge transfer and documentation easy.

## i) Design workflow of knowledge Graph



Steps followed in designing this knowledge graph:

1. Recognizing the named entity reference including the people, organization, location, date etc.
2. Extracting the Classes, Subclasses, Triplets.
3. Designing the data schema i.e., finding the relationship between these entities including data properties, object properties.
4. Constructing the owl file for data set.
5. Representing them in diagrammatical graph using protégé tool or webVowl.

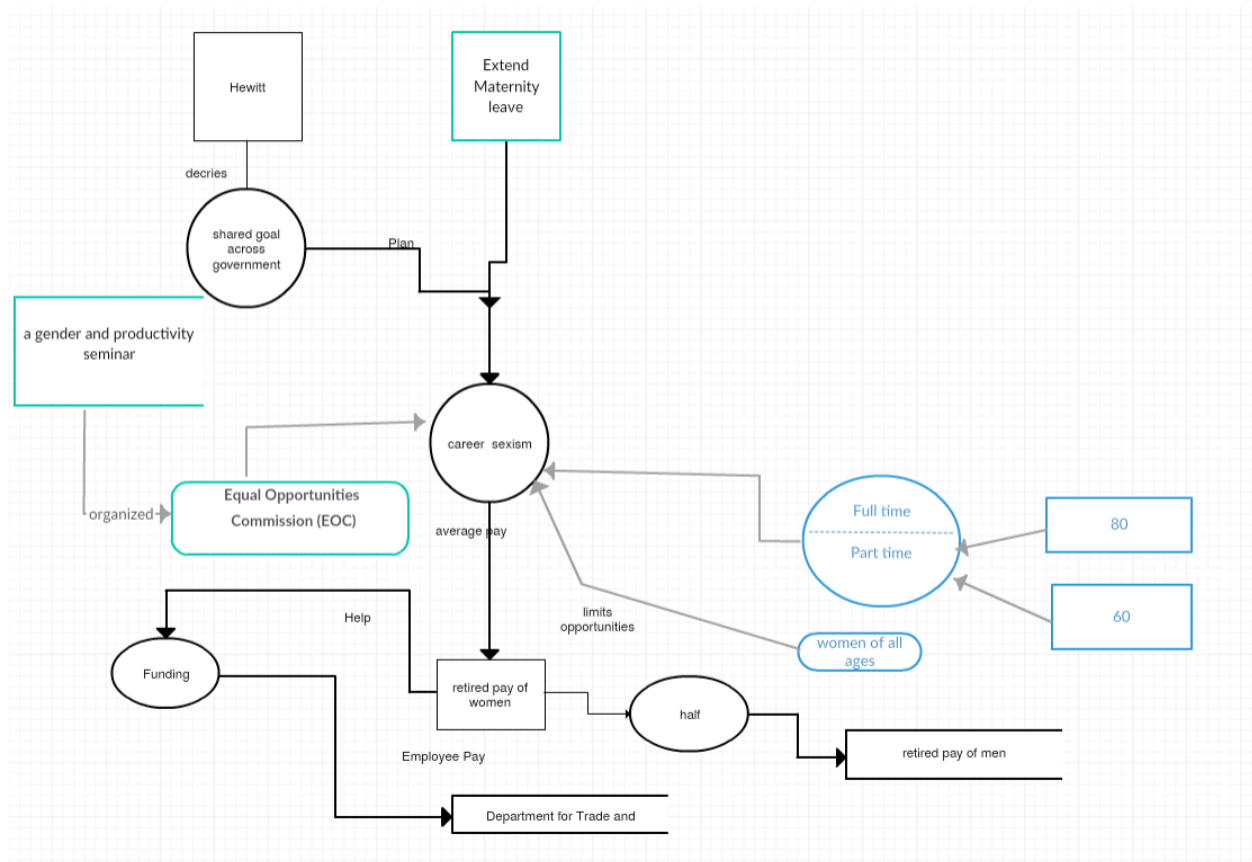
## ii) Knowledge Graph for our dataset:

We do not have any specified rules for designing this knowledge graph. Different companies have their own knowledge graph construction and follows their own rules.

We first recognized the entities in our dataset and designed the data schema to generate the relationships between the entities. Finalized the flow of data.

Below is the diagrammatical representation of the knowledge graph that is designed for our datasets.





### g) Querying for Data:

Spark query or DL query are querying types we are using our application. Constructing a query for a question to extract the answer is very fast and gives us the high accuracy answer.

#### i) Spark Query:

Spark Sql is the structured query language which is used to query in the spark language program. This is like the general query language.

To fetch the answer for a question who are the people in the community whose occupation is student can be written as below.

*SELECT ?persons*

*WHERE { ?persons x:hasOccupation?Occupation}*

*group by ?Occupations=student*

## ii) DL Query:

DL query is the simplified version implemented in protégé tool to fetch the instances for the question. It is more simple and fast.

To fetch the instances of people whose occupation is student can be written as :

*hasOccupation value "student"*

## h) Question-Answering

### i) A Question-Answer Set for our Dataset.

We are designing the questions from datasets considering mainly the PERSON, LOCATION, ORGANIZATION, NUMBER entity.

1. When was Obama born?  
Born on Aug. 4, 1961.
2. Where did Obama did his schooling?  
Punahou School.
3. Who is father of Obama?  
Barack Hussein Obama.
4. Whom did Obama compete in primary race?  
Hillary Rodham Clinton.
5. What is the minimum duration for maternity leave?  
6 months.
6. What is the topic about?  
career sexism.

7. Who is the speaker?

Ms. Hewitt.

8. What is the average pay for full-time women.

80p

9. What is the average pay for part-time women.

60p.

10. What is the average pay for retired women compared to men?

Half.

## **ii) Knowledge Graph to extract answer:**

Knowledge Graph is the graphic representation with instances of the properties between the entities.

STEPS involved:

1. Input the dataset for which we want to construct the knowledge graph.
2. Generate the entities i.e., classes, subclasses, data properties, object properties, Triplets.
3. Construct the. owl.
4. Visualize using the protégé vowl plugin or webvowl online.

## **iii) Querying for answering:**

We generate the query for the question using either DL query or SPARK sql query and execute. This will fetch us the answer for the question either in the table form for spark sql or the instances for DL query.

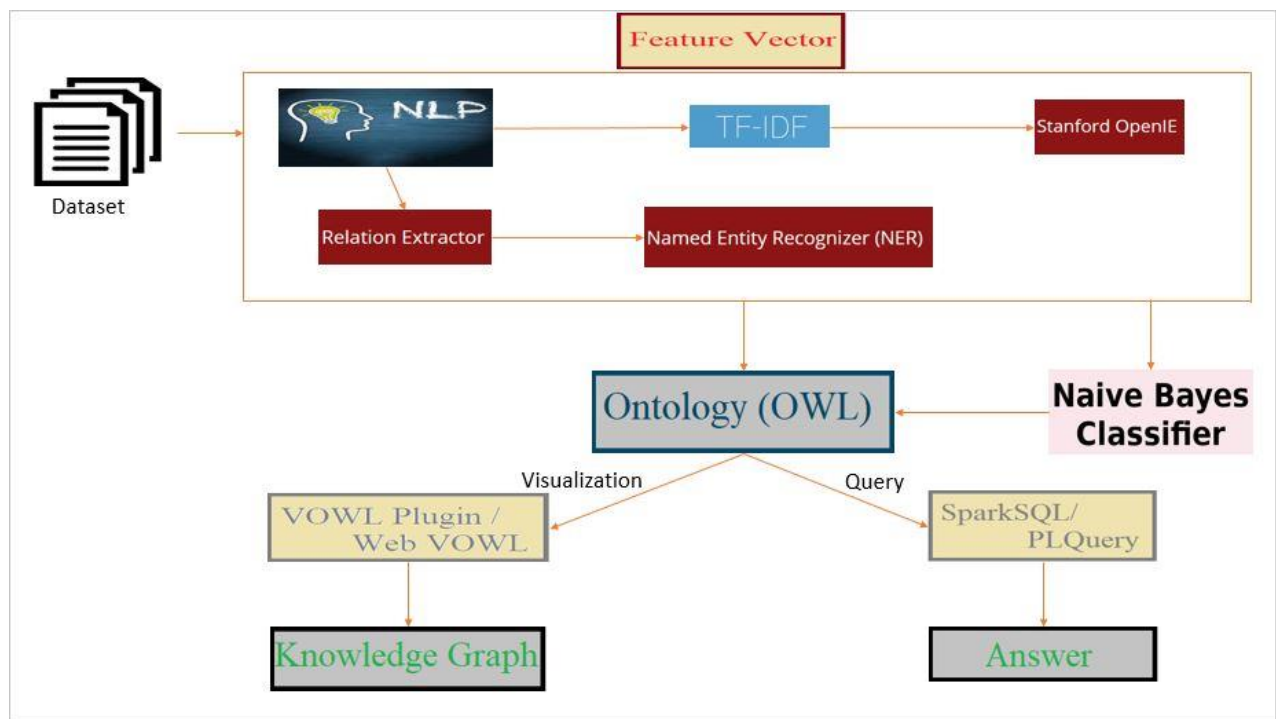
STEPS involved:

1. Construct the. owl for the dataset and generate the knowledge graph.
2. Define the question for which we are expecting the answer.
3. Construct the query.
4. Execute in protégé for the answer.

## 6. Implementation

### 6.a Workflow diagram for our dataset

As the diagram mentions we are taking the question and the dataset and we are applying the NLP operations on the dataset and then we are storing the result of the NLP and on top of it applying the other shown approaches for the better performance.



### 6.b Preprocessing using NLP

#### Output of NLP operations for our dataset

We have performed the NLP operations on the dataset which we have chosen and the result of each operation is shown in the below mentioned screenshots.

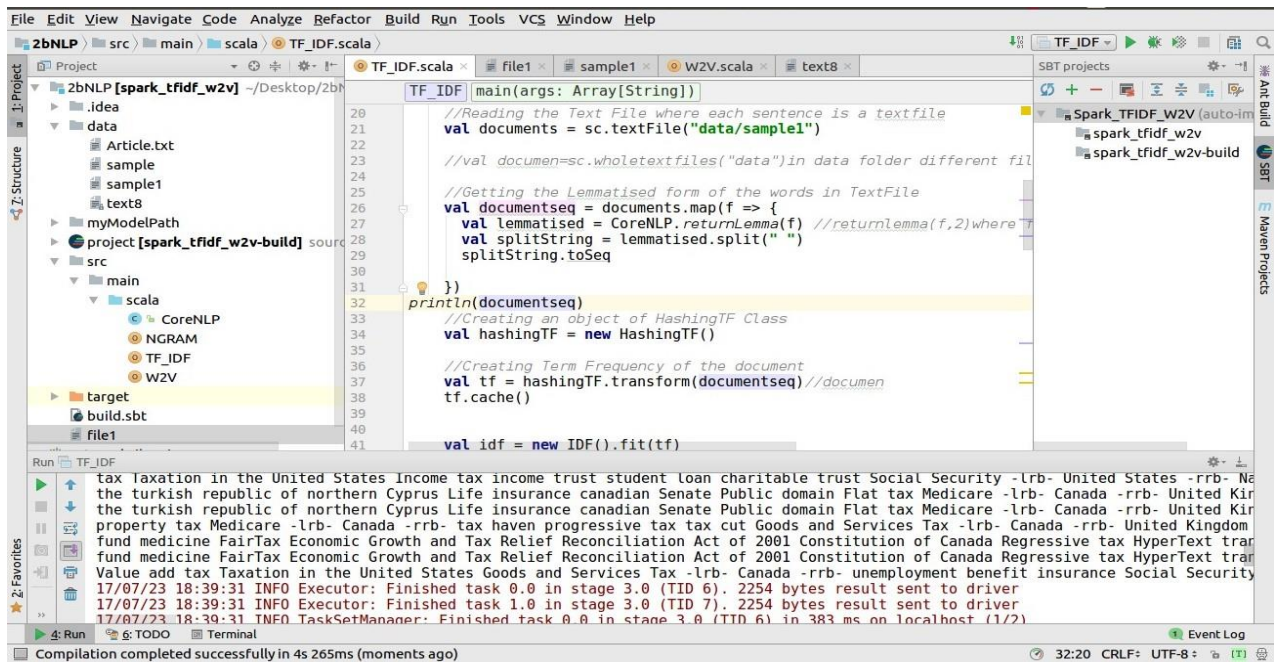
Tokenization:

Lemmatization:

POS Tagging:

NER:

Below is the code for all the operations of the NLP performed on our dataset.



The screenshot shows an IDE with a Scala file named `TF_IDF.scala`. The code defines a `main` function that reads a text file, tokenizes it, lemmatizes the words, and calculates the Term Frequency (TF) for each document. The output of the program is displayed in the Run console, showing a list of words and their corresponding TF values for various documents.

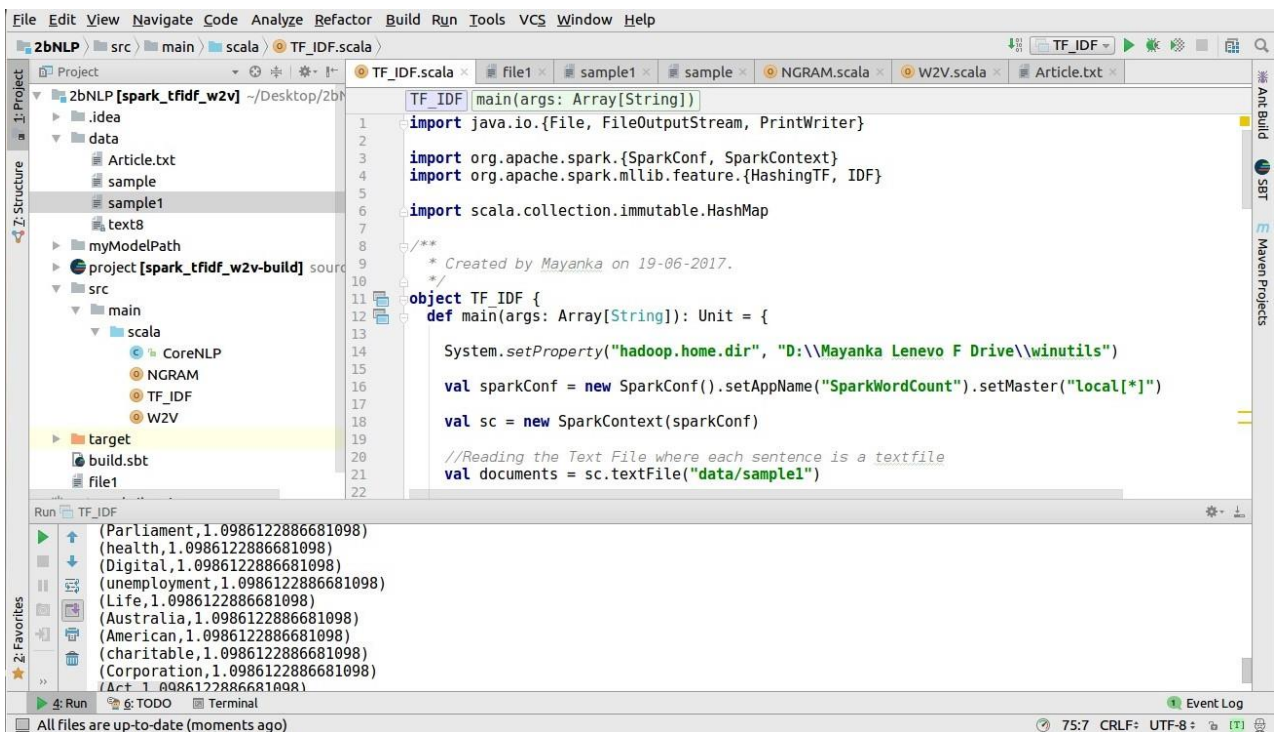
```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
2bNLP [src] main scala TF_IDF.scala file1 sample1 W2V.scala text8
TF_IDF main(args: Array[String])
20 //Reading the Text File where each sentence is a textfile
21 val documents = sc.textFile("data/sample1")
22
23 //val document=sc.wholeTextFiles("data")in data folder different file
24
25 //Getting the Lemmatized form of the words in TextFile
26 val documentseq = documents.map(f => {
27   val lemmatised = CoreNLP.returnLemma(f) //returnLemma(f,2)where
28   val splitString = lemmatised.split(" ")
29   splitString.toSeq
30 })
31
32 println(documentseq)
33 //Creating an object of HashingTF Class
34 val hashingTF = new HashingTF()
35
36 //Creating Term Frequency of the document
37 val tf = hashingTF.transform(documentseq) //document
38 tf.cache()
39
40
41 val idf = new IDF().fit(tf)

Run TF_IDF
tax taxation in the United States Income tax income trust student loan charitable trust Social Security -lrb- United States -rrb- Ne
the turkish republic of northern Cyprus Life insurance canadian Senate Public domain Flat tax Medicare -lrb- Canada -rrb- United Kir
the turkish republic of northern Cyprus Life insurance canadian Senate Public domain Flat tax Medicare -lrb- Canada -rrb- United Kir
property tax Medicare -lrb- Canada -rrb- tax haven progressive tax tax cut Goods and Services Tax -lrb- Canada -rrb- United Kingdom
fund medicine FairTax Economic Growth and Tax Relief Reconciliation Act of 2001 Constitution of Canada Regressive tax HyperText tran
fund medicine FairTax Economic Growth and Tax Relief Reconciliation Act of 2001 Constitution of Canada Regressive tax HyperText tran
Value add tax Taxation in the United States Goods and Services Tax -lrb- Canada -rrb- unemployment benefit insurance Social Security
17/07/23 18:39:31 INFO Executor: Finished task 0.0 in stage 3.0 (TID 6). 2254 bytes result sent to driver
17/07/23 18:39:31 INFO Executor: Finished task 1.0 in stage 3.0 (TID 7). 2254 bytes result sent to driver
17/07/23 18:39:31 INFO TaskSetManager: Finished task 0.0 in stage 3.0 (TID 6) in 383 ms on localhost (1/2)

Compilation completed successfully in 4s 265ms (moments ago) 32:20 CRLF: UTF-8
```

## 6.c TF-IDF for our dataset

Generating the term frequency for the words in the dataset.



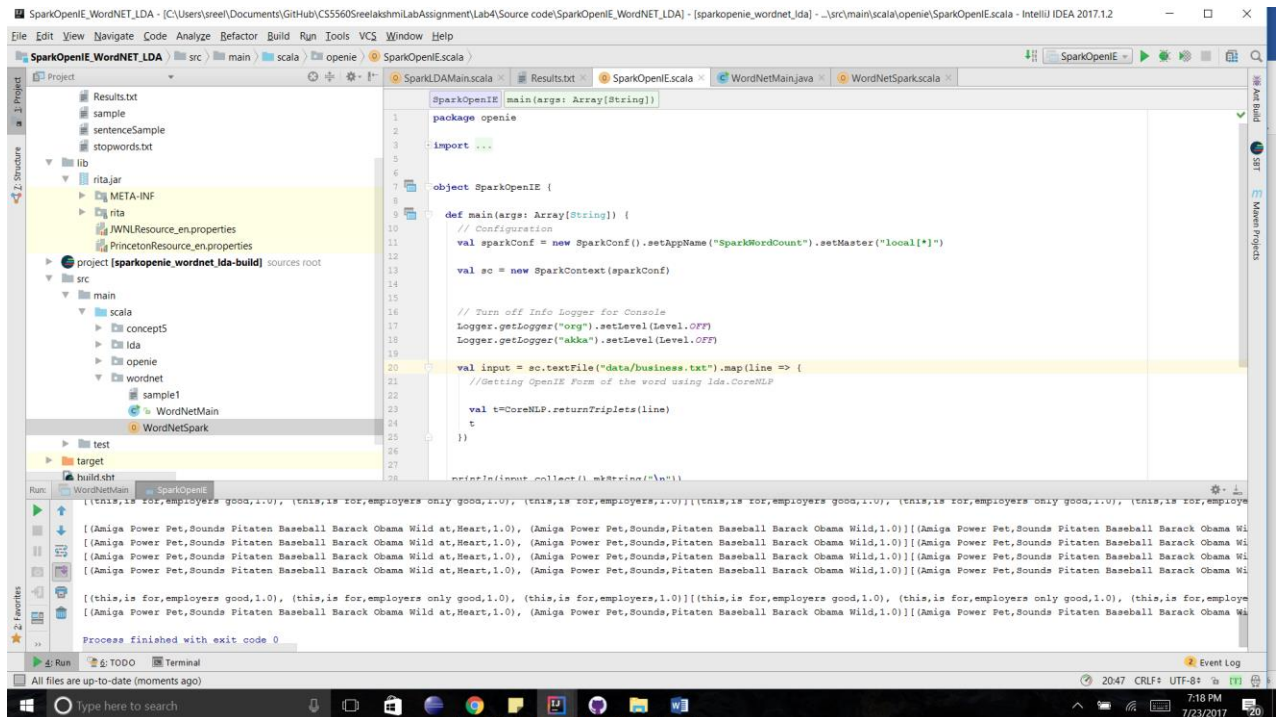
The screenshot shows an IDE with a Scala file named `TF_IDF.scala`. The code defines a `main` function that reads a text file, tokenizes it, and calculates the Term Frequency (TF) for each word. The output of the program is displayed in the Run console, showing a list of words and their corresponding TF values for various documents.

```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
2bNLP [src] main scala TF_IDF.scala file1 sample1 sample NGRAM.scala W2V.scala Article.txt
TF_IDF main(args: Array[String])
1 import java.io.{File, FileOutputStream, PrintWriter}
2
3 import org.apache.spark.{SparkConf, SparkContext}
4 import org.apache.spark.mllib.feature.{HashingTF, IDF}
5
6 import scala.collection.immutable.HashMap
7
8 /**
9  * Created by Mayanka on 19-06-2017.
10  */
11 object TF_IDF {
12   def main(args: Array[String]): Unit = {
13
14     System.setProperty("hadoop.home.dir", "D:\\Mayanka Lenevo F Drive\\winutils")
15
16     val sparkConf = new SparkConf().setAppName("SparkWordCount").setMaster("local[*]")
17
18     val sc = new SparkContext(sparkConf)
19
20     //Reading the Text File where each sentence is a textfile
21     val documents = sc.textFile("data/sample1")
22
Run TF_IDF
(Parlament,1.0986122886681098)
(health,1.0986122886681098)
(Digital,1.0986122886681098)
(unemployment,1.0986122886681098)
(Life,1.0986122886681098)
(Australia,1.0986122886681098)
(American,1.0986122886681098)
(charitable,1.0986122886681098)
(Corporation,1.0986122886681098)
(Act,1.0986122886681098)

All files are up-to-date (moments ago) 75:7 CRLF: UTF-8
```

## 6.d Information Extraction

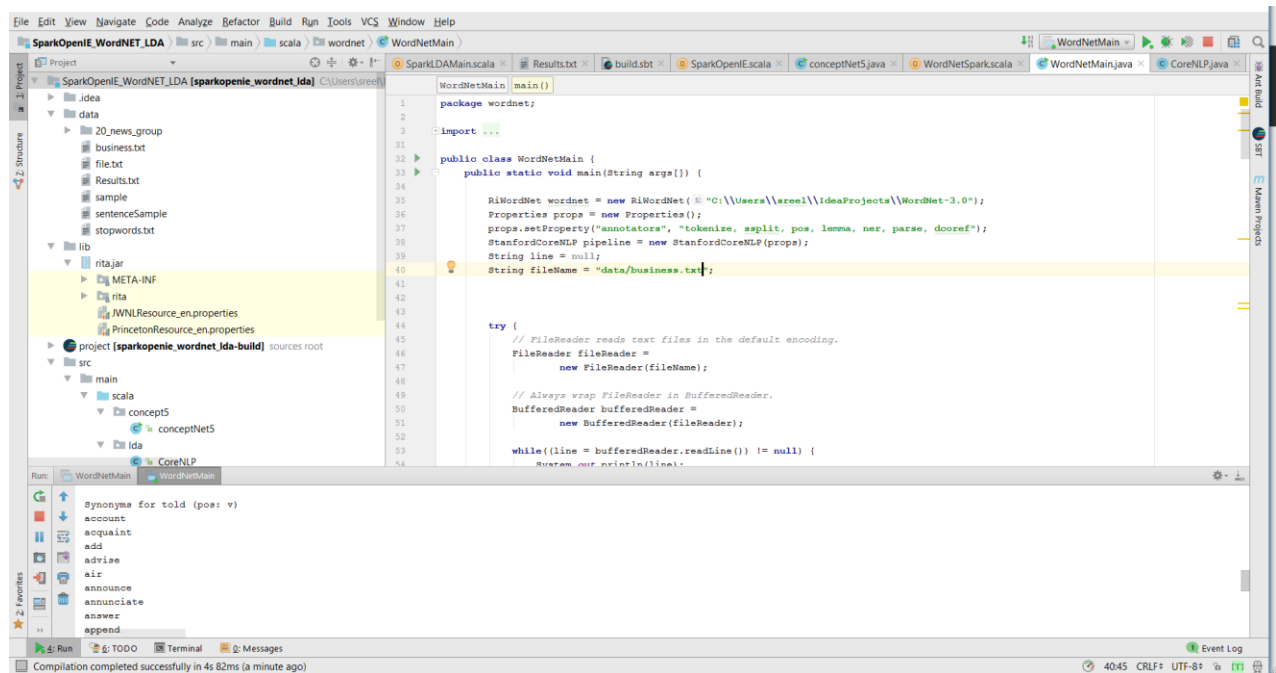
### OpenIE for our dataset:



```
1 package openie
2
3 import ...
4
5
6
7 object SparkOpenIE {
8
9   def main(args: Array[String]) {
10     // Configuration
11     val sparkConf = new SparkConf().setAppName("SparkWordCount").setMaster("local[*]")
12
13     val sc = new SparkContext(sparkConf)
14
15     // Turn off Info Logger for Console
16     Logger.getLogger("org").setLevel(Level.OFF)
17     Logger.getLogger("akka").setLevel(Level.OFF)
18
19     val input = sc.textFile("data/business.txt").map(line => {
20       // Getting OpenIE form of the word using lda.CoreNLP
21       val t = CoreNLP.returnTriplets(line)
22     })
23
24     println(input.collect().mkString("\n"))
25   }
26 }
27
```

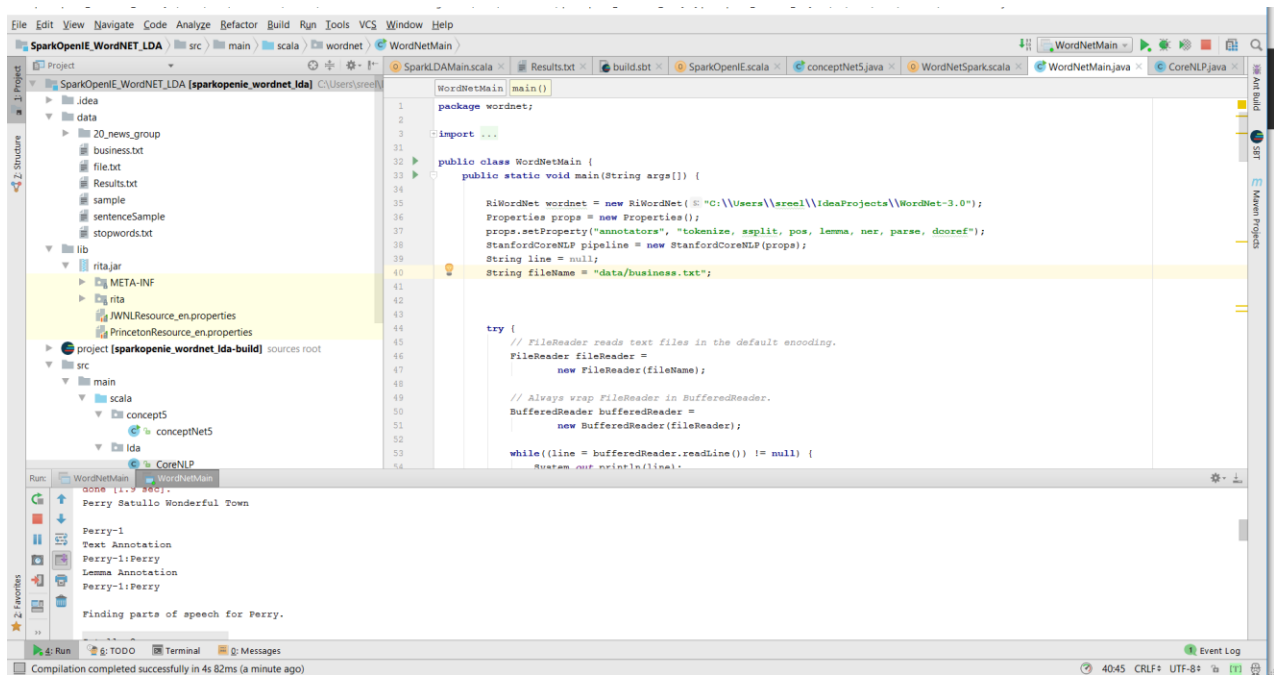
Process finished with exit code 0

### Wordnet for our dataset

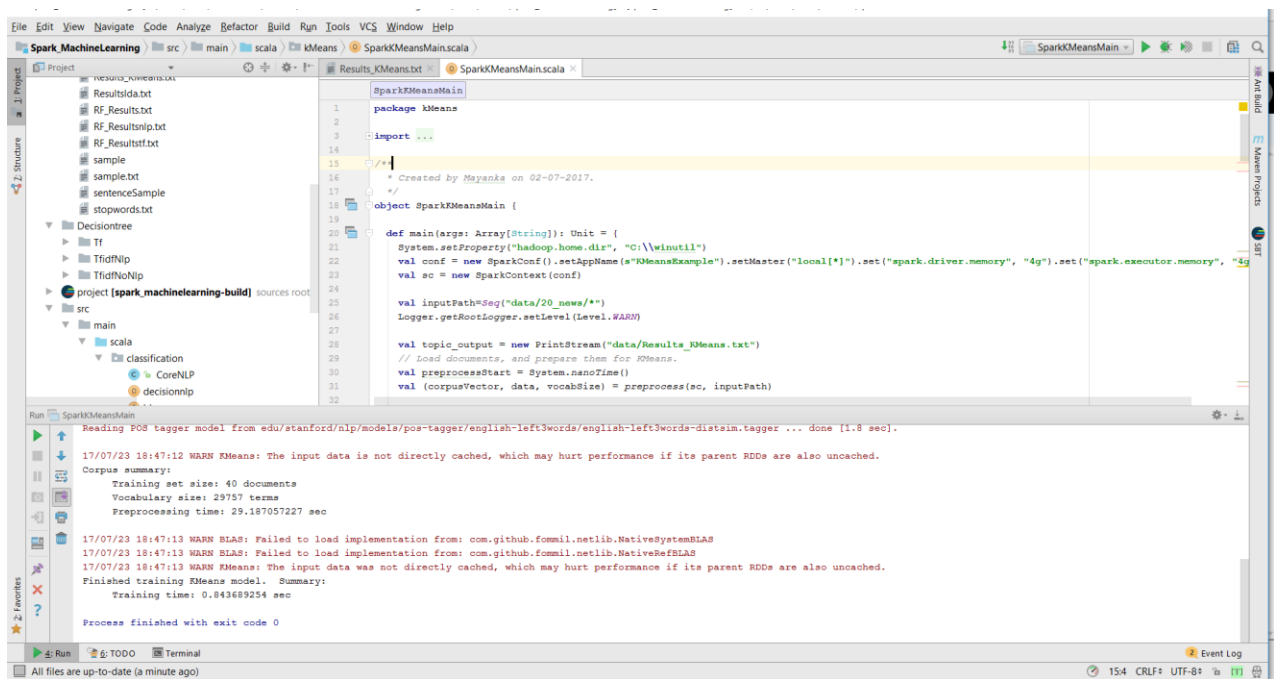


```
1 package wordnet;
2
3 import ...
4
5
6
7 public class WordNetMain {
8
9   public static void main(String args[]) {
10     RIWordNet wordnet = new RIWordNet("C:\\Users\\sreen\\IdeaProjects\\WordNet-3.0");
11     Properties props = new Properties();
12     props.setProperty("annotators", "tokenize, split, pos, lemma, ner, parse, decoref");
13     StanfordCoreNLP pipeline = new StanfordCoreNLP(props);
14     String line = null;
15     String fileName = "data/business.txt";
16
17     try {
18       // FileReader reads text files in the default encoding.
19       FileReader fileReader =
20         new FileReader(fileName);
21
22       // Always wrap FileReader in BufferedReader.
23       BufferedReader bufferedReader =
24         new BufferedReader(fileReader);
25
26       while((line = bufferedReader.readLine()) != null) {
27         System.out.println(line);
28       }
29     }
30   }
31 }
32
```

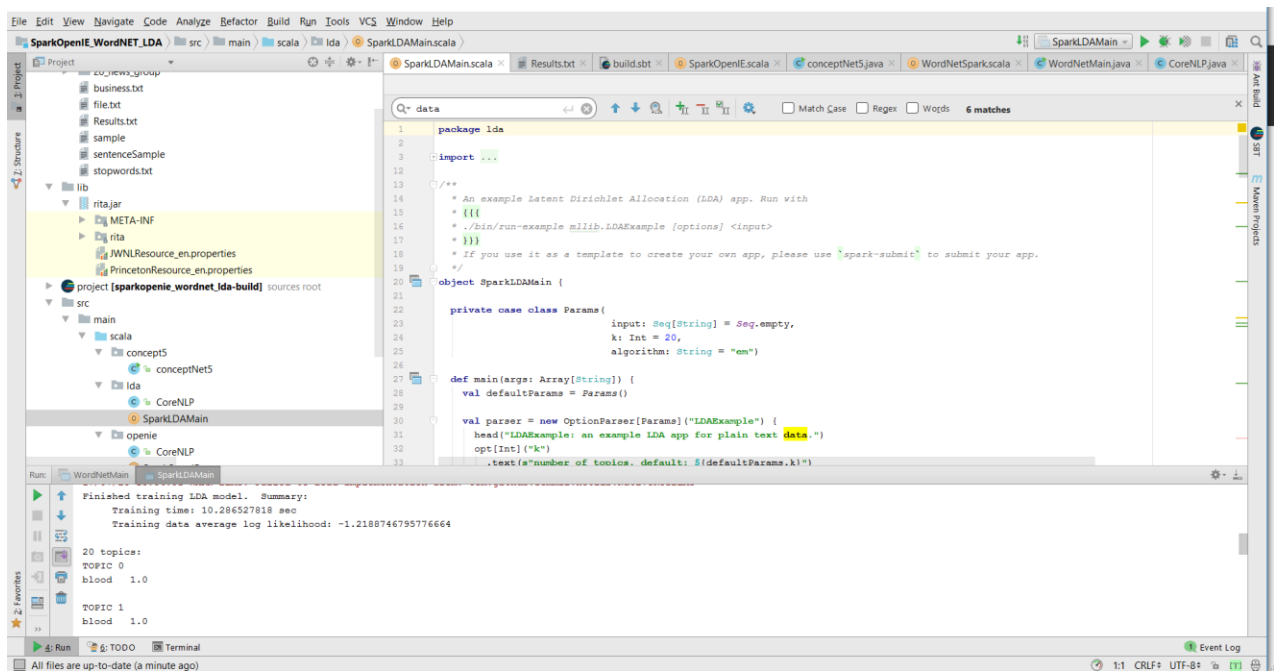
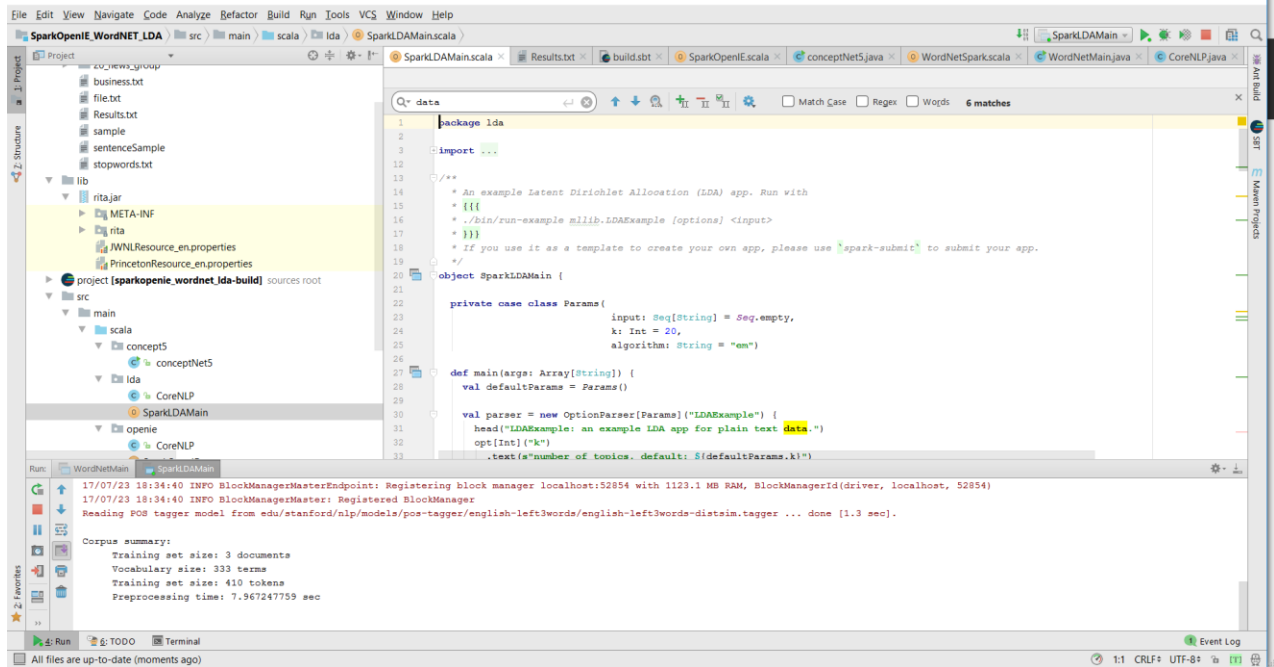
Compilation completed successfully in 4s 82ms (a minute ago)



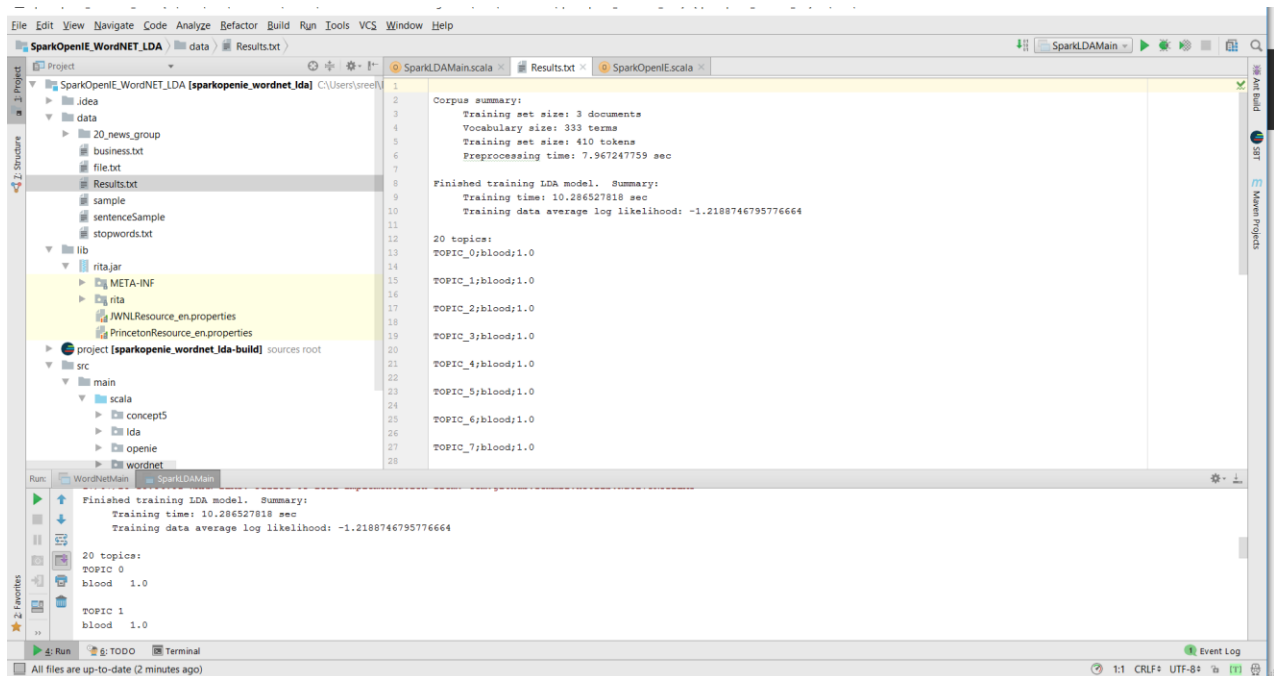
## 6.e Machine learning Techniques KMeans



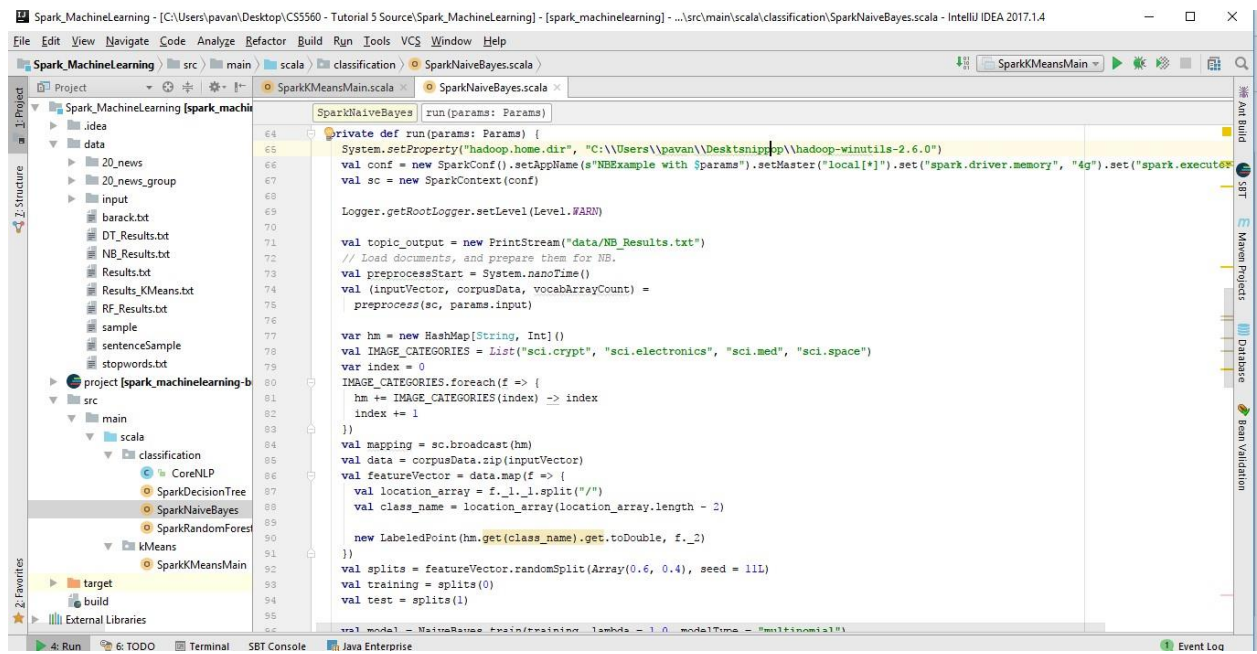




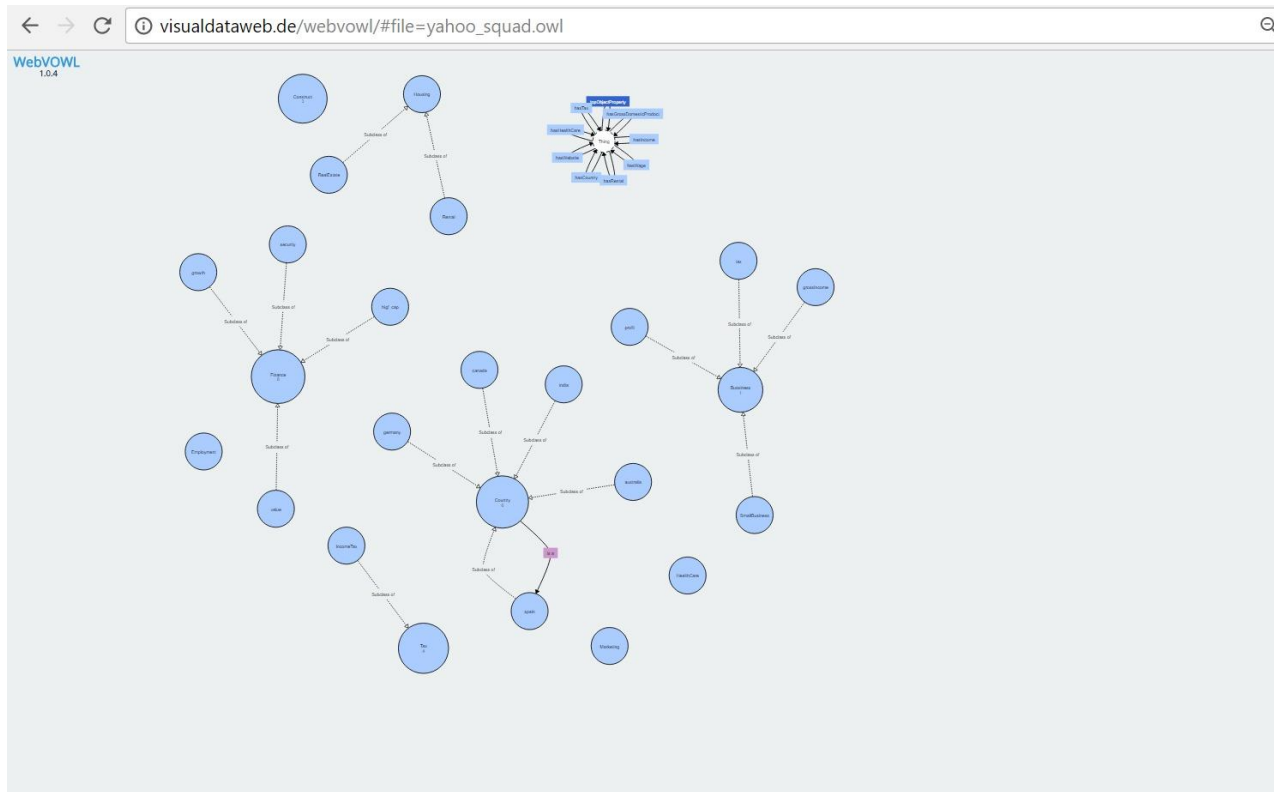




## Naïve Bayes



## 6.f Visualization



### Generation of SPARQL Query:

1. What is the gross domestic product for construction?

Entities × Individuals by class × DL Query × Object Properties × Individuals matrix × VOWL × SPARQL Query × Active Ontology ×

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX x: <http://www.semanticweb.org/putha/ontologies/2017/6/untitled-ontology-21#>
SELECT ?value ?answer
WHERE { ?value x:hasGrossDomesticProduct ?answer }
```

	value	answer
grossDomestic	six_to_nine_percent	

Execute

Reasoner state out of sync with active ontology. ☒ Show Inferences

2. Which distinct country that has tax?

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX x: <http://www.semanticweb.org/putha/ontologies/2017/6/untitled-ontology-21#>
SELECT DISTINCT ?country
WHERE { ?wage x:hasTax ?country }
```

country
Australia
Canada
spain
India

Execute

To use the reasoner click Reasoner > Start reasoner. ☒ Show Inferences

3. What are the wages in different countries?

Object Properties × Individuals by class × Individuals matrix × DL Query × SPARQL Query × VOWL ×

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX x: <http://www.semanticweb.org/putha/ontologies/2017/6/untitled-ontology-21#>
SELECT ?country ?wage
WHERE { ?country x:hasWage ?wage }
```

country	wage
Australia	1200
Canada	1300
Germany	6455
India	689
spain	789

Execute

## Generation of DL Query:

1. Which country has gross wage of 6455?

untitled-ontology-21 (http://www.semanticweb.org/putha/ontologies/2017/6/untitled-ontology-21)

Entities x Individuals by class x DL Query x **VOWL** x Object Properties x

**Class hierarchy: Country**

owl:Thing  
 Business  
 grossIncome  
 profit  
 tax  
 Country  
 australia  
 canada  
 germany  
 india  
 spain  
 Finance  
 growth  
 high\_cap  
 security  
 value

Asserted

**DL query:**

**Query (class expression)**

hasWage value 6455

Execute Add to ontology

**Query results**

Subclasses (1 of 1)

- owl:Nothing

Instances (1 of 1)

- Germany

2. Which country has tax of 30%?

untitled-ontology-21 (http://www.semanticweb.org/putha/ontologies/2017/6/untitled-ontology-21)

Entities x Individuals by class x DL Query x Object Properties x Individuals matrix x **VOWL** x

**Class hierarchy: growth**

owl:Thing  
 Employment  
 Housing  
 Marketing  
 HealthCare  
 Tax  
 Business  
 SmallBusiness  
 grossIncome  
 profit  
 Country  
 australia  
 canada  
 germany  
 india  
 spain  
 Finance  
 growth  
 high\_cap  
 security  
 value

Asserted

**DL query:**

**Query (class expression)**

hasTax value 30

Execute Add to ontology

**Query results**

Subclasses (1 of 1)

- owl:Nothing

Instances (2 of 2)

- India
- Newzealand

**Query for**

- ☐ Direct superclasses
- ☐ Superclasses
- ☐ Equivalent classes
- ☐ Direct subclasses
- ☒ Subclasses
- ☒ Instances

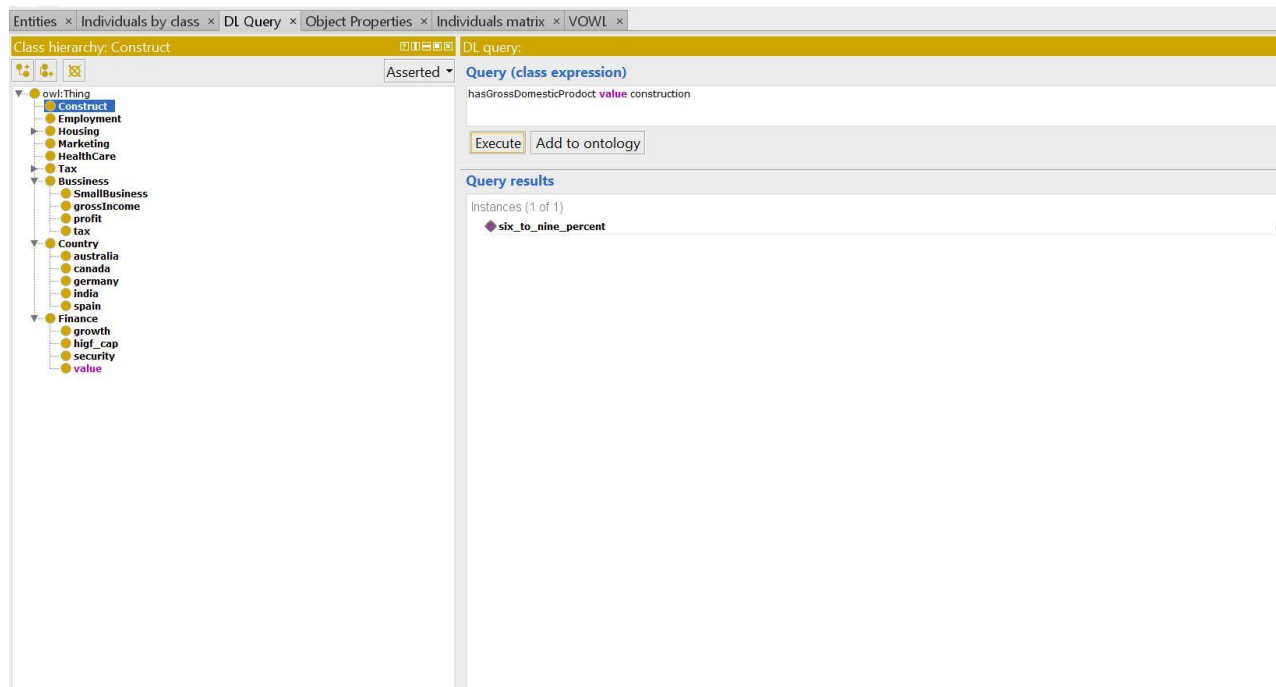
**Result filters**

Name contains

- ☒ Display owl:Thing (in superclass results)
- ☒ Display owl:Nothing (in subclass results)

Reasoner active Show Inferences

3. What is the gross domestic product for construction?



## 7. Results and Evaluation

### 7.1 Datasets used:

- Yahoo Dataset: Business and finance
- SQUAD Dataset: Construction

### 7.2 Describe your results for each step in Design

Below are the results that are generated after each step in the design:

#### 7.2.1 NLP Preprocessing:

- Tokenization – Break the text data into sentence, words.
- Lemmatization – Recognizing the base form of word.
- Morphology – Includes Part of Speech recognition, stemming i.e., excluding the postfix words to get the base root word, Named entity recognition.
- Syntax – Parsing Constituency or dependency
- Semantic – Coreference resolution i.e., finding the context that belongs to same entity.

### **7.2.2 Information Retrieval:**

- TFIDF will retrieve the weight of the word in the document
- Word2Vec will represent the words in the vector i.e., space.

### **7.2.3 Information Extraction:**

- OpenIE is generating the triplets i.e., subject, object, predicate.
- WordNet is generating the synonyms for the word.

### **7.2.4 Machine Learning:**

- Kmean or LDA is generating the topics from the dataset.
- NaiveBayes classifies the topic for cluster into confusion matrix.

### **7.2.5 Knowledge Graph Visualization:**

- Generates the graph from the web ontology language.

### **7.2.6 Querying:**

- Designing the query for the question.
- This is done by either DL query or spark sql query.

### **7.2.7 Question Answering:**

- Execute the query to fetch the required result which would answer the question.

## **7.3 Evaluation and Validation:**

- Accuracy: 60%
- Performance Time: 0.873 sec

## **7.4 Incorrect and correct cases:**

- Kmean clustering is better in performance than the LDA as kmean extracts the unique topics.

- Spark SQL is more accurate and high performance than DL query. As spark SQL has group by, orderBy approaches.

## **8. Project Management**

### **Programming Language Used:**

We have collaborated various languages in the development of the project and in building the application. Some of them are,

- Java
- Scala
- Spark

### **IDE Used:**

Integrated development environments, helps in easy development of software with the facility of comprehensive integrated environment.

- IntelliJ
- PyCharm
- Protege

### **a. Contributors**

- Jakkepalli, Rama Charan Pavan - **25%**
- Puthana, Sujitha - **25%**
- Yalamanchili, Sowmya - **25%**
- Nandanamudi, Sreelakshmi - **25%**





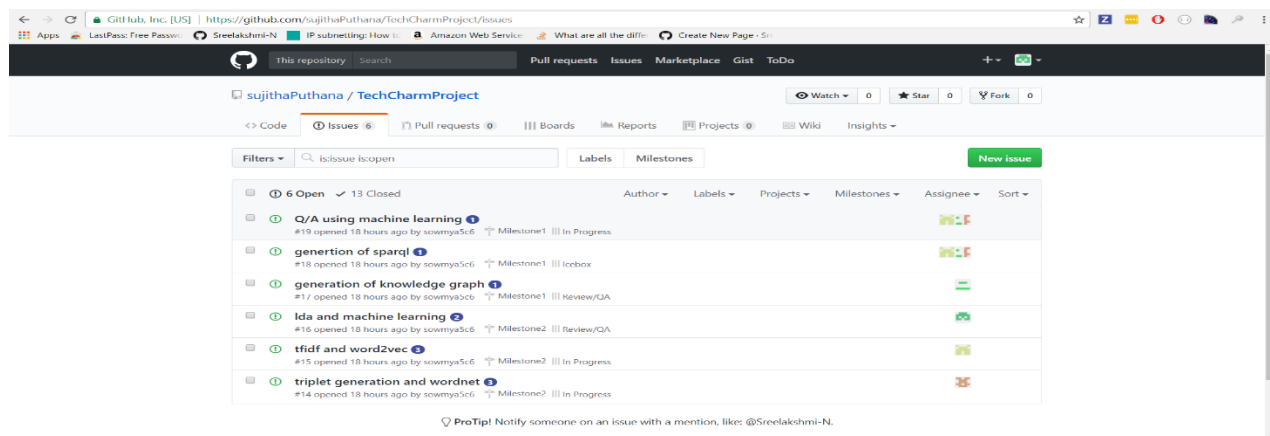
Name	Implementation	Documentation
Pavan	<b>Increment-1:</b> Basic Question Answer System  <b>Increment-2</b> K-means  Question answer using TF-IDF  <b>Increment-3:</b> Data Set collection and NLP.	<b>Increment-1:</b> Domain, Specific Dataset, Future Work  <b>Increment-2:</b> Added more description to document, related work, machine learning  <b>Increment 3:</b> System Design, Results description

Sujitha	<p><b>Increment-1:</b> TF-IDF</p> <p><b>Increment-2:</b> Classification Algorithm TF-IDF question and answer</p> <p><b>Increment-3:</b> OWL construction, Dl query, Sql Query</p>	<p><b>Increment-1:</b> Design workflow, Question Answer, Knowledge Graph</p> <p><b>Increment-2:</b> document, related work, machine Learning</p> <p><b>Increment 3:</b> Related Work, Design</p>
Sowmya	<p><b>Increment-1:</b> Core NLP</p> <p><b>Increment-2:</b> Wordnet Question answer using openIE</p> <p><b>Increment-3:</b> Triplets, Classes, subclasses, object properties, data properties Extraction</p>	<p><b>Increment-1:</b> Project Motivation, Objective, Significance.</p> <p><b>Increment-2:</b> Design of Information Retrieval</p> <p><b>Increment-3:</b> Implementation screenshots, ZenHub tasks creation</p>

Sreelakshmi	<p><b>Increment-1:</b></p> <p>Named Entity Recognition</p> <p><b>Increment-2</b></p> <p>OpenIE</p> <p>Question answer using openIE</p> <p><b>Increment-3:</b></p> <p>Querying, owl construction</p>	<p><b>Increment-1:</b></p> <p>Contribution, Milestone, issues creation,</p> <p>Work Completed</p> <p><b>Increment-2</b></p> <p>Design of Machine Learning</p> <p><b>Increment-3:</b></p> <p>Implementation screenshots, ZenHub tasks creation</p>
-------------	---	---

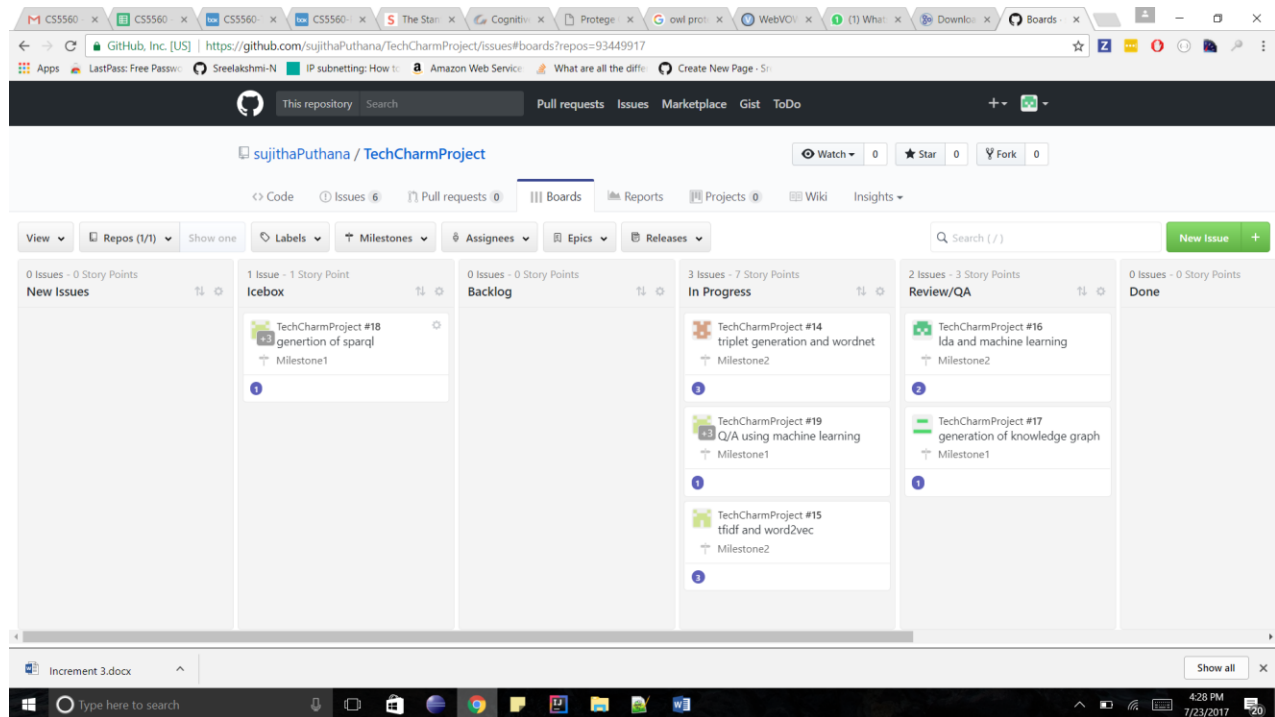
## b. Zen-Hub Screenshots

For the first increment, we had issues regarding the working of the questions and answers section and generating the NLP output for the dataset we have chosen as the size of the dataset is larger.



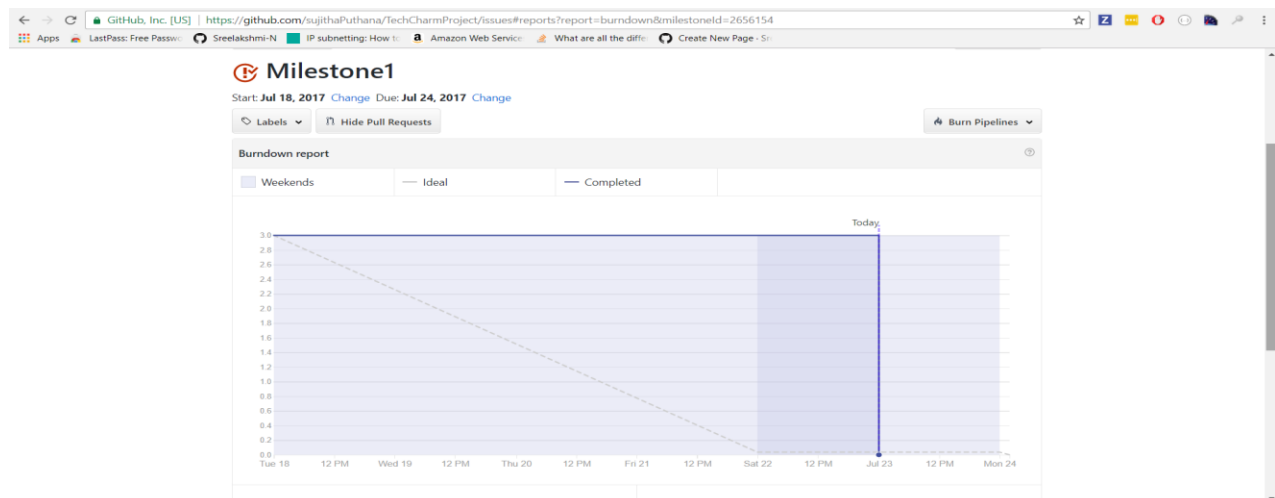
## Project Timeline, Members, and Task Responsibility

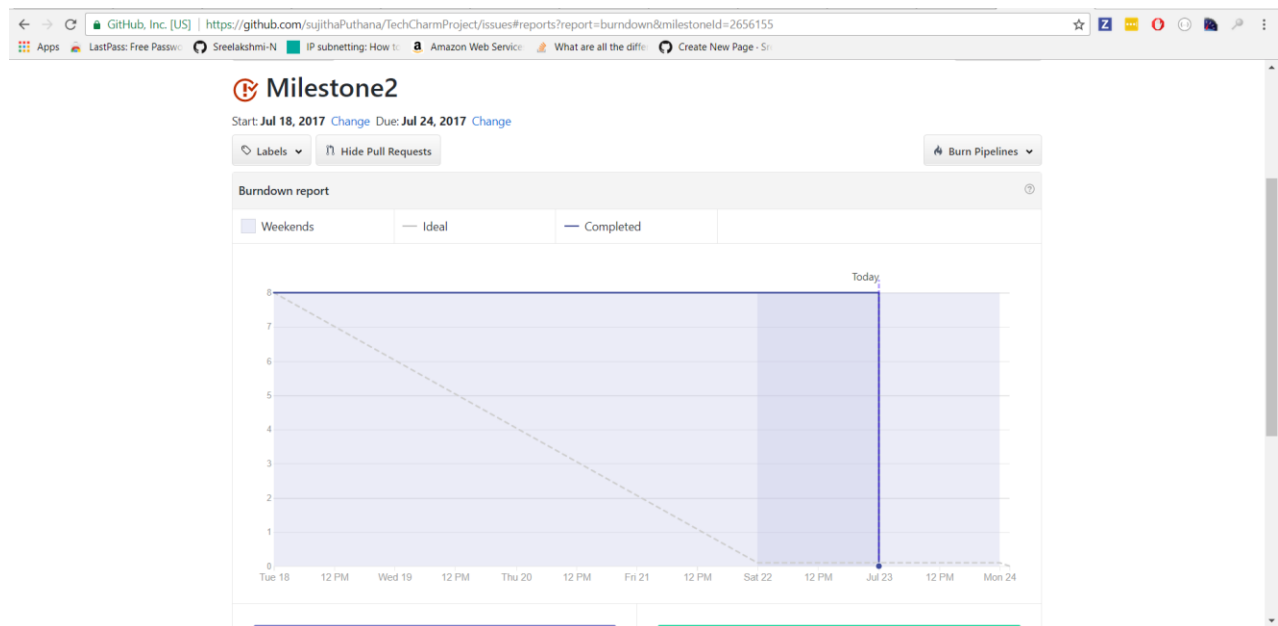
The issues that are registered and current one's which we are working are updated and can be viewed in GitHub repository. The below screenshot will show you the issues and their respective categorization's i.e. New issues, Icebox, Backlog, In Progress, Review, QA, Done.



## Burn-Down Chart:

Burn-Down chart is created for the above issues via Milestones in GitHub. Below is the screenshot for more information,





## GitHub Wiki Page

The GitHub wiki page URL for the screenshots and the process flow is updated in the following link

- <https://github.com/sujithaPuthana/TechcharmProject>

## Work Completed

The completed tasks in this increment are,

- Performed the TF-IDF and N-Gram approach for the dataset and embedded that in the question answering system.
- Implemented the OpenIE, wordnet, clustering and classification techniques for the dataset.
- Question and answer system using the OpenIE.

## c. Concerns

- Faced an issue while implementing the k-means algorithm to the dataset.
- While integrating the OpenIE with the question answering system a bit issue we faced. 4

#### **d. Future Work**

- We need to implement the question and answer approach using the K-means integrated with the NLP operations.
- Need to integrate various techniques for better question and answering system.

#### **Bibliography**

1. <https://blog.algorithmia.com/introduction-natural-language-processing-nlp/>
2. [https://en.wikipedia.org/wiki/Question\\_answering](https://en.wikipedia.org/wiki/Question_answering)
3. <https://nlp.stanford.edu/>
4. <http://visualdataweb.de/webvowl/>
5. <https://rajpurkar.github.io/SQuAD-explorer/>
6. [https://cogcomp.cs.illinois.edu/page/resource\\_view/89](https://cogcomp.cs.illinois.edu/page/resource_view/89)
7. [https://protegewiki.stanford.edu/wiki/Main\\_Page](https://protegewiki.stanford.edu/wiki/Main_Page)